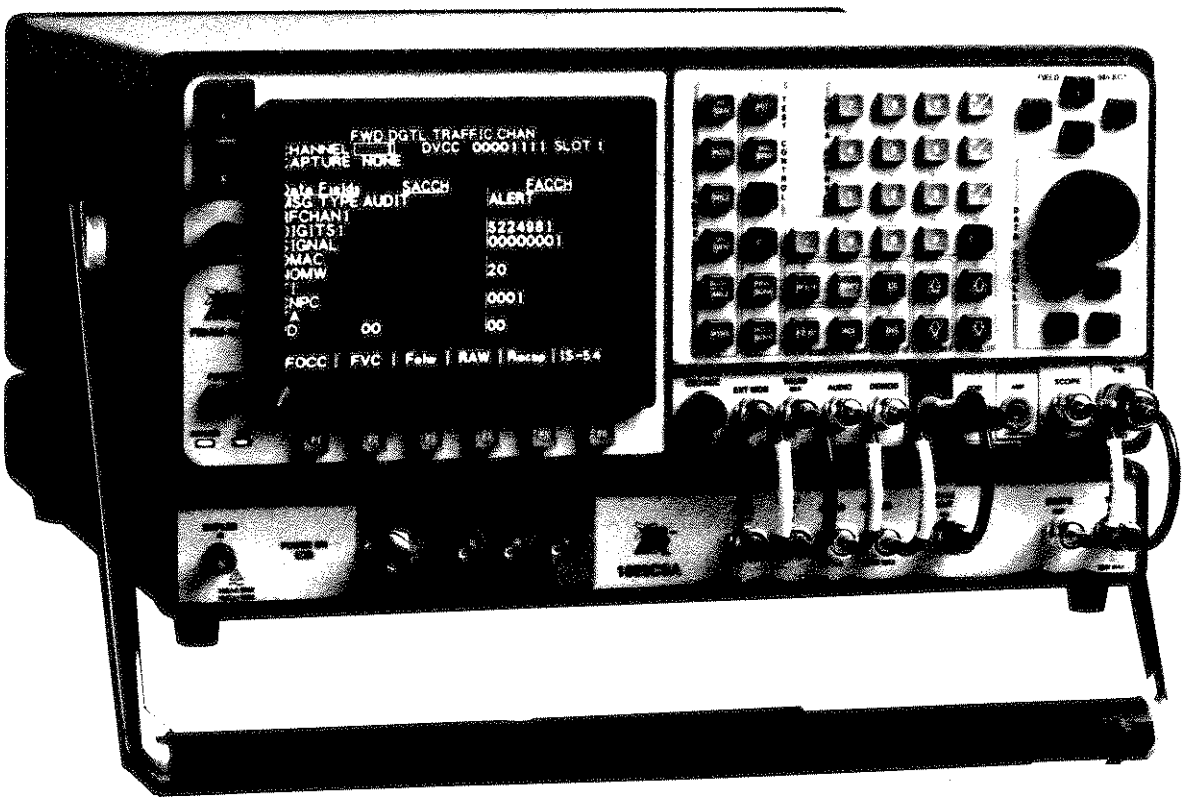
TMAC
USERS
MANUAL

FM/AM - 1600S
COMMUNICATIONS SERVICE MONITOR

FM/AM - 1600CSA
DUAL MODE CELLULAR SYSTEM ANALYZER



10200 West York Street / Wichita, Kansas 67215 U.S.A. / (316) 522-4981 / FAX 316-524-2623

1002-8601-300
1600S Ver 3.02
CSA Ver 5.10

PUBLISHED BY
IFR SYSTEMS INC.
Wichita, Kansas

LIST OF EFFECTIVE PAGES

The manual pages listed below that are affected by a current change or revision, are so identified by a revision number.

Date of issue for original and changed pages are:

Original	0	October 1994
Revision	1	August 1995
Revision	2	January 1996

TOTAL NUMBER OF PAGES IN THIS MANUAL IS 685 CONSISTING OF THE FOLLOWING

Pg. No.	Rev. No.	Pg. No.	Rev. No.
Title	2	7-30H Blank	2
Copyright	2	7-31 through 7-32	1
A	2	7-33 through 7-35	0
B Blank	2	7-36 through 7-37	1
I through xiii	2	7-38 through 7-42	0
xiv Blank	2	7-42A through 7-42B	2
1-1	0	7-43 through 7-51	0
1-2 Blank	0	7-51A through 7-51R	2
2-1 through 2-8	0	7-52	0
3-1 through 3-92	0	7-53 through 7-60	1
4-1 through 4-104	0	7-60A	2
5-1 through 5-23	0	7-60B	2
5-24 Blank	0	7-61 through 7-62	0
6-1 through 6-12	0	7-63	1
7-1	1	7-64	0
7-2 through 7-11	0	7-65 through 7-328	2
7-11A	2	8-1 through 8-22	1
7-11B Blank	2	9-1	1
7-12	2	9-2 through 9-9	0
7-13 through 7-14	0	9-10 through 9-10B	1
7-14A	2	9-11 through 9-15	0
7-14B Blank	2	9-15A through 9-15B	1
7-15 through 7-21	0	9-16 through 9-18	0
7-21A through 7-21H	2	A-1	0
7-22	0	A-2 Blank	0
7-22A	2	B-1	0
7-22B Blank	2	B-2 Blank	0
7-23 through 7-24	0	C-1 through C-3	0
7-24A	2	C-4 Blank	0
7-24B Blank	2	Index-1 through Index-5	0
7-25 through 7-30	0	Index-6 Blank	0
7-30A through 7-30G	2		

PREFACE

SCOPE

This Manual contains instructions for remotely operating the FM/AM-1600S Communication Service Monitor and FM/AM-1600CSA Dual Mode Cellular System Analyzer. The instruction level is relatively basic and presupposes no previous experience on the part of the operator with remote operation of a communication service monitor or cellular system analyzer. A basic understanding of communication electronics and cellular system formats is helpful. It is strongly recommended that the operator be thoroughly familiar with this manual as well as the Operation Manuals of the Test Sets before attempting to operate the Test Sets remotely.

ORGANIZATION

The FM/AM-1600S, FM/AM-1600CSA TMAC Users Manual is composed of the following sections:

SECTION 1 - INTRODUCTION

Provides an introduction to the TMAC language.

SECTION 2 - REMOTE CONFIGURATION

Provides step-by-step procedures for configuring the FM/AM-1600S or FM/AM-1600CSA for various remote operations.

SECTION 3 - GENERAL TMAC (TEST MACRO LANGUAGE)

Describes and explains the various features of the General TMAC language.

SECTION 4 - 1600S SPECIFIC TMAC COMMANDS

Lists and details the Specific TMAC commands for the FM/AM-1600S. Commands are arranged by Operation Mode for convenience.

SECTION 5 - CSA QUICK REFERENCE LIST

Briefly lists the FM/AM-1600S TMAC commands in alphabetical order.

SECTION 6 - CREATING AND LOADING CSA SPECIFIC PROGRAMS

Provides step-by-step procedures and examples for creating and loading CSA specific programs.

SECTION 7 - CSA SPECIFIC TMAC COMMANDS

Lists and details the Specific TMAC commands for the FM/AM-1600CSA. Commands are arranged by Operation Mode for convenience.

SECTION 8 - 1600S QUICK REFERENCE LIST

Briefly lists the FM/AM-1600CSA TMAC commands in alphabetical order.

SECTION 9 - CSA PROGRAM EXAMPLES

Provides functional CSA program examples.

NOMENCLATURE

The following names are used interchangeably to indicate the FM/AM-1600S Communication Service Monitor:

- FM/AM-1600S
- 1600S
- FM/AM-1600S Test Set
- Test Set (Paragraphs applying only to the FM/AM-1600S are so indicated. Test Set is used in other paragraphs to indicate either configuration.)

The following names are used interchangeably to indicate the FM/AM-1600CSA Dual Mode Cellular System Analyzer (and when referring to connectors, the auxiliary unit in particular):

- FM/AM-1600CSA
- CSA
- FM/AM-1600CSA Test Set
- Test Set (Paragraphs applying only to the FM/AM-1600CSA are so indicated. Test Set is used in other paragraphs to indicate either configuration.)

TABLE OF CONTENTS

SECTION 1 - INTRODUCTION

SECTION 2 - INSTALLATION

SECTION 3 - GENERAL TMAC (TEST MACRO LANGUAGE)

SECTION 4 - 1600S SPECIFIC TMAC COMMANDS

SECTION 7 - CSA SPECIFIC TMAC COMMANDS

APPENDICES

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION 1 - INTRODUCTION

TMAC (Test Macro Language) was developed by IFR Systems Inc. to operate IFR test equipment in a remote configuration and allow internal versatile software to be created. TMAC is written so not just single measurements but entire batteries of tests may be performed. Combining this capability with the test functions available with the FM/AM-1600S and FM/AM-1600CSA gives the user a great deal of flexibility in testing communication devices in an ATE environment. Additionally this language also provides a format to store and perform user defined test sequences. TMAC is based on the IEEE-488.2 format and conforms to the SCPI Standard. Some commands and operators, created before and not conforming to the SCPI Standard, are so noted.

TMAC is written on several layers of complexity. The first layer consists of the Machine Specific TMAC Commands. These commands are what actually set up and control the Test Set and take the measurements. The Specific TMAC commands for the FM/AM-1600S and FM/AM-1600CSA encompass the complete operation of the Test Set. Every function of the FM/AM-1600S and FM/AM-1600CSA can be performed remotely with the exception of power up of the Test Set and CRT intensity control.

The second level of TMAC is the framework that is used to combine the machine specific commands into a coherent and fluid process. Decision making commands allow different procedures to be performed under differing circumstances. These decisions may be based on complex expressions as well as simple boolean answers being returned from the Test Set. Looping commands allow the continual monitoring of a parameter while performing tasks. Math and logic function commands together with bit manipulating commands allow the use and evaluation of complex functions. TMAC also provides a variety of data structures to assist in developing test procedures. Graphic commands allow the development of user defined screens and menus to aid the user in performing test sequences. The Status System provided allows the monitoring of the Status Registers called for in the IEEE-488.2 Standard.

The third layer and one of the main strengths of TMAC is the capability to define macros with procedures built of machine specific commands as organized by the framework commands. Macros allow step by step test procedures to be developed, stored and locally or remotely implemented. Using macros is easy. Entering the name of the macro along with any variable parameters, executes the macro. Macros can be executed within other macros, allowing complicated procedures to be divided into smaller tasks. Macros can be initiated remotely, from the Front Panel Keyboard or automatically upon power up of the Test Set.

The final layer of TMAC is the ability to Multitask Macros. The Multitasking feature of TMAC allows several macros to alternate command execution at once giving the appearance of being executed simultaneously. Multitasking in TMAC allows the user to activate a task, pass to the next task, put a task to sleep, wake a sleeping task and stop a task.

SECTION 2 - REMOTE CONFIGURATION

2-1 GENERAL

This section provide general configuration procedures for remote operation for the FM/AM-1600S and the FM/AM-1600CSA. Remote operation depends on having Test Set settings match user equipment and interface settings. Settings in configuration procedures may require variation depending on user equipment.

2-2 CONFIGURING FOR RS-232 OPERATION

2-2-1 REMOTE OPERATION USING HOST SYSTEM

The FM/AM-1600S and FM/AM-1600CSA Test Sets can be remotely operated from a Host System. The Host System can be a PC using Terminal Emulation Software or an RS-232 Terminal.

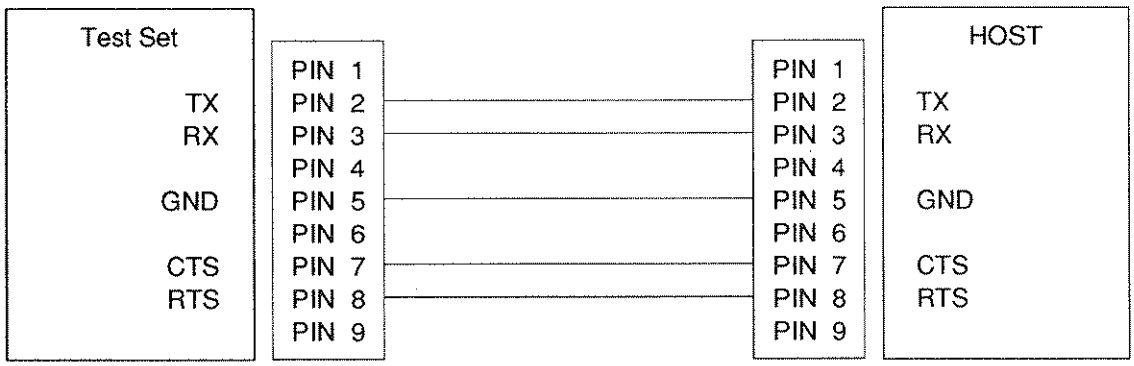


Figure 2-1 Test Set - Host Cable Connections

A. 1600S - HOST SYSTEM SETUP AND OPERATION

STEP	PROCEDURE

1. Refer to Figure 2-1 and connect Host System Comm Connector to Test Set RS-232 Connector on FM/AM-1600S rear panel, using standard 9-pin to 9-pin RS-232 cable.
2. Press MTRS Mode Key to access Meter Menu.
3. Press "AUX" Soft Function Key F6 to access Auxiliary Functions Menu.
4. Press 5 DATA ENTRY Key to access Configure Submenu.
5. Press 1 DATA ENTRY Key to access Configure RS-232 Submenu.
6. Press 1 DATA ENTRY Key to access RS-232 Operation Mode. Press 2 DATA ENTRY Key to select ***Host***.
7. Press 2 DATA ENTRY Key to access Baud Rate Submenu. Use DATA ENTRY Keys to select desired Baud Rate.
8. Press 3 DATA ENTRY Key until Data Bits toggles to ***8***.

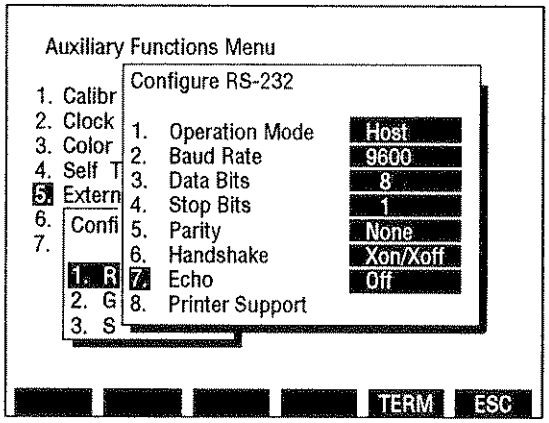


Figure 2-2 FM/AM-1600S RS-232 Configure Menu

9. Press 4 DATA ENTRY Key until Stop Bits toggles to ***1***.

10. Press 5 and 1 DATA ENTRY Keys to select ***None*** for Parity.

11. Press 6 and 3 DATA ENTRY Keys to select ***Xon/Xoff*** Handshaking.

12. Press 7 DATA ENTRY Key until Echo is toggled ***Off*** (faster action and displaying transmission is not desired) or ***On*** (displaying transmission is desired).

13. Press "ESC" Soft Function Key F6 twice to return to Auxiliary Functions Menu.

14. Set Terminal Emulator Software or RS-232 Terminal parameters to match Table 2-1. Set Duplex or Echo mode as desired according to Terminal Emulator software program used.

PARAMETER	SETTING
Baud Rate	Same as Test Set
Data Bits	8
Stop Bits	1
Parity	None
Handshaking	Xon/Xoff

Table 2-1 Terminal Settings for RS-232 Host Operation

15. Send desired 1600S commands from Host terminal to FM/AM-1600S Test Set. Write and save macros, as desired, using Host System text editor. Transfer macros and programs, as desired, to Test Set using Host System.

B. CSA - HOST SYSTEM SETUP AND OPERATION

STEP	PROCEDURE

1. Refer to Figure 2-1 and connect Host System Comm Connector to Test Set RS-232 Connector on FM/AM-1600CSA rear panel, using standard 9-pin to 9-pin RS-232 cable.

2. Press DPLX Mode Key to access Duplex Operation Screen.

3. Press "Sp Tst" Soft Function Key F5 to access Dual Mode Cellular main menu.

4. Press "AUX2" Soft Function Key F5 to access Auxiliary Functions Menu (#2).

5. Press 5 DATA ENTRY Key to access External I/O Parameters Screen.

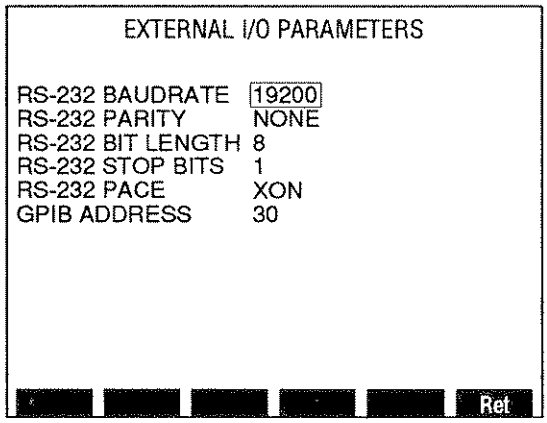


Figure 2-3 CSA External I/O Parameters Screen

6. Use FIELD SELECT Keys to select parameter. Use DATA SCROLL Spinner or DATA SCROLL ↑ and ↓ Keys to edit parameter. Set parameters as desired (see Figure 2-3).

7. Press DPLX Mode Key to access Duplex Operation Screen.

8. Set Terminal Emulator Software or RS-232 Terminal parameters to match CSA settings.

9. Send desired CSA commands from Host terminal to FM/AM-1600CSA Test Set. Write and save macros, as desired, using Host System text editor. Transfer macros and programs, as desired, to Test Set using Host System.

2-2-2 REMOTE OPERATION USING MODEM

The FM/AM-1600S and FM/AM-1600CSA Test Sets can be connected to a Modem and then operated remotely over the telephone line.

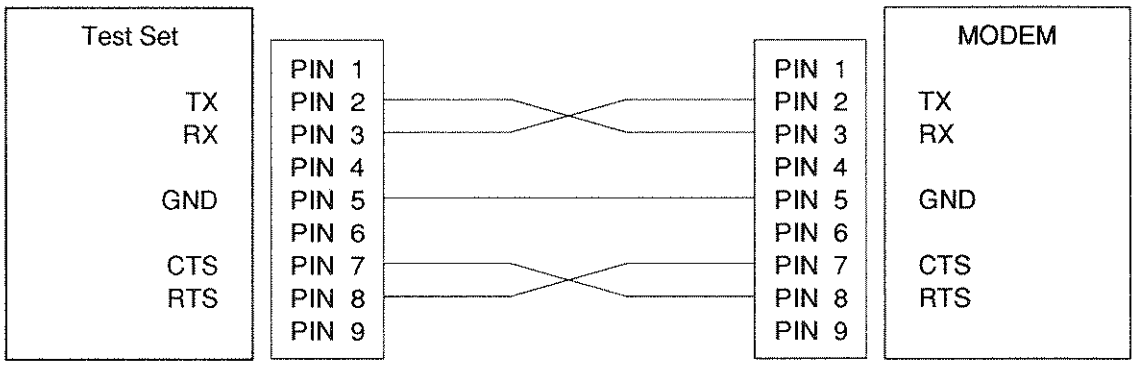


Figure 2-4 Test Set - Modem Cable Connection

A. 1600S - MODEM SETUP AND OPERATION

STEP	PROCEDURE

1. Refer to Figure 2-4 and connect Modem to Test Set RS-232 Connector on FM/AM-1600S rear panel.
2. Press MTRS Mode Key to access Meter Menu.
3. Press "AUX" Soft Function Key F6 to access Auxiliary Functions Menu.
4. Press 5 DATA ENTRY Key to access Configure Submenu.
5. Press 1 DATA ENTRY Key to access Configure RS-232 Submenu.
6. Press 1 DATA ENTRY Key to access RS-232 Operation Mode. Press 2 DATA ENTRY Key to select ***Host***.
7. Press 2 DATA ENTRY Key to access Baud Rate Submenu. Set Baud Rate as desired, according to Modem requirements.
8. Press 3 DATA ENTRY Key until Data Bits toggles to ***8***.
9. Press 4 DATA ENTRY Key until Stop Bits toggles to ***1***.
10. Press 5 and 1 DATA ENTRY Keys to select ***None*** for Parity.
11. Press 6 and 1 DATA ENTRY Keys to select ***None*** for Handshaking.
12. Press 7 DATA ENTRY Key until Echo is toggled ***Off***.
13. Press "ESC" Soft Function Key F6 twice to return to the Auxiliary Functions Menu.

STEP	PROCEDURE

14. Press "TERM" Soft Function Key F5 to access RS-232 Monitor Screen.

15. Use DATA ENTRY Keys to enter attention prefix and dialing command of Modem.

16. Operate FM/AM-1600S Test Set from destination telephone number.

B. 1600CSA - MODEM SETUP AND OPERATION

STEP	PROCEDURE

1. Refer to Figure 2-4 and connect Modem to Test Set RS-232 Connector on FM/AM-1600CSA rear panel.

2. Press DPLX Mode Key to access Duplex Operation Screen.

3. Press "Sp Tst" Soft Function Key F5 to access Dual Mode Cellular main menu.

4. Press "AUX2" Soft Function Key F5 to access Auxiliary Functions Menu (#2).

5. Press 5 DATA ENTRY Key to access External I/O Parameters Screen.

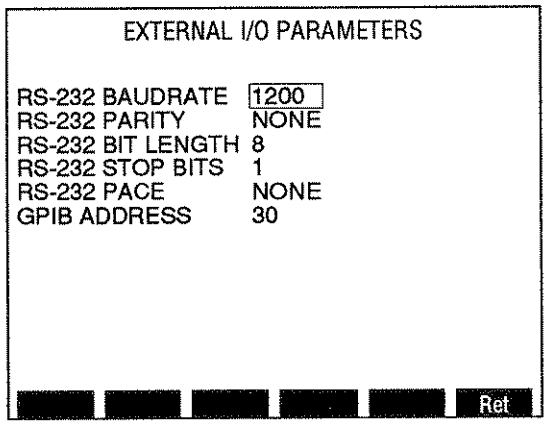


Figure 2-5 CSA External I/O Parameters Screen (Modem)

6. Use FIELD SELECT Keys to select parameter. Use DATA SCROLL Spinner or DATA SCROLL ↑ and ↓ Keys to edit parameter. Set parameters according to Figure 2-5.

7. Press DPLX Mode Key to access Duplex Operation Screen.

8. Call Modem and operate FM/AM-1600CSA Test Set from destination telephone number.

2-2-3 OPERATING TEST SET AS RS-232 TERMINAL (1600S Only)

The FM/AM-1600S Test Set can be used as an RS-232 Terminal.

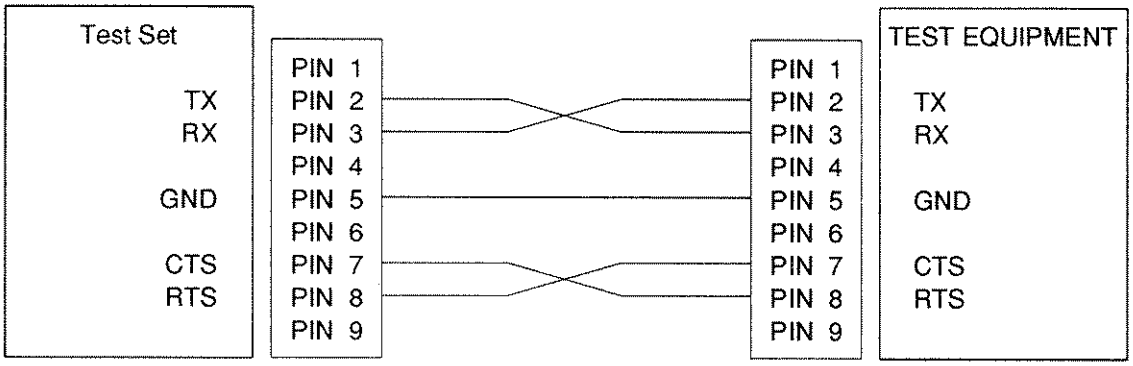


Figure 2-6 Test Set (as RS-232 Terminal) Connections

STEP	PROCEDURE

1. Refer to Figure 2-6 and connect Test Equipment to Test Set RS-232 Connector on FM/AM-1600S rear panel.
2. Press MTRS Mode Key to access Meter Menu.
3. Press "AUX" Soft Function Key F6 to access Auxiliary Functions Menu.
4. Press 5 DATA ENTRY Key to access Configure Submenu.
5. Press 1 DATA ENTRY Key to access Configure RS-232 Submenu.
6. Press 1 DATA ENTRY Key to access RS-232 Operation Mode. Press 2 DATA ENTRY Key to select ***Host***.
7. Press 2 DATA ENTRY Key to access Baud Rate Submenu. Set Baud Rate as desired, according to Test Equipment requirements.
8. Press 3 DATA ENTRY Key until Data Bits toggles to ***8***.
9. Press 4 DATA ENTRY Key until Stop Bits toggles to ***1***.
10. Press 5 and 1 DATA ENTRY Keys to select ***None*** for Parity.
11. Press 6 and 3 DATA ENTRY Keys to select ***Xon/Xoff*** for Handshaking.
12. Press 7 DATA ENTRY Key until Echo is toggled ***On***.
13. Press "ESC" Soft Function Key F6 twice to return to Auxiliary Functions Menu.
14. Set Test Equipment parameters to match Table 2-1.

STEP PROCEDURE

15. Press "TERM" Soft Function Key F5 to access RS-232 Monitor Screen.

16. Use DATA ENTRY Keys to enter commands. Use Soft Function Keys to enter characters not included in DATA ENTRY Keys. Use SHIFT Key to toggle between numeric and alphabetic keys.

17. Press any MODE Key to exit RS-232 Monitor Screen.

2-3 CONFIGURING FOR GPIB OPERATION

The FM/AM-1600S and FM/AM-1600CSA (with GPIB installed) Test Sets can be remotely operated, individually or at the same time (with separate addresses), using GPIB protocol.

2-3-1 1600S - GPIB SETUP AND OPERATION

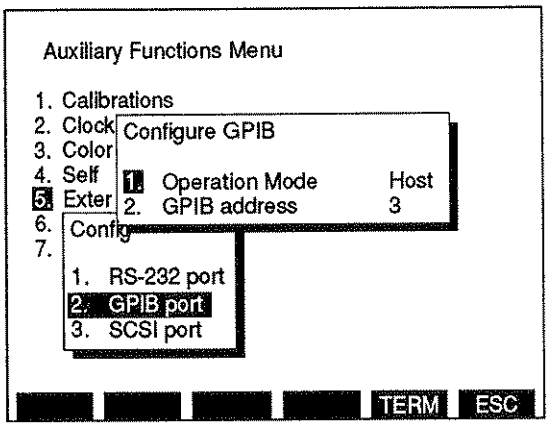
STEP PROCEDURE

1. Press MTRS Mode Key to access Meter Menu.

2. Press "AUX" Soft Function Key F6 to access Auxiliary Functions Menu.

3. Press 5 DATA ENTRY Key to access Configure Submenu.

4. Press 2 DATA ENTRY Key to access Configure GPIB Submenu.



8610025

Figure 2-7 1600S Configure GPIB Submenu

5. Press 1 DATA ENTRY Key to access GPIB Operation Mode. Use DATA ENTRY Keypad to select operation Mode. Select ***Talk/Listen*** to operate FM/AM-1600S from GPIB Controller (see 3-15-2). Select ***Controller*** to operate other test equipment from FM/AM-1600S (see 3-15-3).

6. Press 2 DATA ENTRY Key to access GPIB address. Use DATA ENTRY Keypad to select GPIB address to not conflict with address of any other device connected to GPIB.

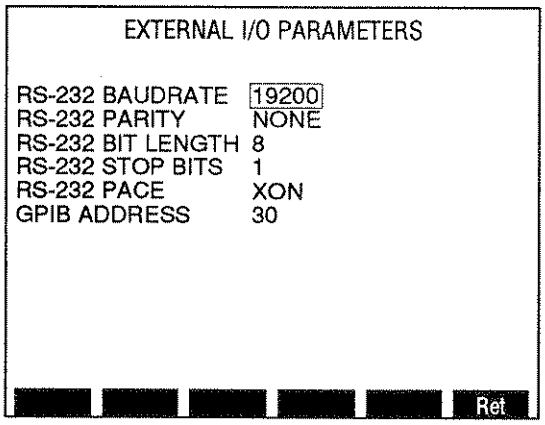
7. Connect GPIB cable to GPIB Connector on FM/AM-1600S Rear Panel.

8. Initiate parallel remote operation.

2-3-2 1600CSA - GPIB SETUP AND OPERATION (GPIB option installed)

STEP PROCEDURE

1. Press DPLX Mode Key to access Duplex Operation Screen.
2. Press "Sp Tst" Soft Function Key F5 to access Dual Mode Cellular main menu.
3 Press "AUX2" Soft Function Key F5 to access Auxiliary Functions Menu (#2).
4 Press 5 DATA ENTRY Key to access External I/O Parameters Screen.



9117127

Figure 2-8 CSA External I/O Parameters Screen (GPIB)

5. Use FIELD SELECT Keys to select GPIB ADDRESS parameter.
6. Use DATA SCROLL Spinner or DATA SCROLL ↑ and ↓ Keys to select GPIB ADDRESS, not conflicting with address of any other device connected to GPIB.
7. Press DPLX Mode Key to access Duplex Operation Screen.
8. Connect GPIB cable to GPIB Connector on FM/AM-1600CSA Rear Panel.
9. Initiate parallel remote operation from GPIB Controller (see 3-15-1).

SECTION 3 - GENERAL TMAC (TEST MACRO LANGUAGE)

3-1 SYNTAX

When working with TMAC, punctuation marks are used to establish a format of how commands and parameters are entered. The syntax needed to use TMAC is detailed below.

- Colon (:)

TMAC commands are arranged in hierarchical levels using the colon to separate the different levels. A colon before the beginning of a command (**:FGEN**) signifies the command starts at the first hierarchical level.

Example: `FGEN:GEN1:SHAPE:SIN`

FGEN precedes all Function Generator commands. The **FGEN:GEN1** precedes all Function Generator commands dealing with Generator 1. **FGEN:GEN1:SHAPE** precedes all Function Generator commands selecting the waveform of Generator 1. **FGEN:GEN1:SHAPE:SIN** selects a sine wave for the waveform of Generator 1 of the Function Generator.

A new line character marks the end of a command and places the next command at the first command level.

- Semicolon (;)

A semicolon signifies the end of a command and starts the next command at the last level of the previous command.

Example: `FGEN:GEN1:FREQ 2000;SHAPE:SIN;:SCREEN:FUNC`

The first command, **FGEN:GEN1:FREQ 2000**, leaves the TMAC Compiler at the **FGEN:GEN1** level. Preceded by a semicolon, the next command, **SHAPE:SIN**, starts from this level. The next semicolon leaves the TMAC Compiler at the **FGEN:GEN1:SHAPE** level. The following colon resets the Compiler to the first hierarchical level to start the next command, **SCREEN:FUNC**.

Example: `DUP:INP:FREQ?;MOD:USER:MOD:DATA;FILTER 30;:SCREEN:DUP`

This command line performs the following commands:

```
DUP:INP:FREQ?
DUP:INP:MOD:USER:MOD:DATA
DUP:INP:MOD:USER:FILTER 30
SCREEN:DUP
```

- Whitespace

At least one space is needed between commands and parameters. Other spaces or lack of them between items do not affect command execution. Spaces inside command words or between command word and ending punctuation (**ME AS:FREQ?** or **MEAS:FREQ ?**) are not allowed.

- Comma (,)

Commas are used between parameters when more than one is listed.

Example: `MACRO 1,2,3,4`

- Back slash (\)

A back slash allows a command to be continued on the next line. The back slash cannot break a command word.

Example:
```
FGEN:GEN1:\
FREQ 2000
```

This is the same command as **FGEN:GEN1:FREQ 2000**.

Example:
```
FGEN:GE\
N1:FREQ 2000
```

This command fails to execute and produces an error message because command word **GEN1** is broken.

Many commands can be entered using a short form or a long form. In this manual, the short form is shown in upper case, while the remainder of the long form is shown in lower case. Upper and lower case letters are used only in this manual to differentiate between the long and short form of commands. TMAC does not differentiate between upper and lower case letters; therefore entries can be either.

Example:
```
STATus:QUEStionable:CONDition?
```

STAT:QUES:COND? is the same as **STATUS:QUESTIONABLE:CONDITION?**

3-2 COMMENTS

Comments provide clarity in macros and are ignored by the TMAC compiler/interpreter. There are two types of comments.

Characters // cause the rest of line to be a comment.

Example:
```
PRINT COS(freq*SQR(2)) // PRINTS COS OF FREQ × SQUARE ROOT OF 2
```

The message `PRINTS COS OF FREQ × SQUARE ROOT OF 2` is not executed by the TMAC compiler/interpreter.

Characters /* and */ start and end the second type of comment. It can contain several lines.

Example:
```
PRINT COS(freq*SQR(2)) /* prints cosine of the quantity RF Generator
    frequency multiplied by the square root of 2 */
```

The TMAC compiler/interpreter ignores everything between the comment characters.

3-3 VARIABLES AND ARRAYS

3-3-1 GENERAL

Variables and arrays are used to hold numeric values and can be local or global. If a variable or array is declared inside a macro, it is local and is used inside that macro only. If it is declared outside a macro, it is global and is used outside or inside any macro (unless a local variable has the same name as the global variable). Locality takes precedence.

The number of variables and arrays allowed is limited by the amount of memory available. There is a maximum of 998 memory locations for global variables. A global variable takes one location, global arrays take a location for each array element and global string variables take 9 locations each. Local variables are also limited by memory space which varies depending on many factors including call nesting (see Macros in 3-9). One to two hundred local variables are usually available with arrays taking the space of a variable for each array element and string variables taking nine times the space needed for a variable.

The first character of a variable or array name is a letter while the remaining characters can be letters, digits 0 through 9 and the underscore. The maximum length is 31 characters. There are 26 free variables that do not need to be declared. These have the single letter names A through Z. To declare a variable, use the **VAR** command:

VAR *name, name,..... ,name*

Example: `VAR Freq,Rad,Mod`

Freq, *Rad* and *Mod* are now declared variables.

Variables can be initialized when they are declared by including an equal sign and the value. Variables are equal to 0 when declared if not changed by the user.

Example: `VAR Rad = 4.32876, Mod`

Rad is declared and assigned a value of 4.32876. Mod is declared and contains a value of 0.

Arrays are also declared using the VAR command. The highest index is included in brackets as shown:

VAR *name*[*index*]

The index count starts with 0, making *index* one less than the total number of variables in the array.

Example: `VAR BB[9]`

This command declares array BB with 10 variables: BB[0] through BB[9].

Braces are used to initialize arrays. More than one can be initialized at a time.

Examples:
```
VAR BUNCH[10] = {1,2,3,4,5,6,7,8,9,10,11}
VAR AA[2]={1,2,3},  BB[1]={7,4},  CC[3]={24, 345.754, 2, 100}
```

A variable and an array cannot have the same name at the same time. Accessing an array without specifying an index yields erroneous results.

3-3-2 SAVING VARIABLES AND ARRAYS (1600S ONLY)

NOTE: Refer to section 7-2 to run 1600S only commands when operating as the FM/AM-1600CSA.

Global variable and array values are not saved in memory when the Test Set power is turned off. To save variable or array values in non-volatile memory, use the **NVSAV** command. To recall a variable, use the **NVRCL** command:

NVSAV *name,location*

NVRCL *name,location*

name is the name of the variable or array and *location* is the memory location. Each variable uses one memory location and arrays use a memory location for each value. Free variables, local variables and local arrays cannot be saved in non-volatile memory.

Example:
```
VAR DD
NVSAV DD,10
```

DD is now saved in non-volatile memory location 10. To recall DD, use the NVRCL command.

Example:
```
NVRCL DD,10
```

The content of memory location 10 is loaded into variable DD.

When an array is saved using the **NVSAV** command, the whole array is saved in consecutive memory locations.

Example:
```
VAR list[5]
NVSAV list,1
```

The values of list are saved in memory locations 1 through 6 (6 values starting at 1). Arrays are recalled the same as variables.

Example:
```
NVRCL list,1
```

The contents of memory locations 1 through 6 are loaded into list[0] through list[5].

NOTE: Overwriting previously used memory locations deletes previous entries.

Example:
```
VAR List[4]={2,4,6,8,10}    // Declares an array named List.
VAR Group[2]={23,46,69}     // Declares an array named Group.
NVSAV List, 10              // Saves List in memory locations
                            // 10 through 14.
NVSAV Group,12              // Saves Group in memory locations
                            // 12 through 14.
NVRCL List,10               // Recalls memory locations 10 through
                            // 14 and assigns the values to List.
```

Three of the saved values of List were lost when the same memory locations were used to save the array Group.

3-4 CONSTANTS AND DATA ARRAYS

Constants are declared using the **CONST** command. Constants are global only and cannot be declared inside a macro.

CONST *name, expression, name, expression,..., name, expression*

A constant *name* starts with a letter while the remaining characters can be letters, numbers and the underscore. A constant *name* can be 2 to 31 characters in length.

Example:
```
CONST PI, 3.1415, FREQ, 3000
```

String data cannot be assigned to a constant. Expressions do not have to be in simplest form.

Example:
```
CONST AA,(4+6)/2
```

AA is set equal to 5.

Once a constant is declared with the **CONST** command, it is changed only with another **CONST** command.

Example:
```
CONST AA,36
AA=20
```

The second command results in an error message and is not performed, leaving AA equal to 36.

Constant data arrays are declared using the **DATA** command:

DATA *name* = {*expression, expression,..., expression*}

A data array *name* starts with a letter while the remaining characters can be letters, numbers and the underscore. A data array *name* can be 2 to 31 characters in length.

Example:
```
DATA offsets_1 = {2,3,4,5,6,7,8,9}
```

Like a constant, data arrays cannot be assigned string data nor do expressions have to be in their simplest form. Once a data array is declared, it is changed only by another **DATA** command. Contents of data arrays are accessed using the index enclosed in brackets.

Example:
```
DATA Points = {11.6, 17.3, 22.4, 27.1}
Y = Points[2]
```

Variable Y is set equal to 22.4.

3-5 NUMERIC FORMATS

3-5-1 NUMERICAL NOTATION

Numerals are expressed as fixed point decimal, scientific notation or as an integer. Numerals are specified as negative with a - character as shown:

Example: `-51`

This denotes a negative 51.

The maximum fixed point decimal numeral that can be entered is ±2147483647. Scientific notation uses the letter e to denote an exponent of 10.

Example: `54e7`

This number signifies 54 multiplied by 10 to the 7th power or 540,000,000.

100 is the maximum power of 10 that can be entered, although the maximum number in scientific form allowed is + 1.797693e308. Numbers with a larger power of 10 than 100 are assigned to a variable by multiplying powers of 10.

Example: `Large = 54e7*1e100*1e100`

This example assigns 54e207 or 54 multiplied by 10 to the 207th power to variable Large.

NOTE: For execution reasons, the CSA uses single precision floating point numbers e.g. 1e±38.

3-5-2 NUMERIC FORM

To enter data in a different format (base 2, 8 or 16), precede data with one of the following characters:

#B	Binary
#Q	Octal
#H	Hexadecimal

Data format for the returned data for queries is changed to base 2, 8, 10 or 16 using the **FORMat** command. The default format is base 10.

FORMat BINary	Data is returned in base 2 format.
FORMat OCTal	Data is returned in base 8 format.
FORMat HEXadecimal	Data is returned in base 16 format.
FORMat ASCII (1600S)	Data is returned in base 10 format.
FORMat DECimal (CSA)	Data is returned in base 10 format.

Binary, octal or hexadecimal data received from the Test Set (1600S only) is preceded by #B, #Q or #H notation. Data in scientific notation is not affected by these commands.

3-6 MATHEMATICAL OPERATORS

TMAC provides a variety of mathematical operators and bit manipulating functions to perform mathematical operations. A 32 bit word is used for binary and bitwise operations.

3-6-1 OPERATORS

= Assignment operator

Assigns a value to a variable. Variable on the left of the = operator is the given value computed for the expression on the right.

Examples: `x=2;  Mod = w * 2 + z`

The assignment operator does not denote an equation.

Example: `x = x + 2`

Assigns a new value to x, 2 greater than the previous value.

** Exponential operator (does not conform to SCPI Standard)

Consists of two consecutive asterisks.

Examples: `y=4**-2;  MM = NN**4;  Freq = C**D`

The first example assigns y to 4 raised to the -2 power. The second example assigns MM to NN raised to the 4th power. The third example assigns Freq to C raised to the D power.

* Multiplication operator

Examples: `x=4*6;  T = 25*S;  JJ = K*LL`

The first example assigns x to 4 multiplied by 6. The second example assigns T to 25 multiplied by the value of S. The third example assigns JJ to the value of K multiplied by the value of LL.

/ Division operator

Examples: `Y=8/3.867;  Z = x/7;  P = Q/R`

The first example assigns y to 8 divided by 3.867. The second example assigns Z to the value of x divided by 7. The third example assigns P to the value of Q divided by the value of R.

NOTE: A command to divide by zero results in an error message.

+ Addition operator and positive unary operator

Examples: `G=4+F;  AA = B1 + B2,  +X`

The first example assigns G to 4 added to the value of F. The second example assigns AA to the value of B1 added to B2. The last example is the value of X.

- Subtraction operator and negative unary operator

Examples: `H= I-5;  V = U-T,  -B`

The first example sets assigns H to 5 subtracted from the value of I. The second example assigns V to the value of T subtracted from the value of U. The last is the negative or opposite value of B.

% Modulo operator (does not conform to SCPI Standard)

Returns the integer truncated remainder of a division operation. If operands are not integers, they are truncated first.

Examples: `x=7%3;  RR=16%4;  N = 20.5%3.3;  M = 27%L;`

The first example assigns x to 7 modulo 3 (1). The second example assigns variable RR to 16 modulo 4 (0). The third example assigns N equal to 20 modulo 3 (2). The fourth example assigns M to 27 modulo the value of L.

-- Decrement and ++ Increment operators

The decrement operator subtracts 1 from the value of a variable either before or after execution of the command it is included in. The increment operator acts in the same manner, except it adds instead of subtracts.

--variable	Decrements variable before execution of the command.
variable--	Decrements variable after execution of the command.
++variable	Increments variable before execution of the command.
variable++	Increments variable after execution of the command.

Examples:
```
y = 7
x = --y
```

The `x = --y` command subtracts 1 from y and then assigns this value to x. After the second command is executed, x and y equal 6.

Examples:
```
y = 7
x = y++
```

The `x = y++` command assigns x to the y value (7) and then adds 1 to y. After the command is executed, x equals 7 and y equals 8.

~ Bitwise complement

Inverts the binary representation of the operand using the one's complement form of negative numbers.

Example: `X = ~6`

0110 (6) is inverted to 1001 (one's complement of -7) and -7 is assigned to X.

! Logical negation (NOT) unary operator

Reverses the truth value.

Example: `!D`

The example is true when D equals 0.

| Bitwise OR

Performs OR operation on each bit of the binary representation of the two operands and returns the result in the original form of the operands. Table 3-1 lists the bit combinations and their OR operation results.

operand 1	operand 2	OR result
0	0	0
0	1	1
1	0	1
1	1	1

Table 3-1 OR Operation Results

Example: y = 4 | 2

OR	4	1 0 0
	2	0 1 0
	6	1 1 0

The left bit of 4 (100) and 2 (010) are 1 and 0, providing an OR result of 1. The middle bits are 0 and 1, providing an OR result of 1. The right bits are both 0, providing an OR result of 0 and setting y equal to 6 (110).

Example: Z = 12 | 10

OR	12	1 1 0 0
	10	1 0 1 0
	14	1 1 1 0

& Bitwise AND

Performs AND operation on each bit of the binary representation of two operands and returns the result in the original form of the operands. Table 3-2 lists the bit combinations and the AND operation results.

operand 1	operand 2	AND result
0	0	0
0	1	0
1	0	0
1	1	1

Table 3-2 AND Operation Results

Example: `JJ = 4 & 6`

```
     4      1 0 0
AND  6      1 1 0
     4      1 0 0
```

The left bit of 4 (100) and 6 (110) are 1, providing an AND result of 1. The middle bits are 0 and 1, providing an AND result of 0. The right bits are both 0, providing an AND result of 0 and setting JJ equal to 4 (100).

Example: `w = 12 & 10`

```
     12     1 1 0 0
AND  10     1 0 1 0
      8     1 0 0 0
```

^ Bitwise XOR (conflicts with SCPI Standard for Exponential operator)

Performs XOR operation on each bit of the binary representation of two operands and returns the result in the operands original form of the operands. Table 3-3 lists the bit combinations and the XOR operation results.

operand 1	operand 2	XOR result
0	0	0
0	1	1
1	0	1
1	1	0

Table 3-3 XOR Operation Results

Example: `y = 4 ^ 6`

```
     4      1 0 0
XOR  6      1 1 0
     2      0 1 0
```

The left bit of 4 (100) and 6 (110) are 1 providing a XOR result of 0. The middle bits are 0 and 1, providing a XOR result of 1. The right bits are both 0, providing a XOR result of 0 and setting y equal to 2 (010).

Example: `v = 12 ^ 10`

```
     12     1 1 0 0
XOR  10     1 0 1 0
      6     0 1 1 0
```

>> Shift right and
<< left operators

Shift right (>>) and shift left (<<) operators shift the operands binary representation a specified number of bits and returns the result in the operands original format. During shifts, emptied bit positions are filled with zeros. Shift operators use the twos complement representation for negative numbers.

Example: `Q = 14>>1`

14 (1110) shifts right 1 bit to become 7 (0111) and is loaded into variable Q.

Example:
```
VAR DIV4
m = 2
DIV4 = 75<<m
```

75 (1001011) shifts left 2 bits to become 300 (100101100) and is loaded into variable DIV4.

Example: `X = -19<<3`

-19 (11101101) shifts left 3 bits to become -152 (111101101000).

3-6-2 ORDER OF CALCULATION

When more than one operator is contained in an expression, the operators are calculated in the following order:

- Contents of parenthesis, calculating from inner sets to outer sets of parenthesis.
- Positive (+), negative (-), bitwise complement (~) and logical negation (!).
- Exponentiation (**).
- Multiplication (*), division (/) and modulus (%).
- Addition and subtraction.
- Shift left (<<) and right (>>) operations.
- Bitwise AND (&).
- Bitwise XOR (^).
- Bitwise OR (|).
- Logical AND.
- Logical OR.
- Condition? *true*:*false* (Refer to **IF ELSE** [Shorthand] command in section 3-18.)

3-7 MATHEMATICAL FUNCTIONS

TMAC provides mathematical functions to assist in mathematical calculations:

TRUE,ON

Produces 1.

Example:
```
X=TRUE
```

X is set to 1.

FALSE,OFF

Produces 0.

Example:
```
Y=OFF
```

Y is set to 0.

RND(n)

Produces a random integer between 0 and n. If n is not an integer, it is truncated.

Example:
```
WW=RND(100)
```

WW now contains a randomly generated number between 0 and 100.

RAND x

Sets the seed (starting point) for the random generator which allows the random generator to produce the same series of random numbers each time the starting point is set to the same value.

Example:
```
RAND 12
FOR X=1 TO 10 STEP 1
Z=RND(100)
PRINT Z
NEXT X
```

The RAND 12 command sets a starting point for the RND command. When the RND(100) command is executed 10 times the following numbers are generated:

47, 77, 31, 27, 39, 100, 57, 46, 41 and 30

Each time RAND is set to 12, the RND(100) command produces the same series of random numbers.

LOG(n)

Produces the base 10 logarithm of n.

Example:
```
VAR Log_C
C = 5
Log_C = LOG(C)
```

Log_C now holds base 10 logarithm of 5 (0.698970004).

NOTE: If n ≤ 0, LOG(n) command execution results in an error message as this function is not defined for these values.

LN(n)

Produces the natural logarithm (base e) of n.

Example:
```
VAR Nat_log
Nat_log = LN(D)
```

Nat_log now holds natural logarithm of 5 (1.609438).

NOTE: If n ≤ 0, LN(n) command execution results in an error message as this function is not defined for these values.

ABS(n)

Produces the absolute value of n.

Example:
```
Y=ABS(-4)
```

Y now equals +4.

Example:
```
Z=-7
POS=ABS(Z)
```

POS now holds +7.

FLOOR(n)

Produces the truncated value of n.

Example:
```
VAR Round_off
Round_off = FLOOR(5.99)
```

Round_off now holds 5 the truncated value of 5.99.

SQR(n)

Produces the positive square root of n.

Example:
```
VAR Square_Root
Square_Root = SQR(30)
```

Square_Root now holds the positive square root of 30 (5.477226).

NOTE: The square root function returns 0 for the square root of negative numbers.

EXP(n)

Produces e to the power of n.

Example:
```
VAR Expo
x=1.57
Expo=EXP(x)
```

Expo now holds e to 1.57th power (4.810255).

SIN(n)

Produces the sine of n measured in radians.

Example:
```
VAR Sine_of_radians
Sine_of_radians=SIN(3.1415/3)
```

Sine_of_radians now holds the sine of $\pi/3$ (0.86601).

COS(n)

Produces the cosine of n measured in radians.

Example:
```
VAR Cosine_of_radians
Cosine_of_radians=COS(5*3.1415/6)
```

Cosine_of_radians now holds the cosine of $5\pi/6$ (-0.8659868).

SIGN(n)

Produces -1 if n<0, 0 if n=0 or 1 if n>0.

Example:
```
S=SIGN(-6)
```

Value of S is set to -1.

3-8 STRING VARIABLES AND FUNCTIONS

A string is a segment of text consisting of ASCII characters. String variables are declared using the **STRING** command. There is a free string variable, represented by the $ character that does not need to be declared. String variables are either local (declared inside a macro) and are used only inside that macro or are global (declared outside a macro) and are used outside or inside any macro, depending on scope.

A literal string (actual ASCII text segment) is enclosed in quotation marks or apostrophes. To signify a quotation mark inside a literal string enclosed with quotation marks, a pair of quotation marks are used. When quotation marks are used to signify a literal string, apostrophes are treated as normal characters. To signify an apostrophe inside a literal string enclosed in apostrophes, a pair of apostrophes are used. When apostrophes are used to signify a literal string, quotation marks are treated as normal characters.

String variable names start with a letter while the remaining characters can be letters, numbers and the underscore. String variable names can be 2 to 31 characters in length. The **STRING** command has the following form:

STRING *name, name,... , name*

String variables cannot be initialized when declared. Only literal strings and/or other declared string variables can be assigned to the initialized string variable.

Example:
```
STRING WARNING,  Note1,  Message_no_2
Note1 =  "This is a string variable"
```

Strings and string variables are combined by using + character as follows:

Example:
```
STRING Note_1, Note_2, Note_3
Note_1 =  "A STRING"
Note_2 =  "COMBINING"
Note_3 =  Note_1 + " " + Note_2 + " EXAMPLE"
```

Note_3 now equals A STRING COMBINING EXAMPLE.

Segments are removed from strings using brackets in the following manner:

name1 = *name2*[*start*][*end*]

name1 is set equal to the segment in *name2* that begins with the *start* number and ends with the *end* number. Count of string *name2* begins with 0. String *name2* is not affected unless *name1* and *name2* are the same name.

Example:
```
STRING Test
$ = "Test1"
Test = $[4][4]
```

Test equals 1, the fourth (count begins with 0) character of string $.

Example:
```
STRING MEMO1
MEMO1 = "Message example"
MEMO1 = MEMO1[10][15]+MEMO1[0][3]
```

MEMO1 now contains the string "ample mess".

String Arrays are also declared using the **STRING** command by including the highest index in brackets as shown:

STRING *name*[*index*]

The index count starts with 0, making *index* one less than the actual number of variables in the array. String arrays cannot be initialized when declared. Only one string array element is assigned a value at a time.

Example:
```
STRING MESSAGE[5]
```

This command declares a string array named message containing six elements.

Example:
```
MESSAGE[0] = "see"
MESSAGE[1] = "spot"
MESSAGE[2] = "run"
```

This example assigns values to the first three elements of string array MESSAGE.

Substringing on string arrays allows segment selection from one or more elements in a string array. Adding a *start* and *end* number selects the segment in the *element* of the declared and initialized string array (*arrayname*). Segments are selected using the following format:

arrayname[*element*][*start*][*end*]

Example:
```
$ = MESSAGE[0][1][2]
```

This example assigns the string "ee" to $. ee is the segment in the first element [0], starting at character [1] and going through character [2].

There are several string functions that allow manipulations of strings:

CHR(n)

Returns the character equivalent of ASCII number n.

Example:
```
$=CHR(82)
```

$ is set equal to R (ASCII number of R is 82).

TAB(n)

Returns a blank string of length n spaces.

Example:
```
PRINT TAB(7),"FREQ = "
```

FREQ = is printed starting seven spaces from left edge of the CRT.

ASC($)

Returns ASCII value of the first character of a string.

Example:
```
VAR Ascii_value
STRING Char
Char="Example"
Ascii_value = ASC(Char)
```

Ascii_value, a numeric variable, contains 67 (ASCII number of E).

LEN($)

Returns the length of string $ in the number of characters.

Example:
```
Fred = "string"
B = LEN(Fred)
```

B, a numeric variable, contains 6 (the number of characters in "string").

STR(n)

Returns the string equivalent (character representation) of number n.

Example:
```
$ = STR(236)
```

$, a string variable, contains characters 236.

VAL($)

Returns the numeric value represented by string $.

Example:
```
STRING Commd
Commd = "567"
C = VAL(Commd)
```

C, a numeric variable, contains the number 567.

PIXLEN($)

Returns the length of a string in pixels (useful in centering text for the 1600S).

Example:
```
STRING Mode                 // Declare Mode string.
Mode = "Test Mode 4"        // Initialize Mode string.
P = PIXLEN(Mode)            // Assign pixel length of Mode to P.
XY ((640-P)/2)-1,40         // Set point for horizontal center.
Print Mode                  // Print Mode string.
```

P, a numeric variable, contains the number 160, the length of "Test Mode 4" in pixels. (See **PRINT** and **XY** commands in section 3-18.)

Returns the position of string $2 inside string $1. If string $2 is not found, -1 is returned. The count in string $1 begins with 0.

Example:
```
STRING One, Two
One = "preset"
Two = "set"
X = STRPOS(One,Two)
```

X is set to 3, "set" starts with the fourth character in "preset."

3-9 MACROS

Macros are programs, groups of commands arranged to accomplish one or more tasks. Macros are written and transferred to the Test Set by a Host System (PC with terminal emulating software). Once the macro is transferred to the Test Set, it is stored in memory (non-volatile for 1600S) until deleted by a ***PMC** or **FORGET** command or for CSA, power is removed. Once in memory, macros can be transferred to Flash Memory (see 4-14 for 1600S or 7-13 for CSA).

Macros are defined using the ***DMC** command (complying with IEEE-488.2) and have the form:

***DMC** "*name*", *command,... , command*

The name must start with a letter, the rest of the characters can be letters, numbers and the underscore. The maximum length is 31 characters (IEEE-488.2 allows only 12 characters).

Example:
```
*DMC "Square",Y=X**2;PRINT "Y holds X squared"
```

Once the macro is loaded into memory, it is executed by entering the macro name. If the above macro is loaded, entering Square executes the macro Square. Macros can be called out and executed from other macros. However, a macro cannot be transferred into memory if it contains the name of an undefined macro, unless the name is in an **INTERP** command.

Example:
```
*DMC "Test_1",BEGIN
Average
END
```

Example:
```
*DMC "Test_2",BEGIN
INTERP "Total"
END
```

If the macro Average is not already defined, the macro Test_1 cannot be transferred into memory. Macro Test_2 can be loaded into memory because the string Total is not evaluated until execution of the **INTERP** command. The macro Total has to be defined prior to Test_2 execution, otherwise an execution error will occur.

Call nesting is the extended process of calling out other macros that call out other macros that call out other macros, etc. Call nesting uses a limited stack to contain all the internal variables and arrays for each macro called out. Reaching the stack limit initiates a memory error.

With TMAC as opposed to IEEE-488.2, macros are not limited to one line. Using the **BEGIN** and **END** commands lets the macro contain multiple lines, allowing large involved macros, while also providing clarity that one line macros cannot afford.

Example:
```
*DMC "Square_x",BEGIN
    Y=X**2
END
```

A macro is deleted from memory using the **FORGET** command. All macros and variables declared after the deleted macro was declared, are also deleted (because they may contain references to the deleted macro). The ***PMC** command deletes all macros and declared variables in memory regardless of location. Predefined macros, listed in Appendix A, are permanently in memory and are not affected by the ***PMC** or **FORGET** commands. The ***LMC?** query (1600S only) returns a list of the macros and defined variables contained in memory.

The address of a macro (location of macro in memory) is assigned to a variable by using an ampersand in the following manner:

```
X=&name
```

The **EXEC** command executes the macro at the address following the command. It has the form:

EXEC *address*

Example:
```
ADD = &Square_x
EXEC ADD
```

The first command loads the address of Square_x into ADD. The **EXEC** command then executes Square_x.

The **EXEC** command can be used to call out undefined macros by simply declaring a variable and using the variable as the address.

(1600S only) The ***WAI** command pauses command execution until all previous operations are complete. ***WAI** is used following **SCREEN** commands and commands involving routing changes or long execution times. It has the form:

***WAI**

The size of the Test Set memory is the only limit to the number of macros that can be loaded. The **ROOM** command provides the available amount of memory space, in bytes.

Example:
```
PRINT ROOM
```

Prints the number of available bytes left in memory on the CRT.

Refer to 4-13 and 4-14 for 1600S specific **PROGram** and **MMEM** commands used in executing 1600S specific macros from the Front Panel.

Refer to Section 6 for loading CSA macros into Test Set memory for Front Panel execution.

3-10 CONDITIONAL EXPRESSIONS

TMAC allows decision points to be placed inside a macro to allow different courses of action. For instance, If a condition is true, then one course of action is taken. If the condition is false, then another course of action is taken. See **CASE**, **IF**, **IF ELSE** and **IF ELIF**. All conditions have relational operators:

=	Equal
!=	Not equal
<	Less than
>	Greater than
<=	Less than or equal
>=	Greater than or equal

Example:
```
IF x=6
   PRINT y
ENDIF
```

Condition x equals 6 must be true for the next command (PRINT y) to execute; otherwise it is ignored and command execution passes to the command following the **ENDIF** command.

Example:
```
IF x!=z
   PRINT y
ENDIF
```

Condition x != z (x not equal to z) must be true for the next command to execute. If it is not true, the command following the **ENDIF** is executed next.

Conditions may also contain logical operators AND and OR. These operators connect two conditions. An expression with an OR is true if one or both conditions it connects are true. An expression with an AND is true if both conditions it connects are true. There can be many logical operators in a condition.

Example:
```
IF x=6 OR x!=z
   PRINT y
ENDIF
```

This condition is true if either x is equal to 6, if x is not equal to z or if both are true.

Example:
```
IF x=6 AND x!=z
   PRINT y
ENDIF
```

This condition is true if both x is equal to 6 and x is not equal to z.

Example:
```
IF W=3 OR U< 8 AND V=4
   PRINT y
ENDIF
```

This condition is true if either W is equal to 3 or both U is less than 8 and V is equal to 4.

Mathematical operators can be used in conditions.

Example:
```
IF W> 2*X**3
```

The quantity 2*X**3 is calculated before compared to W.

Example:
```
IF T = U&++V
```

In this example, V is incremented and bitwise AND is performed before comparison with T is done.

Mathematical functions can also be used in conditions.

Example:
```
IF Y = SQR(COS(2*Freq))
```

In this example, the square root of the cosine of the quantity of 2 multiplied by Freq is calculated before the quantity is compared to Y.

Conditional expressions can be used without **IF** or other decision commands. Conditional expressions alone return a 1 if they are true and a 0 if they are false.

Example:
```
X=W<2*y**2
```

If W is less than $2 \times y^2$, X is set to 1. If W is greater than $2 \times y^2$, X is set to 0.

Example:
```
X=W=ABS(W)
```

If W is positive, W equals ABS(W) and X is set to 1. If W is negative, W does not equal ABS(W) and X is set to 0.

3-11 VARIABLES AND ARRAYS IN MACROS

Variables and arrays are passed to a macro when entering the macro by entering variables and arrays (parameters) after the name of the macro. Inside the macro, characters $1, $2, $3,...., $9 are placeholders for the passed parameters. Parameters may be passed to macros during execution.

Example:
```
*DMC "Compute",Y=$1**$2;PRINT $1,"TO THE POWER OF ",$2," IS ",Y
```

Entering `Compute 4,3` executes the macro Compute and sets $1 equal to 4 and $2 equal to 3. The CRT displays the result: `4 TO THE POWER OF 3 IS 64.`

Example:
```
VAR LIST[2]={1,3,5}

*DMC "Average",BEGIN
VAR SUM=0
FOR N=0 TO $1-1
SUM=SUM + $2[N]
NEXT N
PRINT "THE AVERAGE IS ",SUM/$1
END
Average 3,LIST
```

The macro Average reads an array and finds the average of the elements of the array. The length and name of the array is passed upon macro execution. When Average 3, LIST is entered, array LIST is passed using a **FOR** loop. The macro then prints the average of the 3 array values.

Example:
```
Add=&Compute
EXEC Add,4,3
```

A comma is placed before the variables when a macro is executed using the EXEC command. This example executes the macro Compute and has the same result as the first example.

The **INPUT** command allows data entry during macro execution. Data is entered using the Test Set DATA ENTRY Keypad (1600S) or the RS-232 terminal keyboard (CSA). After pressing the applicable ENTER key, variable *x* is given the value entered. For the 1600S, the **INPUT** command causes the CRT to display a blinking cursor. The **EDIT:WIDTH** command limits the width of the blinking cursor to *n* number of pixels. These commands have the form:

INPUT *x* **EDIT:WIDTH** *n*

Example:
```
*DMC "Compute",BEGIN
EDIT:COLOR:MENU 4          // Sets Input background color to Red.
EDIT:COLOR:LETTER 11       // Sets entered data color to Cyan.
EDIT:WIDTH 32              // Sets cursor width to 32 pixels.
INPUT X                    // Creates blinking cursor on CRT
                           // and sets X to entered value.
Y=$1**X
PRINT $1,"TO THE POWER OF ",X," IS ",Y
END
```

If Compute 5 is entered to execute the macro and 3 is entered using the DATA ENTRY Keypad (1600S), `5 TO THE POWER OF 3 IS 125` is printed.

3-12 MULTITASKING MACROS

Multitasking allows command execution to alternate between different macros prior to their completion. This allows complex macros to be separated into more manageable macros. Before a macro can share execution time with other macros, it must be declared a task using the **TASK** command and be put into the schedule queue using the **ACTIVATE** command. Multitasking requires the first macro in the schedule queue to also be the last macro to finish execution.

TASK "*name*"
Declares the named macro a task. Macro named must be in memory. A macro must be declared a task before it can be placed on the schedule queue.

ACTIVATE "*name*"
This command places the specified task on the schedule queue.

TPAUSE
This command passes command execution to the next macro or any Test Set activity in the schedule queue.

TSTOP
This command removes the currently executing task from the schedule queue and performs a **TPAUSE**.

SLEEP "*name*"
Takes the specified task out of the schedule queue.

WAKE "*name*"
Puts the specified task into the schedule queue and starts execution from point of last executed command.

Example:

```
*DMC "Main",BEGIN
ACTIVATE "First"
ACTIVATE "Second"
A=0
B=0
WHILE A=0 OR B=0
  TPAUSE
WEND
END
```

```
*DMC "First",BEGIN
FOR x = 1 to 10
  PRINT "FIRST"
  IF x=3
    A = 1
    TSTOP
  ENDIF
  TPAUSE
NEXT x
A = 1
END
```

```
*DMC "Second",BEGIN
n=1
FOR m=1 to 10
  PRINT "SECOND"
  IF n=2
    TPAUSE
    n=0
  ENDIF
    n=n+1
  NEXT m
  B = 1
END
```

```
TASK "Second"
TASK "First"
```

Executing the macro Main places Main in the schedule queue and then loads tasks First and Second into the schedule queue with Main. The macro Main fulfills the requirement of a macro to continue executing with its WHILE loop. This loop assures Main executes until First and Second have finished execution. The following sequence occurs:

- First begins execution, x is set to 1 and FIRST is printed. The **IF** command is skipped ($x \neq 3$), the **TPAUSE** command executes and command execution passes to Second.

- Second begins execution, n and m are set to 1 and SECOND is printed. The **IF** command is skipped ($n \neq 2$) and n is incremented to 2. Command execution loops to top of the **FOR** loop and m is incremented to 2. SECOND is printed and the **TPAUSE** command inside the **IF** command executes passing command execution to Main.

- Command execution begins again inside the **WHILE** loop. A and B are compared and the **TPAUSE** command is executed passing command execution to First.

- In First, command execution loops to the top of the **FOR** loop and x is set to 2. FIRST is printed, the **IF** command is skipped ($x \neq 3$) and the **TPAUSE** command passes command execution to Second.

- In Second, n is set to 0 and then incremented to 1. Command execution loops to the top of the **FOR** loop and m is set to 3. SECOND is printed, the **IF** command is skipped ($n \neq 2$) and n is incremented to 2. Command execution loops to the top of the **FOR** loop and m is incremented to 4. SECOND is printed and the **TPAUSE** command inside the **FOR** loop executes passing command execution to Main.

- Command execution begins again inside the **WHILE** loop. A and B are compared and the **TPAUSE** command is executed passing command execution to First.

- In First, command execution loops to the top of the **FOR** loop and x is set to 3. FIRST is printed, A is set to 1 and the **TSTOP** command inside the **FOR** loop executes taking First out of the schedule queue. Command execution passes to Second.

- In Second, n is set to 0 and then incremented to 1. Command execution loops to the top of the **FOR** loop and m is set to 4. SECOND is printed, the **IF** command is skipped ($n \neq 2$) and n is incremented to 2. Command execution loops to the top of the **FOR** loop and m is incremented to 5. SECOND is printed and the **TPAUSE** command inside the **FOR** loop executes with no result. Second continues to run in this manner until the loop is finished printing SECOND six more times. B is set to 1 and Second finishes execution returning command execution to Main.

- The **WHILE** loops conditional expression is satisfied ending the loop and the macro Main finishes execution.

3-13 CREATING WINDOWS AND GRAPHICS

TMAC allows creation of new screen displays and changes to CRT screen configurations. Colors can be changed. Windows can be created and moved. Pixels, boxes, lines and figures may be drawn and placed anywhere on the CRT screen. Audio cues can be implemented.

Some commands work in either the FM/AM-1600S or FM/AM-1600CSA. Other commands work only with one or the other as indicated.

3-13-1 COLORS

Table 3-4 lists the 16 colors, according to selection number, available for the CRT display.

COLOR	NUMBER	COLOR	NUMBER
Black	0	Dark Grey	8
Dark Blue	1	Blue	9
Dark Green	2	Green	10
Dark Cyan	3	Cyan	11
Dark Red	4	Red	12
Dark Magenta	5	Magenta	13
Brown	6	Yellow	14
Light Gray	7	White	15

Table 3-4 Colors and Color Selection Numbers

COLOR commands change the color selection for the CRT screen except for Soft Function Keys and menus which are changed by **EDIT:COLOR** commands (1600S only).

NOTE: Colors are entered using selection number or constant name (see Appendix A).

COLOR *f,b*
Changes the foreground color to the color selected for *f* and the character contrast color to the color selected for *b*.

COLOR?
Returns the current foreground color.

BCOLOR *b*
Changes the background color to the color selected for *b*.

NOTE: EDIT commands only apply to the FM/AM-1600S.

EDIT:COLOR:MENU *c*
Changes the menu color to the color selected for *c*.

EDIT:COLOR:MENU?
Returns the current menu color.

EDIT:COLOR:LETTER *c*
Changes the menu letter color to the color selected for *c*.

EDIT:COLOR:LETTER?
Returns the current menu letter color.

EDIT:COLOR:SOFT:BOX *c*
Changes the Soft Function Key background color to the color selected for *c*.

EDIT:COLOR:SOFT:BOX?
Returns the current Soft Function Key background color.

EDIT:COLOR:SOFT:LETTER *c*
Changes the Soft Function Key letter color to the color selected for *c*.

EDIT:COLOR:SOFT:LETTER?
Returns current Soft Function Key letter color.

EDIT:COLOR:SOFT:SELECT *c*
Changes the Soft Function Key selected color to the color selected for *c*.

EDIT:COLOR:SOFT:SELECT?
Returns the current selected Soft Function Key background color.

EDIT:WIDTH *n*
Sets the width of the blinking cursor that appears with the **INPUT** command to *n* pixels.

3-13-2 CRT DISPLAY

The display subsystem divides the CRT screen into pixels: 640 in the horizontal direction and 350 in the vertical direction. Window and graphic commands use these pixels to determine screen location. Screen locations are specified using a coordinate system with point 0,0 located in the top left corner of the CRT screen. The positive horizontal component is to the right and the positive vertical component is in the downward direction.

3-13-3 WINDOWS

Windows, used primarily for menus, can be created and moved. There can be up to 15 windows on a screen at one time. Windows are numbered in the order they are opened and must be closed in reverse order if overlapped. The last window opened is the selected window unless selection is changed by a **WSEL** command. Windows can be any size or color and can be moved to any location on the screen. Following are the window commands:

CLS
Clears everything from entire display screen (window 0) or currently selected window except for active readings.

SCREEN:USER (1600S) or USER (CSA)
Clears everything from active window. Displays blank user screen when active window is the default value of 0 (background screen). Decreases run time because of no readings to update.

WOPEN *c, x1, y1, x2, y2*
Opens a *c* colored window. The top left corner is located at point *x1, y1* and the bottom right corner is located at point *x2, y2.*

WMOVE *x, y*
Moves the window so the top left corner is located at point *x, y.*

NOTE: When a window is moved, parts of windows and graphics overlapping the moved window are moved also.

WCLOSE *n*
Closes *n* window. Range of *n* is 1 to 16 with *n*=16 closing all open windows.

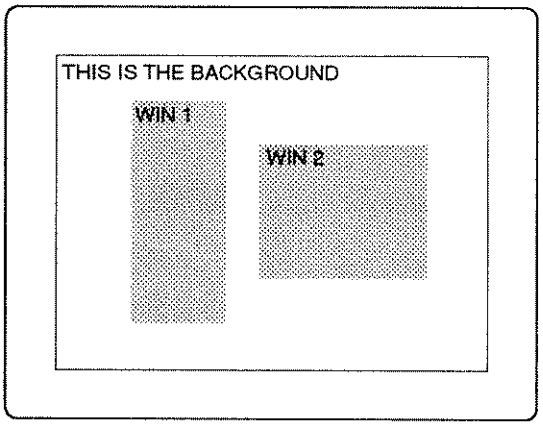
WSEL *n,h*
Makes window *n* the currently selected window. If *h* is set to 1, the window specified is hidden. To show the hidden window again, it must be closed and reopened. Selecting window 0 selects the background window.

WINDOW?
Returns the number of the last opened window (useful when closing or selecting windows).

Print commands executed while a window is open prints inside the currently selected window. To print outside the windows, select window 0 (background screen).

Example:
```
VAR Win1,Win2
CLS
WOPEN 10, 100,50,  250,300
Win1=WINDOW?
WOPEN 12,  300,100,  550,250
PRINT "WIN 2"
Win2=WINDOW?
WSEL Win1
PRINT "WIN 1"
WSEL 0
PRINT "THIS IS THE BACKGROUND"
```

These commands produce the following display:



8618010

Figure 3-1 Window Example

NOTE: The EGA control software forces the window locations and sites to the nearest multiple of eight (byte size). Therefore, it is a good idea to determine the location values prior to opening or moving a window.

3-13-4 SOFT FUNCTION KEY DISPLAYS

Soft Function Key definitions and the Soft Function Key frame at the bottom of the CRT screen are added using the following commands.

KEYPAD:SOFT
(1600S only) Creates a Soft Function Key frame at the bottom of the current screen displayed on the CRT.

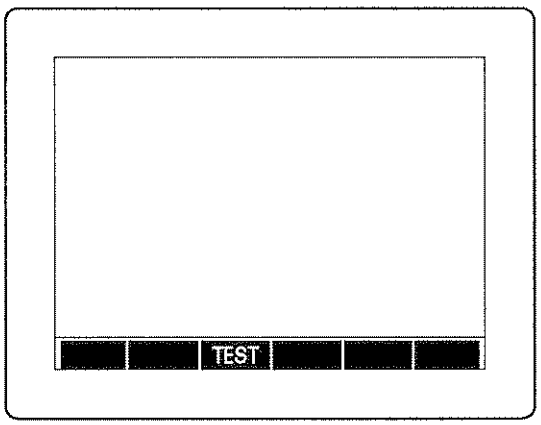
KEYPAD:LABel *n*,"*label*"
Changes the *label* of Soft Function Key *n* (1 to 6) on the current screen to the specified string.

NOTE: KEYPAD:LABel *n*,"" is equivalent to **KEYPAD:ERASE** *n*.

KEYPAD:ERASE *n*
(1600S only) Erases the label of Soft Function Key *n* (1 to 6) on the current screen.

Example:
```
CLS
KEYPAD:SOFT
KEYPAD:LAB 3," Test"
```

The following is shown on the Test Set CRT:



8618012

Figure 3-2 Soft Function Key Example

3-13-5 GRAPHICS AND TEXT

Graphics and text commands create text, pixels, lines, boxes and user defined shapes for display on the CRT:

XY *n,m*
Places a point location into memory with *n* being the x position (0 to 639) and *m* being the y position (0 to 349). 0,0 is the upper left hand corner of the CRT. **PIXEL**, **PRINT**, **HPRINT** and **ICON** commands start at the current xy position.

PIXEL
Displays a point on the CRT at the current xy location. A point is displayed in the current foreground color and does not change color if the foreground color changes. The point does not move if an **XY** command changes the point location in memory.

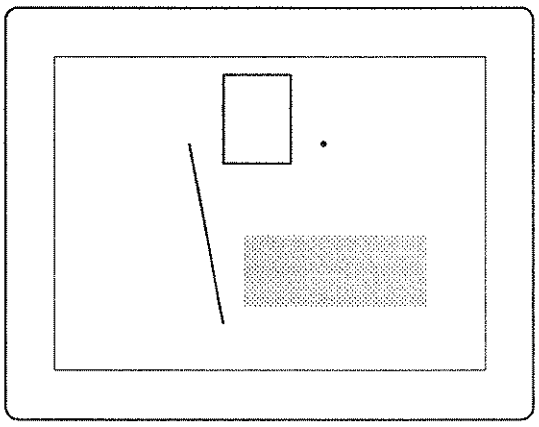
DRAW *x1, y1, x2, y2, c*
Draws a line from point *x1, y1* to point *x2, y2* in the color selected for *c.*

BOX *f, x1, y1, x2, y2, c*
Draws a box with the top left corner at point *x1, y1* and the bottom right corner at point *x2, y2.* Box is drawn in color selected for *c.* If *f* is set to 0, the box is empty. If *f* is set to 1, the box is created solid.

Example:
```
XY 400,100
COLOR  9,0
PIXEL
DRAW 200,100  250,300,  12
BOX 0,  250,20  350,120,  10
BOX 1,  280,200,  550,280,  13
```

This set of commands creates the following graphics displayed in Figure 3-3.

- Blue pixel at point 400, 100 and red line from point 200, 100 to point 250, 300.
- Hollow green box with top left corner at point 250, 20 and lower right corner at point 350, 120.
- Solid magenta box with top left corner at point 280, 200 and lower right corner at point 550, 280.



8618011

Figure 3-3 Graphics Examples

HEIGHT *n*
Changes the text height displayed on the CRT to *n* times the standard height (0.175 inches). *n* is 1,2,3 or 4 to provide text heights of 0.175, 0.35, 0.525 or 0.7 inches, respectively.

The following commands apply only to the FM/AM-1600S except as noted:

PRINT *%0nl, expression,expression, ...,expression,*
Displays information on the CRT, starting at the current xy position. An *expression* may be a variable (displayed as a value), string or characters and spaces designated within quotation marks. The *%0nl* sets the format for numeric values (see **PRINT** command in 3-18).

NOTE: For CSA operation, the **PRINT** command prints out the CSA RS-232 Connector. For CSA operation the **HPRINT** (host print) command prints on the CRT.

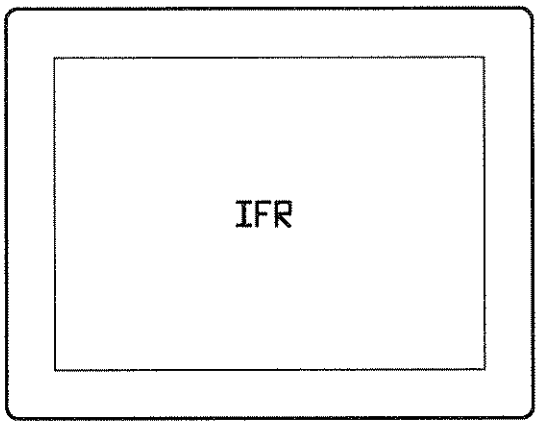
ELLIPSE *b,x,y,r,a,c*
Draws an ellipse centered at *x,y* with radius *r* and aspect ratio *a*. Ellipse is drawn in color selected for *c*. If *b* is 1, the ellipse is solid. If *b* is 0, the ellipse is hollow. An aspect ratio of 0.75 creates a circle. Parameters *x*, *y*, *r* and *a* are measured in pixels. The height and width of other ellipses are found using the following equations:

$a < 0.75$	Width = $0.2 \times r$	Height = $24 \times a$
$0.75 < a < 1$	Width = $2 \times r$	Height = $13.7\ a^2 + 16.3$
$a \geq 1$	Width = $r \div (5 \times a)$	Height = $1.25 \times r$

ICON *b,r,name*
Creates a user defined graphic displayed on the CRT. The CRT displays a predefined bit pattern with each bit represented by a pixel. Parameter *b* sets how many bits per row, *r* sets how many rows and *name* is a data array name which contains the preset bit pattern. The data array consists of 32 bit words. The graphics top left corner is the current xy position.

Example:
```
DATA bits={#hFFFFF3FF,#hFFCFFFFF,#hFFFFF3FF,#hFFCFFFFF,
           #hFFFFF3FF,#hFFCFFFFF,#hFFFFF3FF,#hFFCFFFFF,
           #h00F003C0,#h000F000F,#h00F003C0,#h000F000F,
           #h00F003C0,#h000F000F,#h00F003C0,#h000F000F,
           #h00F003FF,#hFC0FFFFF,#h00F003FF,#hFC0FFFFF,
           #h00F003FF,#hFC0FFFFF,#h00F003FF,#hFC0FFFFF,
           #h00F003C0,#h000F7E00,#h00F003C0,#h000F3F00,
           #h00F003C0,#h000F0F80,#h00F003C0,#h000F07E0,
           #hFFFFF3C0,#h000F01F8,#hFFFFF3C0,#h000F00FC,
           #hFFFFF3C0,#h000F003E,#hFFFFF3C0,#h000F001F}
XY 300,130
ICON 64,20,bits
```

The example creates the following image:



8618006

Figure 3-4 ICON Example

The following commands apply only to the FM/AM-1600CSA:

CENTER *element,x,y,pw*
Displays given text *element* (string or characters and spaces designated within quotation marks) in the center of the CRT display area defined by *x,y* position and pixel width, *pw.*

ERASE:TEXT *x,y,pw*
Erases the CRT display area defined by the *x,y* position and pixel width, *pw.*

HPRINT *%0nl, expression,expression, ...,expression,*
Displays information on the CRT, starting at current xy position. An *expression* may be a variable (displays value), string or characters and spaces designated within quotation marks. The *%0nl* sets the format for numeric values (see **HPRINT** command in 3-18).

LJPRINT *value,x,y,pw*
Erases the CRT display area defined by the *x,y* position and pixel width, *pw.* Displays the entered *value*, left justified, in the defined area on the CRT.

PIXLEN? *value*
Returns number of pixels required to display given *value.*

RJPRINT *value,x,y,pw*
Erases the CRT display area defined by the *x,y* position and pixel width, *pw.* Displays the entered *value*, right justified, in the defined area on the CRT.

XYPRINT *x,y,%0nl, expression,expression, ...,expression,*
Combines the **XY** and **HYPRINT** commands. Displays information on the CRT, starting at the selected *x,y* position. (see **XYPRINT** command in 3-18).

3-13-6 AUDIO TONES

Audio tones are generated using the **SOUND** command:

SOUND *f,t*
This command generates a tone of frequency *f* Hz for *t* ms. The tone is routed to the Test Set Speaker. There are two predefined macros, Chirp_1 and Chirp_2, that produce a short series of tones. Chirp_1 and Chirp_2 are detailed in Appendix A.

3-14 SYSTEM COMMANDS

3-14-1 SYSTEM KEY COMMANDS (1600S Only, except as noted)

SYSTEM:KEY commands assign Front Panel Keys to command strings that can execute macros. Command strings are limited to 80 characters. Only 16 keys can be assigned at one time. Appendix B lists Front Panel Keys with keycodes. Appendix A lists Front Panel Keys having predefined constants.

SYSTem:KEY:DEFine *n,sequence*
Assigns the specified command sequence to the key with keycode *n*. If the key is pressed, the command sequence is executed.

SYSTem:KEY:DELete *n*
Cancels the assignment of a key with keycode *n*.

SYSTem:KEY *n*
Simulates pressing a Front Panel Key with keycode *n*.

Example:
```
SYST:KEY:DEF 8200, Test_1 // Assigns macro Test_1 to the *
                          // DATA ENTRY Key.
SYST:KEY:DEL 8200         // Deletes the *-Test_1 assignment.
SYST:KEY 2112             // Simulates pressing RCVR MODE Key.
```

SYSTem:KEY?
Returns the keycode of the last key pressed or simulated pressed. If a key has not been pressed, -1 is returned.

KEY
Initiates a function to wait and provide the keycode of the next pressed key. Used with a **PRINT**, **PPRINT**, **HPRINT** or **XYPRINT** command.

NOTE: For CSA, the **KEY** command waits and provides the ANSI terminal keycode (usually ASCII) of the next key pressed on the terminal keyboard.

Example:
```
PRINT KEY
```

For 1600S: After a Front Panel Key is pressed, the keycode is printed on the CRT. For CSA: After a terminal key is pressed, the keycode is printed on the terminal monitor.

KEY?
Returns a 1 if a Front Panel Key is pressed, otherwise returns a 0.

NOTE: For CSA, the **KEY?** query returns 1 if a terminal key is pressed or 0, otherwise.

When a macro is running, the Test Set provides keys for both macro and front panel functions. To specify the Front Panel Keyboard input for TMAC use only, use the following commands:

KEYPAD:CLAIM
Directs all input from the Front Panel Keyboard to the TMAC Interpreter. Keyboard input not used for **INPUT** and **SYSTEM:KEY** commands is used once the keypad is unclaimed.

KEYPAD:UNCLAIM
Releases the Front Panel Keyboard for normal use.

3-14-2 SYSTEM ERROR COMMAND

The **SYSTem:ERRor** command reports system errors.

SYSTem:ERRor?
Returns the error number and description of the earliest error not yet read. Once read, the data for the error read is removed from memory. The earliest 16 errors that have not been read are held in memory with the rest being ignored. The 16 stored errors can be cleared from memory using the ***CLS** command.

3-14-3 SYSTEM DEFAULTS COMMAND (1600S Only)

SYSTEM:DEFaults
Restores the Test Set to the original factory default state.

3-14-4 SYSTEM PLOT COMMANDS (1600S Only)

SYSTem:PLOT:GPIB
Selects the GPIB Connector for plotter output.

SYSTem:PLOT:SERial
Selects the RS-232 Connector for plotter output.

3-14-5 SYSTEM TIME AND DATE COMMANDS

System commands allow setting and recording the date and time.

SYSTem:DATE?
Returns date in the form: year, month, day.

SYSTem:DATE *year, month, day*
Sets calendar to the entered date.

SYSTem:TIME?
Returns 24 hour time in the form: hour, minutes, seconds and milliseconds.

NOTE: Although 0.1 milliseconds are displayed, accuracy is assured to 16.5 milliseconds.

Example: `SYST:TIME?`

13,17,32.760 (1:17:32.760 PM) returns to the Host.

SYSTem:TIME *hour, minutes, seconds*
Sets the clock to the entered time.

3-14-6 SYSTEM RS-232 CONFIGURE COMMANDS

The following commands, applying only to the 1600S, allow the Test Set to control another device using serial RS-232 communication. Commands unique to the device controlled are sent as strings. Responses are received as strings.

SYSTem:PTHRough:SERial *"string"*
Passes the *string* through the RS-232 Connector.

SYSTem:PTHRough:SERial?
Waits for and returns a string from the RS-232 Connector.

SYSTem:PTHRough:SERial:QUEue?
Returns a non-zero value if data is in the RS-232 queue, 0 if the RS-232 queue is empty.

SYSTem:PTHRough:SERial:KEY?
Waits for and returns one character from the RS-232 Connector.

The following commands, applying to the 1600S and CSA, edit the RS-232 parameters remotely:

SYSTem:COMMunicate:SERial:BAUD *n*
Sets the RS-232 Connector baud rate to *n*. Select 300, 600, 1200, 2400, 4800, 9600 or 19200 or 38400 (CSA).

SYSTem:COMMunicate:SERial:PARity *type*
Sets the RS-232 Parity to *type*. Select NONE, EVEN, ODD, MARK or SPACE.

SYSTem:COMMunicate:SERial:BITS *n*
Sets the number of RS-232 Data Bits to *n*. Select 7 or 8.

SYSTem:COMMunicate:SERial:SBITs *n*
Sets the number of RS-232 Stop Bits to *n*. Select 1 or 2.

SYSTem:COMMunicate:SERial:PACE *type*
Sets the RS-232 Handshaking to *type*. Select XON (Xon/Xoff), HW (Hardware) or NONE.

SYSTem:COMMunicate:SERial:ECHO *b*
Sets the RS-232 Echo on if *b* is 1, off if b is 0.

Example:

```
*DMC "Control",BEGIN          // Defines macro to send query to
                              // controlled device & receive result.
STRING Result                 // Defines string to hold query result.
SYST:COMM:SER:BAUD 19200      // Sets RS-232 baud rate to 19200.
SYST:COMM:SER:PAR NONE        // Sets RS-232 parity to none.
SYST:COMM:SER:BITS 8          // Sets RS-232 data bits to 8.
SYST:COMM:SER:SBITS 1         // Sets RS-232 stop bits to 1.
SYST:COMM:SER:PACE XON        // Sets RS-232 handshaking to XON-XOFF.
SYST:COMM:SER:ECHO 1          // Sets RS-232 echo on.
SYST:PTHR:SER "query?"        // Sends string "query?" to device.
WHILE !(SYST:PTHR:SER:QUE?) // Loops while RS-232 queue is empty.
  TPAUSE                      // Allows Test Set operation while
                              // looping.
WEND                          // End of WHILE loop.
Result = SYST:PTHR:SER?       // Stores received string in Result.
END                           // End of macro Control.
```

3-14-7 SYSTEM SECURITY COMMANDS (1600S Only)

SYSTem:SECurity:STATe: *n*
Enables the Security System if *n* is 1 to 255, disables if *n* is 0.

SYSTem:SECurity:STATe?
Returns the Security setting. If not zero, Security is activated.

SYSTem:FREQuency:LOCK *b*
Enables the Receiver, Generator and Spectrum Analyzer RF lock feature if *b* is 1 or disables frequency lock feature if *b* is 0.

SYSTem:FREQuency:LOCK?
Returns the Security setting. If not zero, Security is activated.

SYSTem:KLOCk *b*
Locks the Test Set keyboard if *b* is 1. Unlocks the keyboard if *b* is 0 and System Security is deactivated. Security System must be enabled for keyboard to be locked. Pressing keys on the locked keyboard has no affect on the Test Set.

3-15 GPIB OPERATION

The FM/AM-1600CSA operates as a GPIB Talker/Listener device. The FM/AM-1600S operates as a GPIB Talker/Listener, Controller, Talk Only or Listen Only device. The GPIB Mode of Operation is selected using commands received through the RS-232 Connector, commands from a macro executed within the Test Set or from the Front Panel Auxiliary Functions Menu.

Commands are received and transmitted as strings on GPIB. Convert all commands to strings before transmitting. Precede non-printable characters with the \ character to transmit.

NOTE: Both the CSA and 1600S can operate on the same GPIB at the same time, using different addresses. Refer to Section 2.

3-15-1 OPERATING TEST SET USING GPIB COMMUNICATION (CSA Only)

The CSA is a GPIB Talker/Listener. The CSA Test Set address setting should not conflict with another device. The CSA provides a user defined Status Byte to generate Service Requests. The following commands are used when operating the CSA on the GPIB:

GPIB:ADDRess *a*
Sets CSA Test Set address to *a* (0 to 31).

GPIB:MASK *n*
Sets SRQ interrupt mask (eight bit value). The corresponding user defined Status Byte bit is masked unless the mask bit is active (set to 1). Range of *n* is 0 to 255.

GPIB:SRQ *n*
Sets a user defined Status Byte bit to trigger a Service Request (eight bit value). The corresponding mask bit must be active to generate an SRQ. Range of *n* is 0 to 255. The **GPIB:SRQ** command is used in decision loops (**IF**, **WHEN**, **FOR**, etc.) to trigger a Service Request when a user defined event, such as a cellular phone message capture, occurs.

Example:
```
GPIB:ADDR 30;MASK 32            // Set address and unmask bit 5.
FOCC:SET;CHAN 333;STAR          // Setup and monitor FOCC 333.
FOCC:CAPT:MIN "316/522-4981"  // Set to capture specified MIN.
IF FOCC:CAPT?=1 GPIB:SRQ 32;ENDIF // If capture occurs, send SRQ.
```

3-15-2 OPERATING TEST SET USING GPIB COMMUNICATION (1600S Only)

For the Test Set to become a GPIB Talker/Listener, Talk/Listen is selected as the GPIB Mode and the Test Set address is set to an address that does not conflict with another device. Both of these functions are performed remotely by executing the following command:

SYSTem:COMMunicate:GPIB:ADDRess *a*
Sets 1600S Test Set address to *a* (0 to 31) and selects Talk/Listen for GPIB Mode.

The following suggested procedures operate the 1600S Test Set using GPIB communication.

A. Initialization of the Test Set

- Prepare Controller for adding the 1600S Test Set to the GPIB.
- Clear the 1600S Test Set.
- Send "*ESE 1" command to Test Set. This command sets the event register of the Standard Event Status Register to 1 to allow reading of the operation complete bit.

B. Sending Commands to Test Set.

- Send "*SRE 36" command to the Test Set. This command sets the Service Request enable register to generate an RQS upon the occurrence of an OPC or Error.
- Send desired command with "*OPC" command following.
- Wait for RQS bit to be set (bit 6 of Status Byte) or for a time out.
- Retrieve Status Byte of Test Set and store response to value variable.
- If (value AND 4) is true, send error has occurred.
- If (value AND 4) is false and (value AND 32) is true, command executed.

C. Sending Queries to Test Set.

- Send "*SRE 20" command to Test Set. This command sets the Service Request enable register to report the Message Available (MAV) bit and error bit.
- Send query command to Test Set.
- Wait for RQS bit to be set (bit 6 of Status Byte) or for a time out.
- Retrieve Status Byte of Test Set and store response to value variable.
- If (value AND 4) is true, send error has occurred.
- If (value AND 4) is false and (value AND 16) is true, send read command to return results of query.

D. Sending Macros to Test Set.

Perform the following procedure until all command lines of macro are sent to Test Set.

- Send command line to Test Set.
- Wait for CMPL (completed I/O) or for a time out.
- If time out, stop sending of macro. Restart sending of macro (see *PMC in 3-17).

3-15-3 TEST SET OPERATING AS GPIB CONTROLLER (1600S Only)

For the Test Set to become a GPIB controller, the address of the Test Set must be set to an address that does not conflict with a slave device address (30 is advised) and the GPIB Operation Mode must be set to Controller. The commands used to set these parameters are:

SYSTem:COMMunicate:GPIB:ADDRess *a*
Sets 1600S Test Set address to *a* (0 to 31) and selects Talk/Listen for GPIB Mode.

SYSTem:COMMunicate:GPIB:CONTroller
Selects Controller for the GPIB Operation Mode making the Test Set a GPIB controller.

Once the Test Set is made a GPIB Controller, the talker/listener status is returned by repeating the **SYST:COMM:GPIB:ADDR** command. The following commands are used when operating the Test Set as a GPIB Controller:

SYSTem:COMMunicate:GPIB:SLAVe *a*
Sets the destination of **SYST:PTHR:GPIB** commands to the peripheral device with address *a* (0 to 31).

SYSTem:PTHRough:GPIB *address,"string"*
Sends out a *string* through the GPIB Connector. Optional peripheral *address* (0 to 31) defaults to the address set by the last **SYSTem:COMMunicate:GPIB:SLAVe** command.

SYSTem:PTHRough:GPIB? *address*
Waits for and receives a command at the GPIB Connector. Optional peripheral *address* (0 to 31) defaults to address set by last **SYSTem:COMMunicate:GPIB:SLAVe** command.

SYSTem:COMMunicate:GPIB:DCL
Issues a Device Clear message.

SYSTem:COMMunicate:GPIB:GET
Issues a Group Execute Trigger.

SYSTem:COMMunicate:GPIB:CMD *"string"*
Issues a command sequence through the GPIB Connector.

SYSTem:COMMunicate:GPIB:SRQ?
Checks peripheral devices for a Service Request. Returns a 1 if there was a Service Request since last check. Once read, the value is reset to 0 until next Service Request.

SYSTem:COMMunicate:GPIB:SPOLL? *a*
Performs a Serial Poll on the device with address *a* and returns the result. The 8 bit result is device dependent, except bit 6, which is the Service Request bit.

Examples:
```
*DMC "Control",BEGIN          // Defines a macro that sets the
SYST:COMM:GPIB:ADDR 30        // Test Sets address to 30 and
SYST:COMM:GPIB:CONT           // makes the Test Set a GPIB
END                           // Controller.
```

The macro Control establishes the Test Set as a GPIB Controller.

```
*DMC "Slave",SYST:COMM:GPIB:SLAV $1
```

The macro Slave sets the slave address for a peripheral device to $1.

```
*DMC "M1200",SYST:PTHR:GPIB 5,STR($1)+"\n"
```

The macro M1200 passes $1, as a string, through the GPIB Connector to the peripheral device with an address of 5. The "\n" provides a line feed and marks the end of the command passed.

```
*DMC "Read_1200",BEGIN
Control                          // Activates Control macro.
M1200 "DVMAC"                    // Sets 1200 to operate voltmeter.
WHILE SYT:PTHR:SER:QUE? =0       // While no Input from RS-232 Host
   IF SYST:COMM:GPIB:SRQ? =1     // If there is a Service Request,
    P=SYST:COMM:GPIB:SPOLL? 5    // Poll to see if it is from 1200.
       IF P&64 != 0              // If bit 6 is 1, SRQ is from 1200.
         M1200 "DVMRM"           // If SRQ is from 1200, take a
                                 // reading.
        ENDIF
   ENDIF
WEND
END
```

The macro Read_1200 assumes a FM/AM-1200S/A (1200) is connected to the GPIB Connector of the Test Set and has its address set to 5. The following sequence occurs:

- Makes the Test Set a controller.
- Passes the DVMAC command to the 1200. Sets the 1200 for autoranging voltmeter readings.
- While there is no Input from the RS-232 Host of the Test Set, continually checks for Service Requests from the 1200.
- Upon each 1200 Service Request, the Test Set passes the DVMRM command to the 1200. This command returns a voltmeter reading to the Host of the Test Set.

3-15-4 TEST SET OPERATING AS TALKER ONLY (1600S Only)

When the Test Set becomes a GPIB Talk Only device, there are no address considerations. The Test Set must be the only talking device on the GPIB bus and the remaining devices must be compatible listeners. The following command selects Talk Only as the GPIB Mode of Operation.

SYSTem:COMMunicate:GPIB:TONly
Selects Talk Only as the GPIB Operation Mode.

To select the Talk/Listen GPIB Mode of Operation again, reset the Test Set address using the **SYST:COMM:GPIB:ADDR** command.

3-15-5 TEST SET OPERATING AS LISTENER ONLY (1600S Only)

When the Test Set becomes a GPIB Listen Only device, the address is set first using the **SYST:COMM:GPIB:ADDR** command. After setting the address, the following command selects Listen Only as the GPIB Mode of Operation.

SYSTem:COMMunicate:GPIB:LONly
Selects Listen Only as the GPIB Operation Mode.

To select the Talk/Listen GPIB Mode of Operation again, reset the Test Set address using the **SYST:COMM:GPIB:ADDR** command.

3-16 STATUS SUBSYSTEM (1600S Only)

TMAC provides an implementation of the Event Status register structure as designated by IEEE-488.2 and SCPI Standards. Standard Event, Operation, Operation Instrument, Questionable, Instrument and Instrument Summary Status Registers and their relationship to one another are shown in Figure 3-6.

3-16-1 STATUS REGISTERS

Each Status Register shown in Figure 3-6 consists of a condition, event and enable register shown in Figure 3-5. The condition register continually receives data coming into the Status Register. When a bit is changed from 0 to 1 for the first time since being cleared, the same bit in the event register is set to 1. The event register is only changed by the first 0 to 1 transitions that occur in the condition register. The condition or event registers are cleared when read.

The 16 bits of the event register and the 16 bits of the enable register undergo a bitwise AND operation. In order for an event register bit to be reported, its corresponding bit in the enable register must be set to 1. The 16 bits resulting from this bitwise AND operation undergo an OR operation with each other. The result is 1 if one or more bits were 1.

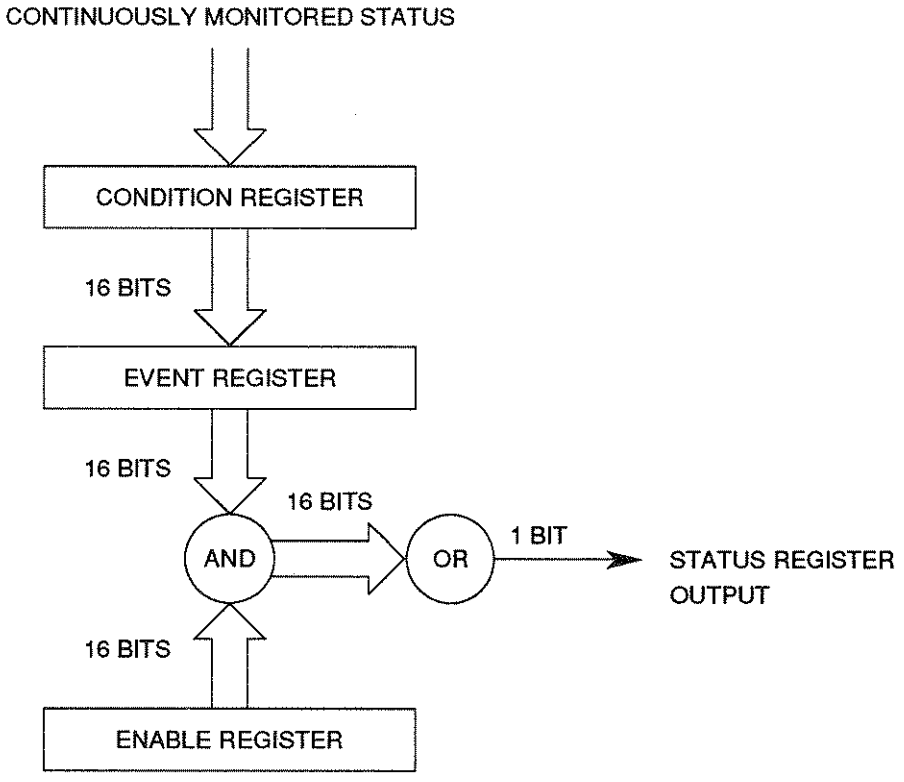


Figure 3-5 1600S Status Register Description

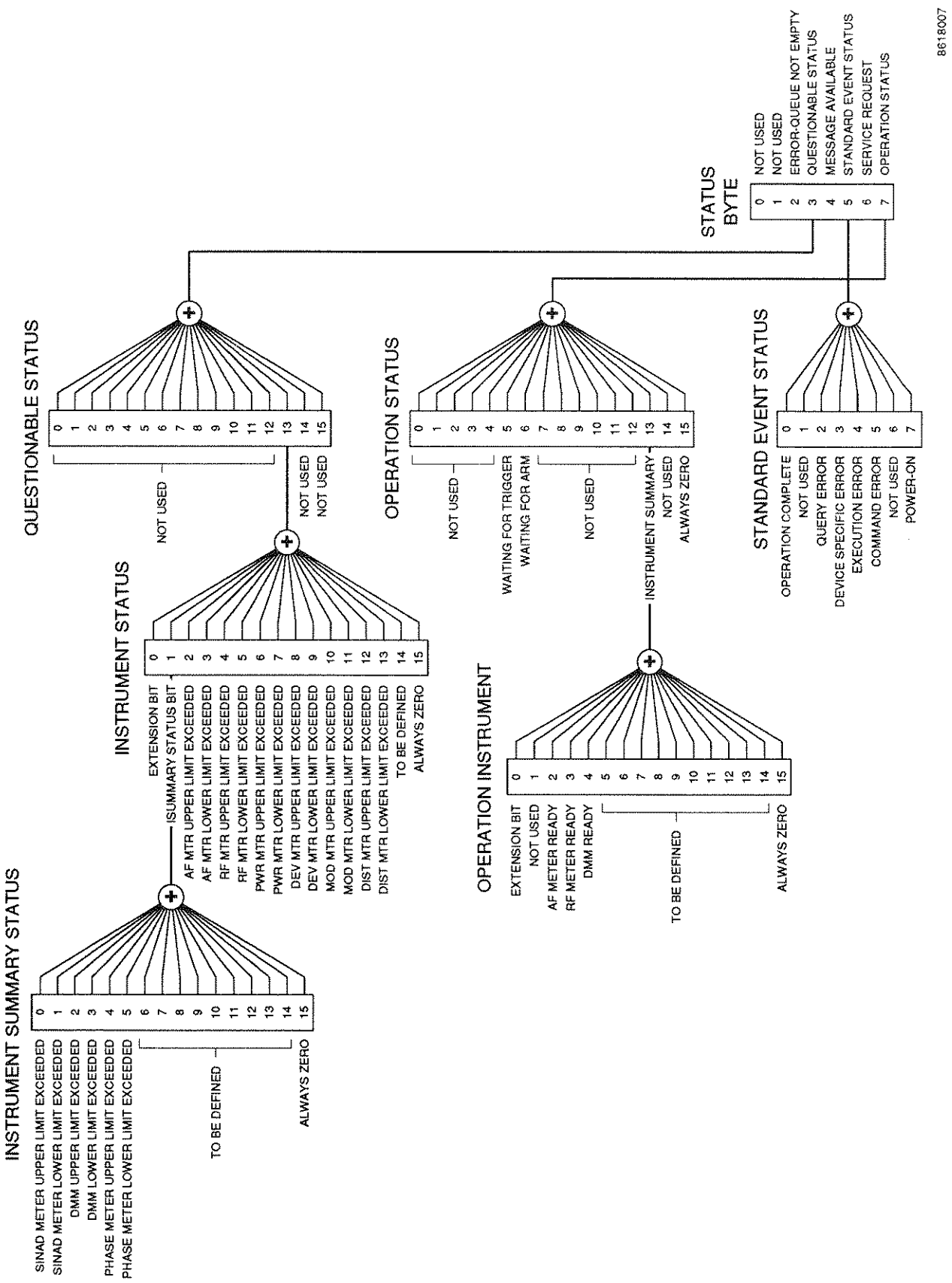


Figure 3-6 1600S Status Register Configuration

3-16-2 STATUS BYTE

The Status Byte reports an occurrence in any of the Status Registers. and is used to activate a SRQ message and the RQS bit (bit 6 of the Status Byte) (see Figure 3-6). A Status Byte bit determines an SRQ and RQS if the corresponding Service Request enable register bit is set to 1. Bits 0 to 5 and bit 7 of the Status Byte and the same bits of the Service Request enable register undergo a bitwise AND operation. The resulting bits of the AND operation undergo an OR operation. If any of the enabled Status Byte bits are 1, an SRQ is generated and the RQS Status Byte bit (bit 6) is set to 1.

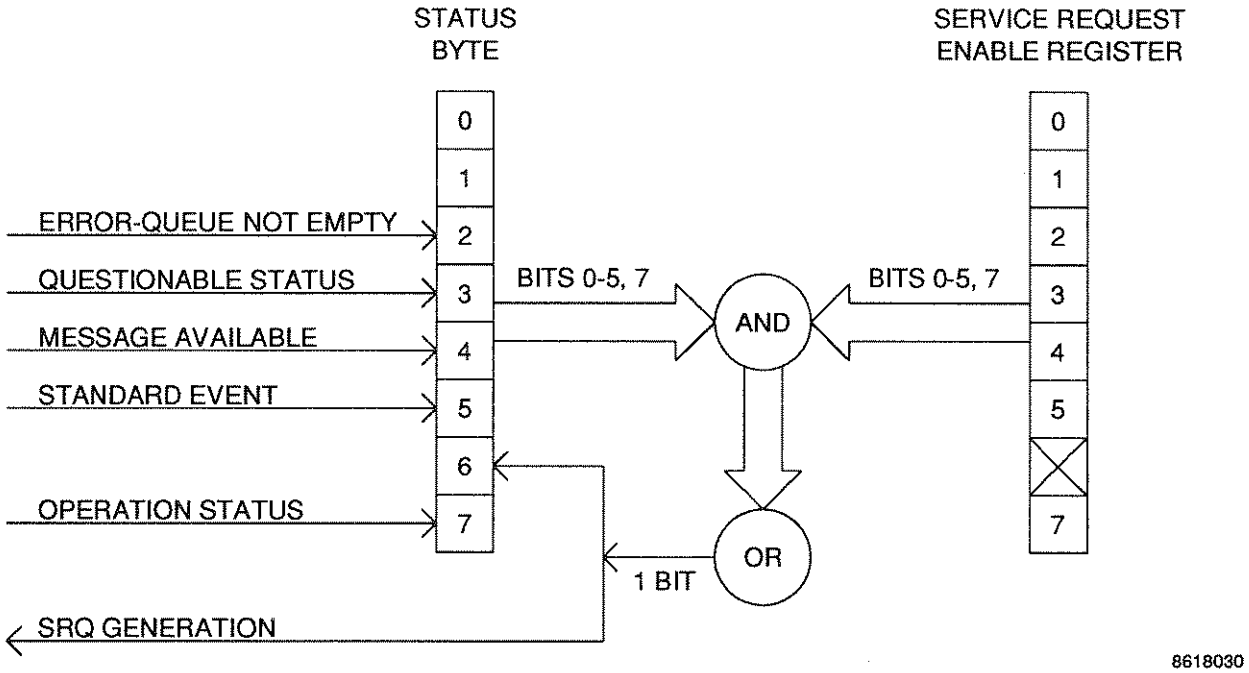


Figure 3-7 1600S RQS Bit and SRQ Generation

The following commands pertain to the Status Byte and Service Request enable register:

***SRE** *n*
Sets the Service Request enable register contents to *n* (eight bit value).

***SRE?**
Returns the value of the Service Request enable register contents.

***STB?**
Returns the value of the Status Byte. The Status Byte is not cleared when read but maintains the current value until the ***CLS** command is used (***CLS** command also clears the condition and event registers of each Status Register).

3-16-3 STANDARD EVENT STATUS REGISTER

The Standard Event Status Register contains an operation complete bit, several error bits and a power on bit. Figure 3-6 shows the Standard Event Status Register bits. The power on bit is set to 1 following power up of the Test Set. The following commands control this register:

***ESE** *n*
Sets the enable register of the Standard Event Status Register to *n* (eight bit value).

***ESE?**
Returns the enable register contents of the Standard Event Status Register.

***ESR?**
Returns the event register contents of the Standard Event Status Register.

***OPC**
Sets the Operation Complete bit of the event register to 1 when all pending selected device operations have finished.

***OPC?**
Generates an ASCII 1 in the output queue when operation is complete, activating the Message Available bit in the Status Byte.

Example:
```
*DMC "ERROR_CHECK",BEGIN   // Define macro named ERROR_CHECK.
*ESE 61                    // Enable Std Event Status Register.
IF (*ESR & 1)= 0           // Check Operation Complete bit.
  IF (*ESR? & 4)=1         // If Standard Event Status Register
    PRINT "QUERY ERROR"    // reports Query Error, print message.
  ELIF (*ESR? & 8)=1       // If Standard Event Status Register
                           // reports Device Specific Error,
    PRINT "DEVICE SPECIFIC ERROR" // print message.
  ELIF (*ESR? & 16)=1      // If Std Event Status Rgtr reports
    PRINT "EXECUTION ERROR" // Execution Error, print message.
  ELIF (*ESR? & 32)=1      // If Standard Event Status Register
    PRINT "COMMAND ERROR"  // shows Command Error, print message.
  ELSE                     // If Std Event Status Rgtr does not
    PRINT "NO ERROR DETECTED" // report an error type, print msg.
```

3-16-4 OPERATION STATUS REGISTER

The Operation Status Register contains the Waiting For Arm, Waiting For Trigger and Instrument Summary bits. The Operation Instrument Register sets the Instrument Summary bit (see 3-16-5). The following commands control the Operation Status Register:

STATus:OPERation:CONDition?
Returns the condition registers contents of the Operation Status Register.

STATus:OPERation:ENABle *n*
Sets the enable register contents of the Operation Status Register to *n* (16 bit value).

STATus:OPERation:ENABle?
Returns the enable registers contents of the Operation Status Register.

STATus:OPERation:EVENt?
Returns the event registers contents of the Operation Status Register.

3-16-5 OPERATION INSTRUMENT REGISTER

The Operation Instrument Register is used with the **INITiate** and **FETCh** commands. Once an **INITiate** command is completed and the meter is ready to be read, the appropriate meter ready bit is set to 1. This indicates the meter is ready for the **FETCh** command. The following **STATUS** commands control the Operation Instrument Register:

STATus:OPERation:INSTRument:CONDition?
Returns the condition register contents of the Operation Instrument Status Register.

STATus:OPERation:INSTRument:ENABle *n*
Sets enable register contents of Operation Instrument Status Register to *n* (16 bit value).

STATus:OPERation:INSTRument:ENABle?
Returns the enable register contents of the Operation Instrument Status Register.

STATus:OPERation:INSTRument:EVENt?
Returns the event register contents of the Operation Instrument Status Register.

3-16-6 QUESTIONABLE STATUS REGISTER

The Questionable Status Register reports events that could bring Test Set results into question. The Instrument Status Register provides the input. Figure 3-6 shows events reported by both registers. The following **STATUS** commands control the Questionable Status Register:

STATus:QUEStionable:CONDition?
Returns the condition register contents of the Questionable Status Register.

STATus:QUEStionable:ENABLe *n*
Sets the enable register contents of the Questionable Status Register to *n* (16 bit value).

STATus:QUEStionable:ENABLe?
Returns the enable register contents of the Questionable Status Register.

STATus:QUEStionable:EVENt?
Returns the event register contents of the Questionable Status Register.

3-16-7 INSTRUMENT STATUS REGISTER

The Instrument Status Register reports meter readings exceeding the upper and lower limit of the AF, RF Power, FM Deviation, AM Modulation and Distortion Meters. The results of the Instrument Summary Status Register (an extension of the Instrument Status Register) are also reported. The following **STATUS** commands control the Instrument Status Register.

STATus:QUEStionable:INSTRument:CONDition?
Returns the condition register contents of the Instrument Status Register.

STATus:QUEStionable:INSTRument:ENABLe *n*
Sets the enable register contents of the Instrument Status Register to *n* (16 bit value).

STATus:QUEStionable:INSTRument:ENABLe?
Returns the enable register contents of the Instrument Status Register.

STATus:QUEStionable:INSTRument:EVENt?
Returns the event register contents of the Instrument Status Register.

3-16-8 INSTRUMENT SUMMARY STATUS REGISTER

The Instrument Summary Status Register acts as an extension of the Instrument Status Register. The Instrument Summary Status Register reports the exceeding of the upper and lower limit of the SINAD and Phase Meter and Digital Multimeter. Following are **STATUS** commands controlling the Instrument Summary Status Register.

STATus:QUEStionable:INSTRument:ISUMmary:CONDition?
Returns the condition registers contents of the Instrument Summary Status Register.

STATus:QUEStionable:INSTRument:ISUMmary:ENABLe *n*
Sets the enable register contents of the Instrument Summary Status Register to *n* (16 bit value).

STATus:QUEStionable:INSTRument:ISUMmary:ENABLe?
Returns the enable registers contents of the Instrument Summary Status Register.

STATus:QUEStionable:INSTRument:ISUMmary:EVENt?
Returns the event registers contents of the Instrument Summary Status Register.

The following example performs a DMM reading and checks the Status Byte for the reporting of the DMM Upper or Lower Limit being exceeded. The enable registers of the Instrument Summary, Instrument and Questionable Status Registers are set to pass the information to the Status Byte.

Example:

```
*DMC "DMM_READ",BEGIN                 // Define macro named DMM_READ.
STAT:QUES:INSTR:ISUM:ENABLE 12        // Set enable register to read DMM
                                      // Upper and Lower Limit exceeded.
STAT:QUES:INSTR:ENABLE 2              // Set enable register to pass the
                                      // Instrument Summary Status
                                      // Register result.
STAT:QUES:ENABLE 8192                 // Set enable register to pass the
                                      // Instrument Status Register result.
*CLS                                  // Clear all condition and event
                                      // registers.
SCREEN:DMM                            // Display DMM Operation Screen.
PPRINT M_DMM?                         // Print DMM reading to Host.
IF (*STB? & 8) != 0                   // If Status Byte reports
                                      // Questionable Status bit as 1,
    PPRINT "limit exceeded"           // print limit exceeded to Host.
  ELSE                                // If Status Byte reports
                                      // Questionable Status bit as 0,
    PPRINT "limit not exceeded"       // print limit not exceeded to Host.
ENDIF                                 // End IF statement.
END                                   // End macro DMM_READ.
```

3-17 IEEE 488.2 COMPLIANCE COMMANDS

TMAC contains the following IEEE 488.2 mandated commands (1600S only except as noted):

***CLS**
(CSA also) Clear Status command. Sets all Status Registers including the Status Byte to zero.

***DDT**
Define Device Trigger command. Defines what commands are executed by the GET (Group Execute Trigger) command received at the GPIB Connector or the ***TRG** command.

Example: `*DDT SCOPE:TRIGGER:ONE;SCOPE:ARM`

These commands are executed when a GET (Group Execute Trigger) command is received at the GPIB Connector or a ***TRG** command is executed.

***DMC**
(CSA also) Define Macro command. Assigns commands to a macro label (see 3-9).

***EMC** *b*
Enable Macro command. Disables macros when *b* is set to 0, enables macros when *b* is set to 1. Used to execute commands which have the same name as a macro.

***EMC?**
Enable Macro Query. Returns the Enable Macro command status. 0 is returned if macros are disabled, 1 is returned if macros are enabled.

***ESE** *n*
Standard Event Status Enable command. Sets Standard Event Status Enable Register to value specified by *n* (see 3-16-3).

***ESE?**
Standard Event Status Enable Query. Returns the contents of the Standard Event Status Enable Register (see 3-16-3).

***ESR?**
Standard Event Status Register Query. Returns the contents of the Standard Event Status Register (see 3-16-3).

***IDN?**
(CSA also) Identification Query. Returns device identification. Manufacturer, model, serial number and version of device is returned.

***LMC?**
List Macro Query. Returns the currently defined macros (see 3-9).

***OPC**
Operation Complete command. Causes the device to activate the operation complete bit in the Standard Event Status Register when operation is finished (see 3-16-3).

***OPC?**
Operation Complete Query. Puts an ASCII 1 into the device output queue when operation is finished (see 3-16-3).

***PMC**
(CSA also) Purge Macro command. Deletes all macros and declared variables from memory (see 3-9).

***RCL** *n*
Recall command. Restores device to the environment (routings and settings) stored in local memory location *n*. Range of *n* is 0 to 9.

***RST**
Reset command. Resets device to preset condition.

***SAV** *n*
Save command. Saves current environment (routings and settings) in local memory location *n*. Range of *n* is 1 to 9.

***SRE** *n*
Service Request Enable command. Sets Service Request enable register to value specified by *n* (see 3-16-2).

***SRE?**
Service Request Enable Query. Returns contents of Service Request enable register (see 3-16-2).

***STB?**
Read Status Byte Query. Returns Status Byte contents (see 3-16-2).

***TRG**
Trigger command. Executes trigger command as defined by the ***DDT** command See **SCOPe:TRIGger:SOURce** (4-9) command.

***TST?**
Self-Test Query. Performs Self Test and returns 0 if passed, 1 if failed.

***WAI**
Wait To Continue command. Stops device command execution until ***OPC?** is true. Used following Screen commands to allow the Screen to mature before readings are taken.

3-18 FM/AM-1600S and FM/AM-1600CSA GENERAL TMAC COMMANDS

The following pages contain General TMAC commands; listing syntax, use and action. Examples are also provided to aid in the understanding of the commands. Cross references to relating commands are also included. Commands are in alphabetical order.

*DMC

USE

To define macros and functions.

SYNTAX

***DMC** "*name*", *command*; .... ; *command*

ACTION

Defines a macro or function with the *name* specified and containing the *commands* listed. After a macro or function is defined, entering the macros or functions *name* executes it (and all the *commands* contained within it).

Macros utilize several lines using the **BEGIN** and **END** commands. A function is a macro with a **RETURN** command used to return values. The **RETURN** command returns results directly through the active interface connector.

VARIABLE	DESCRIPTION
name	Name of the macro or function defined. The first character of the *name* is a letter while the remaining characters can be letters, digits and the underscore. The length of the *name* can be from 2 to 31 characters.
command	Commands performed when macro or function is executed.

SEE ALSO

Macros (3-9), **BEGIN**, **END**, **RETURN**, ***LMC**, **FORGET**, ***PMC**

EXAMPLES

```
*DMC "Sq",SCREEN:USER;:INPUT X;Y=X**2;PPRINT X," squared is ",Y
```

The above ***DMC** command defines these commands as a macro named Sq. Upon entering Sq, the **INPUT** command is executed, Y is set to X^2, and the **PPRINT** command is executed.

```
*DMC "Square",BEGIN
SCREEN:USER
INPUT X
Y=X**2
PPRINT X," squared is ", Y
END
```

The macro Square accomplishes the same as the Sq macro. The **BEGIN** and **END** command allows the macro to be written over several lines.

```
*DMC "SQ",RETURN($1**2)  // Defines a function named SQ.
PRINT SQ 11              // Prints the result of the function
                         // named SQ (121).

*DMC "Av",RETURN(($1+$2+$3)/4)   // Defines a function named Av.
X=Ave 13,14,17                   // Executes the function Av.
                                 // Returns 11.

*DMC "Sph_Sec_Surface",RETURN(3.1415*$1/2*(4*$2+$3))
PRINT Sph_Sec_Surface 4.4, 2.12, 6.7
```

The ***DMC** command defined the function (3.1415*$1/2*(4*$2+$3)) with the name Sph_Sec_Surface. 104.9135 is printed on the CRT.

*IDN?

USE To return Test Set Identification.

SYNTAX ***IDN?**

ACTION Returns the identification parameters of the Test Set. Includes the manufacturer, model, serial number and various firmware versions.

EXAMPLE

```
*IDN?
```

An example return from a query to an FM/AM-1600S is as follows:

```
IFR SYSTEMS,FM/AM-1600S,1583,3.02/C-03.01S/F-03.02S/M-02.00S
```

The serial number is 1583. The system firmware version is 3.02. The Counter firmware version is 3.01S. The Function Generator firmware version is 3.02S. The Monitor firmware version is 2.00S.

*PMC

USE To delete all macros and declared variables.

SYNTAX ***PMC**

ACTION Deletes all macros and declared variables except the predefined macros listed in Appendix A.

SEE ALSO Macros (3-9), **FORGET**

*WAI (1600S Only)

USE To pause command execution until previous operations are complete.

SYNTAX ***WAI**

ACTION Stops command execution until all operations are complete. ***WAI** is used after **SCREEN** and **SETUP** commands and other commands involving routing changes or lengthy measurement times.

SEE ALSO **DELAY**

EXAMPLE

```
SCREEN:REC          // Displays the Receiver Screen.
*WAI                // Stops command execution until
                    // the screen matures.
RECEIVE:INPUT:TR    // Selects T/R Connector for Receiver Input.
*WAI                // Stops command execution until
                    // the routing change is completed.
```

ACTIVATE

USE To place a task on the schedule queue so the task can be multitasked.

SYNTAX **ACTIVATE** "*name*"

ACTION A macro must be declared a task using the **TASK** command before it can be put in the schedule queue using the **ACTIVATE** command. This must be done to each macro before it can be multitasked. Once execution of the macro begins, execution starts from the beginning of the macro.

VARIABLE	DESCRIPTION
name	Name of the task to be loaded into the schedule queue.

SEE ALSO Multitasking Macros (3-12), **TASK**, **TPAUSE**, **TSTOP**

EXAMPLES See Multitasking examples (3-12).

BEGIN

USE To allow macros to contain multiple lines.

SYNTAX

***DMC** "*name*", **BEGIN**
sequence
END

ACTION Combines with the **END** command to define the command lines in multiple line macros. The **BEGIN** command is not used with single line macros.

VARIABLE	DESCRIPTION
name	Name of the macro being defined.
sequence	Command sequence performed inside the macro.

SEE ALSO Macros (3-9), ***DMC**, **END**

EXAMPLES

```
*DMC "MPt",VAR Mid;Mid=(Freq1+Freq2)/2;NVSAV Mid,60;PRINT Mid
```

The above macro can be entered as:

```
VAR Mid
*DMC "MPt",BEGIN
Mid=(Freq1+Freq2)/2
NVSAV Mid,60
PRINT Mid
END
```

BOX

USE To create a user defined box on the CRT.

SYNTAX **BOX** *f, x1, y1, x2, y2, c*

ACTION Creates a box using the points specified as the top left and bottom right corners. The box can be created hollow or solid. Text can not be printed in boxes as it can in windows.

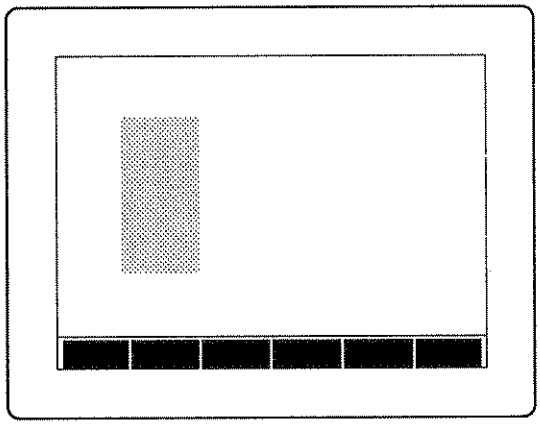
VARIABLE	DESCRIPTION
f	Creates the box solid or hollow. *f* is 1 for a solid box. *f* is 0 for the box to appear hollow.
x1, y1	Point signifying the top left corner of the box.
x2, y2	Point signifying the bottom right corner of the box.
c	Number specifying the color of the box. See Table 3-4 for the available colors and their numbers.

SEE ALSO Graphics (3-13-5)

EXAMPLES

```
BOX 1,150,50,200,250,7
```

The above command displays a light gray box on the CRT as follows:



8618021

Figure 3-8 Box Example

CALCULATE

USE To calculate a function and return the result to the Host.

SYNTAX **CALCulate?** *expression*

ACTION The *expression* is calculated and returned through the active interface connector. This command cannot be used inside a macro.

VARIABLE	DESCRIPTION
expression	Mathematical expression or formula calculated.

SEE ALSO Mathematical Functions (3-7)

EXAMPLES

```
X=4,Y=3                 // Initializes variables.
CALC? X*Y**3            // Calculates the mathematical
                        // expression and returns the result
                        // to the Host.  108 is returned.
```

CASE OF OTHERWISE ENDCASE

USE To provide a predetermined decision point inside a macro.

SYNTAX

CASE *variable*
  OF *value1*: *command*
  OF *value2*: *command*
  .
  .
  OTHERWISE: *command*
ENDCASE

ACTION There is no limit to the number of **OF** statements and the **OTHERWISE** statement is optional. The **OF** values are compared to the **CASE** *variable*. The *command* sequence following the **OF** value that equals the **CASE** *variable* is executed. If no **OF** value equals the **CASE** *variable*, the **OTHERWISE** *command* is executed if present, and macro execution passes to the command following the **ENDCASE** command.

VARIABLE	DESCRIPTION
variable	Expression compared with the values to decide which **OF** command to perform.
value	Expression compared with *variable*. If equal to *variable*, the command following *value* is performed.
command	Command performed if *value* is equal to *variable*.

SEE ALSO Conditional Expressions (3-10), **IF**, **IF ELSE**, **IF ELIF**

EXAMPLES

```
*DMC "Val"
SCREEN:USER
CASE X
  OF 1: PRINT "X=1"
  OF 2: PRINT "X=2"
  OF 5: PRINT "X=5"
  OTHERWISE:PRINT "X unknown"
ENDCASE
```

X is compared to the **OF** values 1, 2 and 5. If X is equal to one of the **OF** values, the commands following that **OF** value are executed. For example: if X=2, the **PRINT** "X=2" command is executed, if X=6, the **PRINT** "X unknown" command is executed.

```
* "Menu",BEGIN
SCREEN:USER
VAR Choice
PRINT "Press a DATA ENTRY Key to select a Test"
PRINT "     and press the ENTER Key."
PRINT "1. Test 1"
PRINT "2. Test 2"
PRINT "3. Test 3"
PRINT "Any other selection other than the ones listed"
PRINT "   displays the Receiver Operation Screen."
XY 100,250
INPUT Choice
CASE Choice
  OF 1: PRINT "Test_1 is running"
        DELAY 3000
        CLS
        Test_1
  OF 2: PRINT" Test_2 is running"
        DELAY 3000
        CLS
        Test_2
  OF 3: PRINT" Test_3 is running"
        DELAY 3000
        CLS
        Test_3
  OTHERWISE: PRINT "No Test Performed"
ENDCASE
SCREEN:REC
*WAI
END
```

The macro Menu displays a user created menu allowing the selection of one of three tests. These tests are actually macros Test_1, Test_2 and Test_3. which are executed if 1, 2 or 3 is entered in response to the **INPUT** command. If the entered data is not 1, 2 or 3, the Otherwise command is performed and `No Test Performed` is printed. After a Test is executed or the **PRINT** command is performed, command execution resumes beneath the **ENDCASE** command.

CLS

USE To erase the contents of a window or the CRT.

SYNTAX **CLS**

ACTION Clears the currently selected window. To clear the entire CRT, select window 0 (the background) using the **WSEL** command.

SEE ALSO **WSEL**

CONST

USE To define constants.

SYNTAX **CONST** *name, expression, name, expression,.... ,name, expression*

ACTION Declares constants and their values. Constants are global and can not be declared inside a macro. String data cannot be assigned to a constant. Constant values can only be changed using another **CONST** command.

VARIABLE	DESCRIPTION
name	Name of the constant defined. The first character of the *name* is a letter while the remaining characters can be letters, digits and the underscore. The length of *name* can be from 2 to 31 characters.
expression	Value the constant is defined as. The *expression* does not have to be in its simplest form.

SEE ALSO Constants and Data Arrays (3-4), **DATA**

EXAMPLES

```
CONST PI,3.1415,Wid,X/4.3  // Defines PI as a constant and
                           // gives it a value of 3.1415.
                           // Defines Wid as a constant,
                           // evaluates X/4.3 and assigns this
                           // value to Wid.
```

DATA

USE To define constant data arrays.

SYNTAX **DATA** *name*={*expression*, *expression*, ...,*expression*}

ACTION Declares data arrays and their values. A data array is an array of constants. Data arrays are global and can not be declared inside a macro. String data cannot be assigned to a data array. Data arrays values can only be changed using another **DATA** command. Data array values are specified by including the order of occurrence in brackets following the name. This order begins with 0 so that Stat[3] returns the fourth constant in the data array Stat.

VARIABLE	DESCRIPTION
name	Name of the constant defined. The first character of the *name* is a letter while the remaining characters can be letters, digits and the underscore. The length of the *name* can be from 2 to 31 characters.
expression	Value the constant is defined as. The expression does not have to be in its simplest form. The first *expression* is assigned to the first index (name[0]) and so on.

SEE ALSO Constants and Data Arrays (3-4), **CONST**, **ICON**

EXAMPLES

```
DATA STAT={2,34.7,2.874836593,336/4}
X=STAT[1]
Y=STAT[4]
Z=STAT[0]
```

The value of X is set to 24.7. The `Y=STAT[4]` command does not execute and an error is produced because STAT[4] is not defined. The value of Z is set to 2.

DELAY

USE To delay command execution.

SYNTAX **DELAY** *t*

ACTION Delays provide a time delay before the next command is executed. Delays are used to allow physical changes enough time to occur before command execution continues. Such changes could be: code generation, routing changes and displaying Operation Screens.

VARIABLE	DESCRIPTION
t	The length of time delayed in ms.

SEE ALSO Macros (3-9), ***WAI**

EXAMPLES

```
SCREEN:GEN
*WAI
GEN:DCS:INV 311
DELAY 2000
GEN:DCS:STOP
```

The **DELAY** command delays for 2 seconds to allow time for the DCS Code generation.

DO UNTIL

USE To perform a set of commands repeatedly until a specified condition occurs (becomes true).

SYNTAX

DO
sequence
UNTIL *condition*

ACTION The **DO** loop repeatedly executes a command sequence until a condition is true. If *condition* is true when first evaluated, the command sequence is executed once. If the *condition* does not eventually become true, the **DO** loop continues to loop until power is cycled.

VARIABLE	DESCRIPTION
condition	Conditional expression evaluated. If true, command execution loops to the top of the **DO** loop.
sequence	Command sequence performed at least once, then performed repeatedly until *condition* is true.

SEE ALSO **FOR**, **WHILE**

EXAMPLES

```
*DMC "Count",BEGIN    // Defines a macro named Count.
N=0                   // Sets N to 0.
DO                    // Starts the DO loop.
  N = N + 1           // Increments N by 1.
  PRINT N             // Prints current value of N.
UNTIL N=2             // Loops N to the DO command
                      // until N equals 2.
END                   // Ends macro Count.
```

N is set to 0 and the **DO** command is executed, initiating the following sequence:

- 1 is added to N and N is printed (1).
- N is compared to its ending value (1 compared to 2).
- 1 is added to N and N is printed (2).
- N is compared to its ending value (2 compared to 2).
- The command following the **UNTIL** command is executed as command execution continues on.

```
*DMC "Compute",RETURN(FLOOR(50-$1)/4+50)
                          // Defines a function named Compute.
*DMC "Compute_x",BEGIN    // Defines a macro named Compute_x.
X = 6                     // Sets X to 6.
DO                        // Starts the DO loop.
  X = Compute X           // Sets X to the result of the
                          // function Compute with the old
                          // value of X as the input.
PRINT X                   // Prints X.
UNTIL X=50                // Loops X to the DO command
                          // until X equals 50.
END                       // Ends the macro Compute_x.
```

After setting X to 6 and executing the **DO** command, the following sequence occurs:

- The Compute function is calculated and sets X equal to 61. X is printed.
- The conditional expression (X = 50) is evaluated and, if not true, command execution loops up to the **DO** loop.
- The Compute function is calculated and sets X equal to 48. X is printed.
- The conditional expression (X = 50) is evaluated and, if not true, command execution loops up to the **DO** loop.
- The Compute function is calculated and sets X equal to 50. X is printed.
- The conditional expression (X = 50) is evaluated and, if true, command execution continues below the **UNTIL** command.

The conditional expression does not have to involve a variable manipulated inside the loop.

```
DO
  commands
  .
  .
UNTIL SYSTEM:KEY? > 0
```

This **DO** loop continues looping until a Front Panel Key is pressed (for **SYSTEM** Commands, see 3-14).

DRAW

USE To create a user defined line on the CRT.

SYNTAX **DRAW** *x1, y1, x2, y2, c*

ACTION Creates a line from one point to the another point in a specified color.

VARIABLE	DESCRIPTION
x1, y1	Point signifying the first end of the line.
x2, y2	Point signifying the second end of the line.
c	Number specifying color of the line. See Table 3-4 for colors and their numbers.

SEE ALSO Graphics (3-13-5)

EXAMPLES See Graphic examples (3-13-5).

END

USE To allow macros to contain multiple lines.

SYNTAX
***DMC** "*name*", **BEGIN**
sequence
END

ACTION The **END** command defines the end of a multiple line macro.

VARIABLE	DESCRIPTION
name	Name of macro being defined.
sequence	Command sequence performed inside the macro.

SEE ALSO Macros (3-9), ***DMC**, **BEGIN**

EXAMPLES

```
* "Print_x",BEGIN        // Defines a macro named Print_x.
INPUT X                  // Receives a value from the
                         // keyboard.
PRINT X                  // Prints the received value.
END                      // End of the macro Print_x.
```

EXEC

USE To execute a macro at a specified address.

SYNTAX **EXEC** *address* or **EXEC** &*name*

ACTION Executes the macro located at the *address* specified. An & with the *name* of the macro as shown is equivalent to the address of the macro. Macros requiring variables to be passed when the macros *name* is entered cannot be executed using the **EXEC** command. Unpredictable results occur when using an invalid *address*.

VARIABLE	DESCRIPTION
address	Address of macro to be executed.
name	Name of the macro to be executed.

SEE ALSO Macros (3-9)

EXAMPLES

```
ADD=&First               // The address of the macro First
                         // is loaded into the variable ADD.
EXEC ADD                 // The macro located at the address
                         // loaded in the variable ADD is executed.

EXEC &First              // Macro First is executed.
```

FLUSH

USE To return pending responses.

SYNTAX **FLUSH**

ACTION Returning data from the queries in a macro is collected in the output queue of the Test Set. The contents of the output queue is returned at the end of any loops present, at the end of the macro or when the queue becomes full (the output queue is 2 kilobytes in size). The **FLUSH** command returns all pending return data upon execution.

SEE ALSO Macros (3-9)

EXAMPLES

```
*DMC "Without_FLUSH",BEGIN // Defines a macro named
                           // Without_FLUSH.
SCREEN:REC                 // Displays the Receiver Operation
                           // Screen.
*WAI                       // Waits for the Receiver Operation
                           // Screen to be completely displayed.
REC:FREQ?                  // Queries the Receiver Frequency.
M_RF?                      // Queries the Frequency Error Meter
                           // reading.
M_PWR?                     // Queries the Power Meter reading.
END                        // Ends macro Without_FLUSH.
```

The returned data in this macro without using the **FLUSH** command appears as follows:

```
100320,100318,62
```

```
*DMC "With_FLUSH",BEGIN  // Defines a macro named With_FLUSH.
SCREEN:REC               // Displays the Receiver Operation
                         // Screen.
*WAI                     // Waits for the Receiver Operation
                         // Screen to be completely displayed.
REC:FREQ?                // Queries the Receiver Frequency.
FLUSH                    // Returns impending responses.
M_RF?                    // Queries the Frequency Error Meter
                         // reading.
FLUSH                    // Returns impending responses.
M_PWR?                   // Queries the Power Meter reading.
END                      // Ends macro With_FLUSH.
```

The returned data in this macro with the **FLUSH** commands appears as follows:

```
100320
100318
62
```

FOR NEXT

USE

To perform a set of commands in a loop a given number of times within a macro.

SYNTAX

FOR *variable*= *initial* **TO** *ending* **STEP** *step*
sequence
NEXT *variable*

ACTION

The **FOR** command sets *variable* equal to *initial*, executes commands until the **NEXT** command, increments *variable* by the *step* value and starts command execution again just below the **FOR** command. The *sequence* between the **FOR** and **NEXT** command is executed repeatedly until *variable* is equal to *ending*. When *variable* equals *ending*, the *sequence* is executed once again before command execution proceeds below the **NEXT** command.

The values, *initial*, *ending* and *step*, can be negative. If optional **STEP** and *step* are left off, the default *step* is +1 even when *initial* and/or *ending* are negative. If step is set so *variable* will not equal ending exactly, the looping stops when *variable* comes the closest while not crossing the *ending* value. If *step* is given the opposite sign needed to increment *variable* towards the *ending* value or if *step* is larger than *ending* minus *initial*, the command sequence is executed once and no looping takes place.

Do not set *step* equal to 0 as this loops indefinitely and requires cycling power of the Test Set to stop execution of the loop.

VARIABLE	DESCRIPTION
variable	*variable* used to step through the looping process.
initial	Starting value of *variable*.
ending	Ending value of *variable*. When *variable* reaches this value, looping is stopped and the *command sequence* is executed once more.
step	Amount *variable* is incremented for each pass through the loop.
sequence	Command sequence is executed once and then repeatedly until *variable* equals *ending*.

SEE ALSO

DO, WHILE

EXAMPLES

```
FOR N=1 TO 3 STEP 1
  command1
NEXT N
command2
```

The following sequence occurs:

- N is set equal to 1.
- Command1 is executed.
- N is incremented by 1 (N=2) and compared to *ending* (3).
- Command1 is executed.
- N is incremented by 1 (N=3) and compared to *ending* (3).
- Command1 is executed.
- Command2 is executed and macro execution proceeds after command2.

```
FOR A=10 TO 4 STEP -3
  PRINT A
NEXT A
PRINT 2*A
```

The following sequence occurs:

- A is set equal to 10.
- A is printed (10).
- -3 is added to A (A = 7) and A is compared to *ending* (4).
- A is printed (7).
- -3 is added to A (A = 4) and A is compared to *ending* (4).
- A is printed (4).
- 2*A is printed (8) as command execution proceeds below the **NEXT** command.

```
FOR MM = -1 TO -6 STEP -2
  PRINT MM
NEXT MM
```

-1, -3 and -5 is printed. Since MM=-7 is beyond the ending value, the last loop occurs when MM is equal to -5.

```
FOR N=1 TO 4 STEP -1
  PRINT N
NEXT N
```

N is set equal to 1 and printed once and command execution proceeds below the **NEXT** command. No looping takes place because *step* does not have the sign needed to proceed the *variable* value toward the *ending* value.

FORGET

USE To delete a macro from memory.

SYNTAX **FORGET** "*name*"

ACTION Deletes the macro specified. All macros and variables declared after the specified macro was declared are also deleted.

VARIABLE	DESCRIPTION
name	Name of macro to be deleted.

SEE ALSO ***PMC**

EXAMPLES

```
*DMC "FIRST",BEGIN
  PRINT "first macro"
END
*DMC "SECOND",BEGIN
  PRINT "second macro"
END
VAR TO_BE_ERASED

FORGET "SECOND"
```

Upon execution of the **FORGET** command, the macro SECOND and the variable TO_BE_ERASED are deleted from memory. The macro FIRST is still contained in memory.

FORMat

USE To specify the form of the returned data for queries.

SYNTAX
FORMat BINary
FORMat OCTal
FORMat HEXadecimal
FORMat ASCii (1600S Only)
FORMat DECimal (CSA Only)

ACTION Specifies form of the returned data for queries. The four forms are Binary (base 2), Octal (base 8), Hexadecimal (base 16) and ASCII (1600S) or Decimal (CSA) (base 10). Default is base 10. For the 1600S, all returned data not in ASCII format is preceded with a prefix signifying the format: #B for Binary, #Q for Octal and #H for Hexadecimal. Data in scientific notation is not affected by these commands.

SEE ALSO Numeric Form (3-5-2)

EXAMPLES

```
REC:FREQ 103.7 MHZ     // Sets Receiver Frequency to 103.7 MHz.
FORM BIN               // Selects binary format for the
                       // returned data.
REC:FREQ?              // Queries the Receiver Frequency.
                       // #B11001010100010100 is returned.
FORM HEX               // Selects Hexadecimal format for
                       // the returned data.
REC:FREQ?              // Queries the Receiver Frequency.
                       // #H19514 is returned.
```

HEIGHT

USE To change height of text printed on CRT.

SYNTAX **HEIGHT** *n*

ACTION The height of text printed on the CRT after this command is changed to *n* times the standard height (0.175 inches). Text can be printed in four heights: 0.175, 0.35, 0.525 or 0.7 inches.

VARIABLE	DESCRIPTION
n	Height parameter setting height n x the standard height. *n* is an integer from 1 to 4.

SEE ALSO Graphics (3-13-5), **PRINT**, **ROTATE**

EXAMPLES

```
HEIGHT 1                      // Sets the text height to 0.175''.
PRINT "standard"              // PRINT command.
PRINT                         // Each PRINT command moves the xy
PRINT                         // position down a line the current
PRINT                         // height set by the HEIGHT command
                              // (currently 0.175'').
HEIGHT 4                      // Sets the text height to 0.7''.
PRINT "large"                 // PRINT command.
HEIGHT 2                      // Sets the text height to 0.35''.
PRINT "double standard"       // PRINT command.
HEIGHT 1                      // Returns the HEIGHT setting to 1.
```

The following is printed on the CRT: `standard` in 0.175 inch high text, `large` in 0.7 inch high text and `double standard` in 0.35 inch high text.

Each **PRINT** command without an ending comma moves the xy position to the beginning of the next line using the height setting for the height of the line. The first four **PRINT** commands move the xy position down 0.7" (0.175" for each PRINT command). The **PRINT** "large" command moves the xy position down 0.7". The **PRINT** "double standard" command moves the xy position down 0.35".

NOTE: The **HEIGHT** setting should be returned to 1 because all printing on the CRT uses this setting.

HFLUSH (CSA Only)

USE To quickly send **HOST** commands through the SCSI Bus.

SYNTAX **HFLUSH**

ACTION Gives previous **HOST** commands priority by flushing all commands currently in the task queue out the SCSI at that time. (Normally, to avoid time consuming SCSI traffic, **HOST** commands are held in the task queue. When the task queue becomes full, commands are sent across the SCSI Bus to the 1600S.)

EXAMPLES

```
host "fgen:gen1:mod1 5;shape:sin"
host "fgen:gen1:freq 1000;mod:fm
hflush
```

HPRINT (CSA Only)

USE

To print on the CRT screen.

SYNTAX

HPRINT *%0nl, expression, ... ,expression,* or
HPRINT *expression, ... ,%0nl, expression,*

ACTION

The **HPRINT** command (**HOST PRINT** command) prints numerical and string values at the current xy location on the CRT. Several values are printed on the same line by separating them with commas. If the optional ending comma is used, the next **HPRINT** command continues on the same line, otherwise a new line is started. If an expression is a variable, the value of the variable is printed. Mathematical expressions are calculated before printing.

The *%0nl* is the format settings for numeric expressions and can be placed anywhere in the **HPRINT** command. There can be several format settings in each **HPRINT** command, each one changing the leading zero format, the numeric base and the field width for the expressions that follow. Format settings do not affect the printing of strings.

VARIABLE	DESCRIPTION
0	Zero signifies leading zeros are added to fill the field width. The *0* is omitted for no leading zeros. Applies only to numeric expressions.
n	Specifies field width. *n* is the minimum number of digit spaces (decimal point counts as one). Applies only to numeric expressions. If *l* is D, *n* can be in the form of *b.a*, where *b* designates minimum number of total digit spaces (decimal point counts as one) and *a* designates the maximum number of digits after the decimal point.
l	Specifies the numeric base the data is printed in. Applies only to numeric expressions. Omit for a default of Signed Decimal or enter one of the following: B Binary Q Octal H Hexadecimal U Unsigned decimal D Signed decimal
expression	Expression to be printed. Can be a mathematical expression, a literal string or the contents of variables. Mathematical expressions are calculated and string functions are performed before the results are printed.

SEE ALSO

PRINT, XYPRINT, PPRINT, PSCREEN, XY

EXAMPLES

```
HPRINT "CELLULAR PHONE TEST"
```

This **HPRINT** command prints the string text, between the quotation marks, on the CRT, starting at the current xy location.

```
HPRINT "The Receiver Frequency is ",%3.0,HOST "REC:FREQ?"
```

No leading zeros and a field width of 3 is specified after the string text is printed. The Receiver frequency setting is printed with the field width of 3, rounded to the nearest kHz because no digits are printed after the decimal point.

ICON (1600S Only)

USE To create a user defined graphic on the CRT.

SYNTAX **ICON** *b*, *r*, *name*

ACTION A graphic is created by defining an area of pixels and a data array specifying which pixels to appear and which not to appear (1's to appear, 0's not to appear). The top left corner of the defined area is placed at the current **XY** setting. The data array must contain 32 bit words.

VARIABLE	DESCRIPTION
b	Number of bits per row in the icon to be generated.
r	Number of rows in the icon to be generated.
name	Name of data array containing the bit pattern used for the icon.

SEE ALSO Graphics (3-13-5), **XY**, **DATA**

EXAMPLES See Graphic examples (3-13-5).

IF ENDIF

USE

To provide a decision point inside a macro.

SYNTAX

IF *condition sequence*; **ENDIF**

or

IF *condition*
sequence
ENDIF

ACTION

The condition must be true for *sequence* to execute. If the condition is not true the *sequence* is not executed and execution flow passes to the next command after the **ENDIF** command. Every **IF** command must have an **ENDIF**. **IF** commands can be put inside other **IF** commands if the **ENDIF** to the inner **IF** command is also inside the outer **IF** command.

VARIABLE	DESCRIPTION
condition	Conditional expression evaluated to determine which command sequence is executed.
sequence	Command *sequence* performed if the condition is true.

SEE ALSO

IF ELSE, IF ELIF, CASE

EXAMPLES

```
IF Mod>20 Flag=1;ENDIF
```

This set of commands sets Flag equal to 0 if Mod is greater than 20. If Mod is not greater than 20, this set of commands has no effect, as the Flag=0 command is not performed.

```
IF Dev > 3
  IF Freq_Err < 1000
    Freq=200000
  ENDIF
ENDIF
```

The Inner **IF-ENDIF** commands are executed if Dev is greater than 3. The Freq=200000 command is executed if Freq_Err is less than 1000 and Dev is greater than 3.

IF ELIF ELSE ENDIF

USE

To provide a decision point inside a macro.

SYNTAX

IF *condition*
  sequence
ELIF *condition*
  sequence
ELIF *condition*
  sequence
  .
  .
ELSE
  sequence
ENDIF

ACTION

IF ELIF command allows several *conditions* to be checked instead of one or two. Each **ELIF** (Else If) part allows the checking of a *condition*. If the *condition* is true, the following *sequence* is executed. The optional **ELSE** *sequence* is executed if no **ELIF** *condition* is true. An **ENDIF** command must be included at the end of a **IF ELIF** group of commands. If two **ELSE IF** commands are true, only the *sequence* following the first one is executed.

VARIABLE	DESCRIPTION
condition	Conditional expression
sequence	Command sequence (set of commands) performed if previous conditional expression is true.

SEE ALSO

IF, IF ELSE, IF ELIF, CASE

EXAMPLES

```
IF X=3
  PRINT "X=3"
ELIF X=4
  PRINT "X=4"
ELIF X=5
  PRINT "X=5"
ENDIF
```

Executes the **PRINT** command that follows a true *condition* if there is one and ignores all other **PRINT** commands.

IF ELSE ENDIF

USE

To provide a decision point inside a macro.

SYNTAX

IF *condition sequence1*
ELSE *sequence2*
ENDIF

or

IF *condition*
sequence1
ELSE
sequence2
ENDIF

ACTION

IF ELSE command allows one of two command sequences to be executed depending on the validity of the *condition*. If the *condition* is true, *sequence1* is executed. If the *condition* is false, *sequence2* is executed. Every **IF ELSE** command must have an **ENDIF**.

VARIABLE	DESCRIPTION
condition	Conditional expression that determines which command sequence is executed.
sequence1	First command sequence. *sequence1* is executed if *condition* is true.
sequence2	Second command sequence. *sequence2* is executed if *condition* is not true.

SEE ALSO

IF, IF ELIF, CASE

EXAMPLES

```
IF D>4 PRINT "D HIGH"
ELSE PRINT "D LOW"
ENDIF
```

Example prints D HIGH if D is greater than 4; otherwise D LOW is printed.

```
IF T=3
  T=T+1
ELSE
  T=T-1
ENDIF
```

1 is added to T if T is equal to 3. If T is not equal to 3, 1 is subtracted.

IF ELSE (Shorthand)

USE As an abbreviated version of the **IF ELSE** command.

SYNTAX *command* (*condition*? *true*:*false*)

ACTION The shorthand version of the **IF ELSE** command allows one of two arguments to be performed with the same *command*. If the *condition* is true, the first argument (*true*) is performed. If the *condition* is false, the second argument (*false*) is performed. No **ENDIF** is used with this command.

VARIABLE	DESCRIPTION
command	Incomplete command which is performed with one of the arguments.
condition	Conditional expression that determines which argument is performed with the command.
true	Argument *command* performs if *condition* is true.
false	Argument *command* performs if *condition* is false.

SEE ALSO **IF, IF ELSE, IF ELIF, CASE**

EXAMPLES

```
PRINT (D>4? "D HIGH":"D LOW")
```

The conditional expression (D>4?) is evaluated, and if true, the command PRINT "D HIGH" is executed. If the conditional expression is false, the command PRINT "D LOW" is executed.

```
T= (T=3? T+1:T-1)
```

The conditional expression (T=3?) is evaluated, and if true, the command T=T+1 is executed. If the conditional expression is false, the command T=T-1 is executed.

INPUT

USE To enter data during macro execution.

SYNTAX **INPUT** *variable*

ACTION Stops execution of the macro until data is input. The value of the data entered is assigned to the *variable* specified in the command and command execution continues. For the 1600S, the CRT displays a blinking cursor with a width set using the **EDIT:WIDTH** command. Data is entered using the DATA ENTRY Keys and pressing the ENTER Key. For the CSA, there is no blinking cursor and the data is entered using the remote controller keyboard.

VARIABLE	DESCRIPTION
variable	Variable given the value entered.

SEE ALSO Variables and Arrays in Macros (3-11), Colors (3-13-1)

EXAMPLES

```
*DMC "GET_X"
EDIT:WIDTH 30
INPUT X
PRINT X
END
```

Upon execution of the macro GET_X, a blinking cursor appears on the CRT (1600S only). The cursor is set to a width of 30 pixels by the **EDIT:WIDTH** command. If 6, 5, • and 3 DATA ENTRY Keys are pressed, variable X is assigned the value of 65.3 and `65.3` is printed on the CRT.

INTERP

USE To execute commands that may not be known when the macro containing the commands is defined.

SYNTAX **INTERP** "*string*;*string*;.. ;*string*"

ACTION The **INTERP** command executes each *string* as a command and is limited to one line. The **INTERP** command is not required to be inside a macro.

VARIABLE	DESCRIPTION
string	String variables, literal strings and string functions.

EXAMPLES

```
*DMC "Run_tests",BEGIN
INPUT $
INTERP "PRINT $; TEST"+STR($1)
END
```

Entering Run_tests 4 executes this macro. If Test Working is entered at the INPUT prompt, Test Working is printed and the macro Test4 is executed. If the **INTERP** command is not used, the Run_tests macro can not be defined because the $ in the **PRINT** command and TEST4 are not yet defined.

KEYPAD:ERASE (1600S Only)

USE To erase Soft Function Key definitions.

SYNTAX **KEYPAD:ERASE** *n*

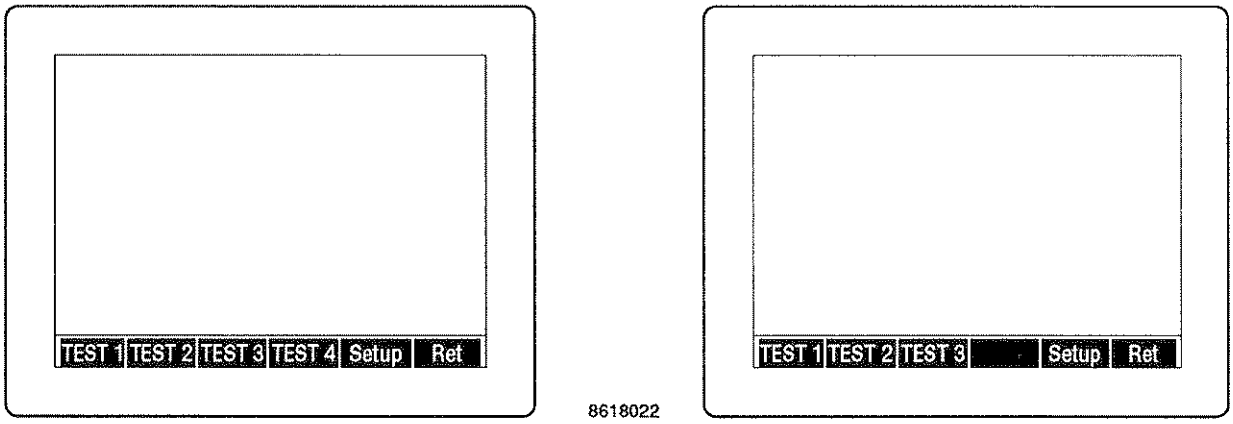
ACTION Erases definition for *n* Soft Function Key.

VARIABLE	DESCRIPTION
n	Number of the Soft Function Key affected. The range of *n* is from 1 to 6.

SEE ALSO Soft Function Key Displays (3-13-4), **KEYPAD:LABEL**, **KEYPAD:SOFT**

EXAMPLES

```
KEYPAD:ERASE 4          // Erases definition for Soft
                        // Function Key F4.
```



Before **KEYPAD:ERASE** command is executed

After **KEYPAD:ERASE** command is executed

Figure 3-9 **KEYPAD:ERASE** Example

KEYPAD:LABEL

USE To create Soft Function Key definitions.

SYNTAX **KEYPAD:LABEL** *n*, "*label*"

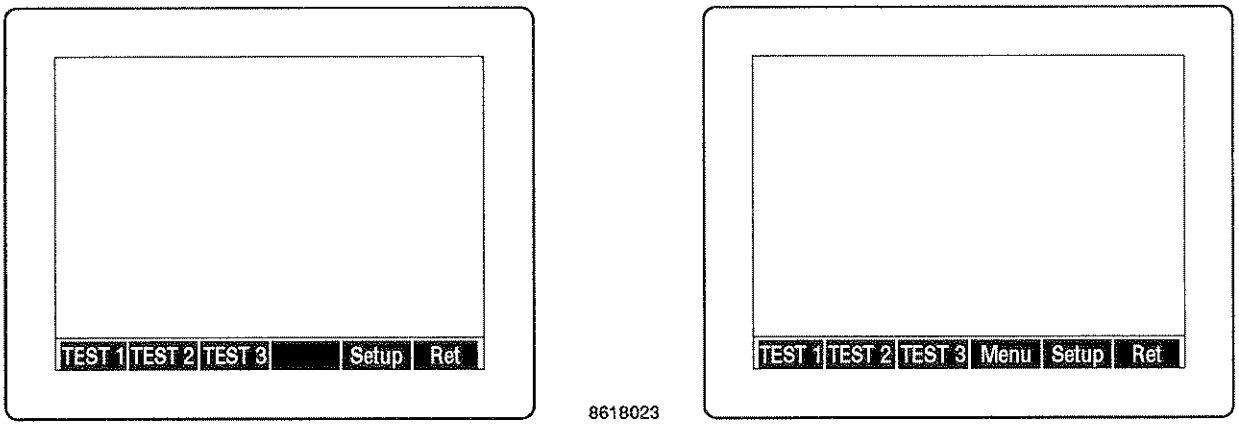
ACTION Creates a definition for the specified Soft Function Key.

VARIABLE	DESCRIPTION
n	Number of the Soft Function Key. Range of *n* is from 1 to 6.
label	Definition for the specified Soft Function Key. Executing without a label between the quotation marks ("") is the same as initiating the **KEYPAD:ERASE** command.

SEE ALSO Soft Function Key Displays (3-13-4), **KEYPAD:ERASE**, **KEYPAD:SOFT**

EXAMPLES

```
KEYPAD:LABEL 4, "Menu"
```



Before **KEYPAD:LABEL** command is executed

After **KEYPAD:LABEL** command is executed

Figure 3-10 **KEYPAD:LABEL** Example

KEYPAD:SOFT (1600S Only)

USE To create the Soft Function Key frame located at the bottom of the current screen.

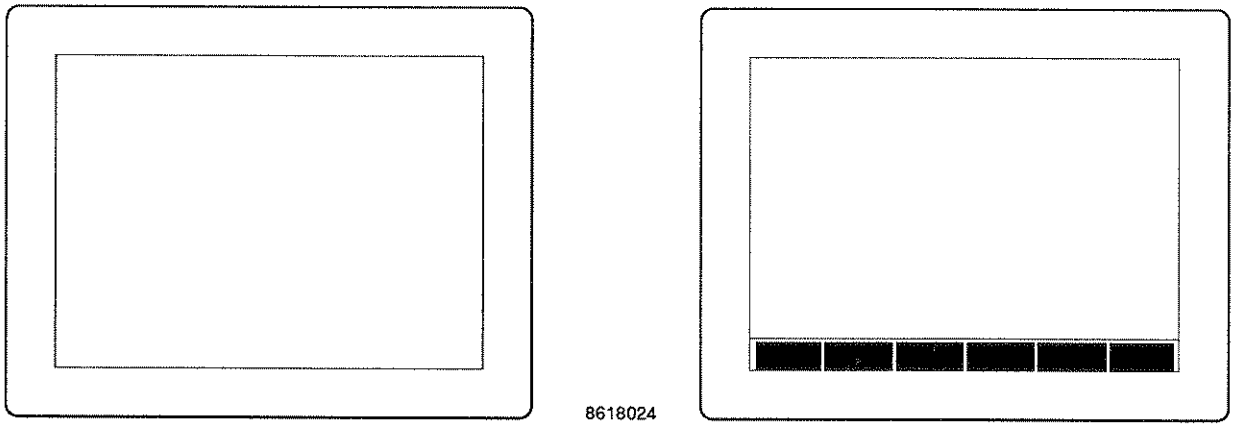
SYNTAX **KEYPAD:SOFT**

ACTION Creates the Soft Function Key frame at the bottom of the current screen.

SEE ALSO Soft Function Key Displays (3-13-4), **KEYPAD:ERASE**, **KEYPAD:LABEL**

EXAMPLES

```
KEYPAD:SOFT
```



Before **KEYPAD:SOFT** command is executed

After **KEYPAD:SOFT** command is executed

Figure 3-11 **KEYPAD:SOFT** Example

NVRCL (1600S Only)

USE To recall a variable or array stored in non-volatile memory.

SYNTAX **NVRCL** *name, location*

ACTION Recalls global variables and arrays saved in nonvolatile memory

VARIABLE	DESCRIPTION
name	Name given to recalled variable or array.
location	Location of the variable or first element of the array.

SEE ALSO Saving Variables and Arrays (3-3-2), **NVSAV**

EXAMPLE See **NVRCL** examples (3-3-2).

NVSAV (1600S Only)

USE To save variables and arrays in non-volatile memory.

SYNTAX **NVSAV** *name, location*

ACTION Saves global variables or arrays in non-volatile memory. Global variables and arrays are not saved when Test Set is turned off and need to be saved in non-volatile memory.

VARIABLE	DESCRIPTION
name	Name of variable or array to be saved.
location	Memory location to store variable or first element of array. There are 111 memory locations available. Arrays take a memory location for each element.

SEE ALSO Saving Variables and Arrays (3-3-2), **NVRCL**

EXAMPLE See **NVSAV** examples (3-3-2).

PIXEL

USE To create a user defined dot on the CRT.

SYNTAX **PIXEL**

ACTION Creates a dot in the current foreground color at the current xy location set by the **XY** command. If the foreground color or xy location is changed, the dot does not change.

SEE ALSO Graphics (3-13-5), **XY**

EXAMPLES

```
XY 300,150
COLOR 6
PIXEL
```

A brown dot appears at the xy location of 300, 150.

PPRINT (1600S Only)

USE To print responses out the RS-232 Connector.

SYNTAX **PPRINT** %*0nl*, *expression*, ... ,*expression*, "*ccc*"

ACTION Sends all responses that would normally be printed on the CRT to the device connected to the RS-232 Connector. (xy location has no affect on the **PPRINT** command.)

VARIABLES	DESCRIPTION
0	Zero signifies leading zeros are added to fill the field width. The *0* is omitted for no leading zeros. Applies only to numeric expressions.
n	Specifies the field width. *n* is the number of digit spaces (decimal point counts as one). Applies only to numeric expressions.
l	Specifies the numeric base the data is printed in. Applies only to numeric expressions. Omit for a default of Signed Decimal or enter one of the following: B Binary Q Octal H Hexadecimal U Unsigned decimal D Signed decimal
expression	Expression to be printed. Can be a mathematical expression, a literal string or the contents of variables. Mathematical expressions are calculated and string functions are performed before the results are printed.
ccc	Non-printable characters in octal form used to specify the output. Select characters specified by the devise being printed to or as follows: n or l Specifies a line feed character. r Specifies a carriage return.

SEE ALSO **PRINT, PSCREEN**

EXAMPLES

```
PPRINT "Ring\007\n"
```

Prints `Ring` on the RS-232 device. The `\007` causes the devices bell to sound (if device has a bell). The `\n` provides a line feed to the device.

PRINT

USE

To print on the CRT screen (1600S) or to print responses out the RS-232 Connector (CSA).

SYNTAX

PRINT *%0nl, expression, ... ,expression,* or
PRINT *expression, ... ,%0nl, expression,*

ACTION

Prints numerical and string values at the current xy location on the CRT. Several values are printed on the same line by separating them with commas. If the optional ending comma is used, the next **PRINT** command continues on the same line, otherwise a new line is started. If an expression is a variable, the value of the variable is printed. Mathematical expressions are calculated before printing.

The *%0nl* is the format settings for numeric expressions and can be placed anywhere in the **PRINT** command. There can be several format settings in each **PRINT** command, each one changing the leading zero format, the numeric base and the field width for the expressions that follow. Format settings do not affect the printing of strings.

VARIABLE	DESCRIPTION
0	Zero signifies leading zeros are added to fill the field width. The *0* is omitted for no leading zeros. Applies only to numeric expressions.
n	Specifies field width. *n* is the minimum number of digit spaces (decimal point counts as one). Applies only to numeric expressions. For CSA, if *l* is D, *n* can be in the form of *b.a*, where *b* designates minimum number of digits before the decimal point and *a* designates the maximum number of digits after the decimal point.
l	Specifies the numeric base the data is printed in. Applies only to numeric expressions. Omit for a default of Signed Decimal or enter one of the following: B Binary Q Octal H Hexadecimal U Unsigned decimal D Signed decimal
expression	Expression to be printed. Can be a mathematical expression, a literal string or the contents of variables. Mathematical expressions are calculated and string functions are performed before the results are printed.

SEE ALSO

HPRINT, XYPRINT, PPRINT, PSCREEN, XY

EXAMPLES

```
X=44
Y=37.83
$="String"

PRINT %04,X," ",$
```

This **PRINT** command contains a format setting specifying leading zeros and a width of 4 characters. Signed decimal is the numeric base by default. `0044 String` is printed.

```
PRINT X,%3," ",Y
```

No leading zeros and a field width of 3 is specified after X is printed. `44 37.83` is printed with the field width of 3 being expanded to fit Y.

```
PRINT %4B,X
```

No leading zeros, a field width of 4 and the Binary numeric base is specified. `101100` is printed. The field width is enlarged to accommodate X.

```
PRINT %7,$,Y
```

No leading zeros and a field width of 7 is specified. `String  37.83` is printed with the 3 blank character spaces provided by the field width of 7.

```
PRINT %03Q,X,$,%6,Y
```

Leading zeros, a field width of 3 and the Octal number base is specified by the first format setting printing `054String`. No leading zeros, a field width of 6 and the Signed Decimal number base (by default) is specified by the second format setting printing `37.83`. Each % changes the format for the expressions that follow. The complete printed statement is `054String 37.83`.

PSCREEN (1600S Only)

USE To print the CRT Screen.

SYNTAX **PSCREEN**

ACTION Prints the CRT display using the device connected to the RS-232 Connector. RS-232 parameters must be configured for printing (see 6-2 of FM/AM-1600S or FM/AM-1600CSA Operation Manual). Command is entered through GPIB Connector or placed in a macro since printer is connected to RS-232 Connector.

SEE ALSO **PRINT, PPRINT**

RETURN

USE

To return a value from a function or to return macro execution to the previous macro.

SYNTAX

RETURN

ACTION

Stops command execution of the macro it is contained in. If the macro was called from another macro, command execution resumes in the previous macro. If a variable is specified with the **RETURN** command, the variables value is returned.

EXAMPLE

```
*DMC "Even",BEGIN
IF $1 % 2 = 0
  PRINT $1," IS EVEN"
ELSE
  RETURN
ENDIF
END

*DMC "Test",BEGIN
Y= $1**$2
PRINT $1,"TO THE POWER OF ",$2," IS ",Y
Even Y
PRINT "TEST IS DONE"
END
```

If passed variable is odd, $1 % 2 equals 1 and the **RETURN** command returns command execution back to the macro Test at the PRINT "TEST IS DONE" command.

```
*DMC "Maximum",BEGIN
VAR Max=0
EDIT:WIDTH 45
FOR X=1 TO 10
INPUT Y
IF Y>Max
  Max=Y
ENDIF
NEXT X
RETURN Max
END
```

The **RETURN** command returns the value of Max. The function Maximum cannot be executed by simply entering it's name. Functions are executed by assigning them to a variable or using them in a **PRINT** command.

```
Z=Maximum
```

Function Maximum is executed and the result is assigned to Z.

```
PPRINT Maximum
```

Function Maximum is executed and the result is printed on the Host Terminal.

ROOM

USE Finds the amount of memory size available.

SYNTAX **ROOM**

ACTION Returns the available memory size, in bytes, when printed.

SEE ALSO Macros (3-9)

EXAMPLES

```
PRINT ROOM
```

130944 is printed on the CRT (1600S) or RS-232 Host device (CSA).

ROTATE (1600S Only)

USE Rotates text character by character.

SYNTAX **ROTATE** *n*

ACTION Rotates the font of text printed on the CRT. Font rotates clockwise character by character in 90° intervals.

VARIABLE	DESCRIPTION
n	Degrees the character is rotated. Table 3-5 shows the amount of clockwise rotation for values of *n*.

n	ROTATION
0 to 89	0
90 to 179	90°
180 to 269	180°
270 to 359	270°

Table 3-5 Character Rotation for *n* Values

SLEEP

USE To temporarily stop multitasking a macro.

SYNTAX **SLEEP** "*name*"

ACTION Removes the specified task from the schedule queue. The task can be put back on the schedule queue using the **WAKE** command.

VARIABLE	DESCRIPTION
name	Name of the macro to take off the schedule queue.

SEE ALSO Multitasking Macros (3-12), **WAKE**

EXAMPLES See Multitasking examples (3-12).

SOUND

USE To create audio tones.

SYNTAX **SOUND** *f*, *t*

ACTION Generates a tone of frequency *f* for a length of time *t* and routes the tone to the Test Set Speaker.

VARIABLE	DESCRIPTION
f	Frequency of tone (in Hz) generated.
t	Length of time (in ms) tone is generated.

EXAMPLES

```
SOUND 600,1000          // Creates a 600 Hz tone for 1 sec.
```

STOP

USE For Multitasking.

SYNTAX **STOP**

ACTION Stops command execution of all macros regardless of where the **STOP** command is located.

EXAMPLES

```
*DMC "Test",BEGIN
Y= $1**$2
IF $2 < 0
  STOP
ENDIF
PRINT $1,"TO THE POWER OF ",$2," IS ",Y
PRINT "TEST IS DONE"
END
```

If the passed variable is negative, the **STOP** command is executed and all macro command execution stops.

SEE ALSO Multitasking Macros (3-12)

STRING

USE To declare string variables and arrays.

SYNTAX
STRING *name, ... ,name*
STRING *name[index], ... ,name[index]*

ACTION String variables and arrays are either local or global. Local string variables are declared inside a macro and have no meaning outside that macro. Global string variables are declared outside macros and are used inside or outside any macro. String variables and arrays cannot be initialized when declared.

VARIABLE	DESCRIPTION
name	Name of the string declared. The first character of the *name* is a letter while the remaining characters can be letters, digits and the underscore. The length of the *name* can be from 2 to 31 characters.
[*index*]	Number of array elements minus one.

SEE ALSO String Variables and Functions (3-8)

EXAMPLES

```
STRING Mess_1,Mess_2       // Declares string variables named
                           // Mess_1 and Mess_2.
Mess_1="Test "             // Assigns "Test " to variable Mess_1.
Mess_2=Mess_1+"complete" // Assigns "Test complete" to the
                           // string variable Mess_2.
```

TASK

USE For Multitasking.

SYNTAX **TASK** "*name*"

ACTION Declares the specified macro as a task. A macro must be declared a task before it can be placed on the schedule queue for multitasking. The macro must already be loaded in memory.

VARIABLE	DESCRIPTION
name	Name of the macro to be declared a task.

SEE ALSO Multitasking Macros (3-12), **ACTIVATE**

EXAMPLES See Multitasking example (3-12).

TPAUSE

USE For Multitasking and sharing execution time with other Test Set activities.

SYNTAX **TPAUSE**

ACTION While multitasking, stops command execution and passes command execution to the next task in the schedule queue.

SEE ALSO Multitasking Macros (3-12), **TSTOP**

EXAMPLES See Multitasking example (3-12)

TSTOP

USE For Multitasking.

SYNTAX **TSTOP**

ACTION While multitasking, stops command execution and removes the currently running task from the schedule queue and performs a **TPAUSE**. Command execution passes to the next task in the schedule queue.

SEE ALSO Multitasking Macros (3-12), **TPAUSE**

EXAMPLES See Multitasking example (3-12).

VAR

USE To declare variables and arrays.

SYNTAX
VAR *name, ... ,name*
VAR *name[index], ... ,name[index]*

ACTION Variables are declared with a *name*. Arrays are declared with a *name* and an *index*. Variables and arrays are either local or global. Local variables and arrays are declared inside a macro and have no meaning outside that macro. Global variables and arrays are declared outside macros and are used inside or outside any macro. Variables and arrays can be initialized when declared using an = character. Using a variable name already used to declare another variable, voids the previous variable.

VARIABLE	DESCRIPTION
name	Name of variable or array declared. The first character of the *name* is a letter while the remaining characters can be letters, digits or the underscore. The length of the *name* can be from 2 to 31 characters.
[index]	Index of an array. Identifies the elements of the array by specifying the order of occurrence. The count of elements starts with 0.

SEE ALSO Variables and Arrays (3-3)

EXAMPLES

```
VAR Flag=0,Set_Squ=.9      // Variables Flag and Set_Squ are
                           // declared and initialized to 0
                           // and 0.9.
VAR Dat[2]={3.2,3.4,3.6}   // Array Data is declared with
                           // 3 elements and each element iss
                           // initialized to a value.
```

WAKE

USE For Multitasking.

SYNTAX **WAKE "***name***"**

ACTION Reenters a task (taken off the schedule queue by a **SLEEP** command) into the schedule queue.

VARIABLE	DESCRIPTION
name	Name of the task put back on the schedule queue.

SEE ALSO Multitasking macros (3-12), **SLEEP**

EXAMPLES See Multitasking examples (3-12).

WCLOSE

USE To erase a generated window on the CRT.

SYNTAX **WCLOSE** *n*

ACTION Closes (deletes) the window specified by *n*.

VARIABLE	DESCRIPTION
n	Number of the window to close. Windows are numbered in the order they are created. Overlapping windows must be closed in the reverse order they are opened.

SEE ALSO Windows (3-13-3), **WOPEN**, **WINDOW?**

EXAMPLES See Window examples (3-13-3).

WHILE WEND

USE To perform a set of commands repetitiously while a conditional expression is true.

SYNTAX
WHILE *condition*
sequence
WEND

ACTION The **WHILE** loop repeatedly executes a command *sequence* until a condition is false. If *condition* is false when first evaluated, the command sequence is not executed once like the **DO** loop since the condition is tested before the command *sequence*.

VARIABLE	DESCRIPTION
condition	Conditional expression evaluated. If true the command *sequence* is performed and command execution is looped up to the **WHILE** command until *condition* is false.
sequence	Command sequence performed as long as *condition* is true.

EXAMPLES

```
N=0
WHILE N!=2
  N=N+1
WEND
```

N is equated to 0 and the **WHILE** command is executed initiating the following sequence:

- N is compared to 2 (N=0).
- 1 is added to N.

- N is compared to 2 (N now equals 1).
- 1 is added to N.
- N is compared to 2 (N now equals 2).
- N skips over commands inside loop and these commands are not executed. Macro execution continues after **WEND** command.

```
VAR FF
WHILE FF<=LN(2*S)
  FF=FF<<2
  PRINT FF
WEND
```

The value of FF is shifted left 2 bits and printed until it is greater than LN(2*S).

```
WHILE SYSTEM:KEY? != 32896
  command
  .
  .
WEND
```

This **WHILE** loop continues looping until the Front Panel Key with a keycode of 32896 is pressed.

WINDOW?

USE To return the window currently selected.

SYNTAX **WINDOW?**

ACTION Returns the number of the currently selected window. The selected window is the window affected by the **WMOVE** and **WCLOSE** commands. The last opened window is the selected window unless a **WSEL** command has selected another window.

SEE ALSO Windows (3-13-3), **WSEL**, **WMOVE**, **WCLOSE**

EXAMPLES See Window examples (3-13-3).

WMOVE

USE To move windows.

SYNTAX **WMOVE** *x, y*

ACTION Moves the currently selected window so that the top left corner of the window is located at the specified point. Moving a window that overlaps another window disrupts the window overlapped.

VARIABLE	DESCRIPTION
x, y	New coordinates on the CRT of the top left corner of the window to be moved. Window size and shape are not affected.

SEE ALSO Windows (3-13-3), **WSEL**, **WINDOW?**

EXAMPLES See Window examples (3-13-3).

WOPEN

USE To create windows.

SYNTAX **WOPEN** *c, x1, y1, x2, y2*

ACTION Creates a window on the CRT of color *c* with a top left point of *x1, y1* and a bottom right corner of *x2, y2*. Height and width of window are forced to 16 pixel increments.

NOTE: A lack of available memory can cause a window not to open.

VARIABLE	DESCRIPTION
c	Number of the color selected for the window.
x1, y1	CRT coordinates of the top left corner of the window being opened.
x2, y2	CRT coordinates of the bottom right corner of the window being opened.

SEE ALSO Windows (3-13- 3), **WCLOSE**

EXAMPLES See Windows examples (3-13-3).

WSEL

USE To change which window is currently selected.

SYNTAX **WSEL** *n*, *h*

ACTION Makes the specified window the currently selected window and hides the window if *h* is set to 1. The selected window is the window affected by the **WMOVE** and **WCLOSE** command.

VARIABLE	DESCRIPTION
n	Number of the window to become the currently selected window. Windows are numbered in the order they are opened.
h	Specifies if window should be hidden (*h* is 1 for window to be hidden, 0 for window to show). A hidden window does not appear on the CRT and must be closed and opened again to reappear.

SEE ALSO Windows (3-13-3), **WINDOW?**, **WMOVE**, **WCLOSE**

EXAMPLES See Window examples (3-13-3).

XY

USE To specify a point on the CRT for other display commands to use.

SYNTAX **XY** *n*, *m*

ACTION Specifies a point (one pixel) on the CRT to be used by the **PRINT**, **PIXEL**, **HPRINT** and **ICON** commands. The point 0,0 is the top left corner of the CRT, with positive *m* in the downward direction and positive *n* in the right direction. The **PRINT** command moves the xy position to a new line using the current **HEIGHT** setting for the height of the new line.

VARIABLE	DESCRIPTION
n	The x coordinate (horizontal position) of the xy location. Range is the number of pixels across the CRT (left to right side) (0 to 639).
m	The y coordinate (vertical position) of the xy location. Range is the number of pixels across the CRT (top to bottom) (0 to 349).

SEE ALSO CRT Display (3-13-2), Graphics (3-13-5)

EXAMPLES See Graphics example (3-13-5).

XYPRINT (CSA Only)

USE To print on the CRT screen.

SYNTAX **XYPRINT** *x,y, %0nl, expression, ... ,expression,* or
XYPRINT *x,y, expression, ... ,%0nl, expression,*

ACTION Combines the **XY** and **HPRINT** commands. Prints numerical and string values at the selected *x,y* location on the CRT. Several values are printed on the same line by separating them with commas. If an expression is a variable, the value of the variable is printed. Mathematical expressions are calculated before printing.

The *%0nl* is the format settings for numeric expressions and can be placed anywhere in the **XYPRINT** command. There can be several format settings in each **XYPRINT** command, each one changing the leading zero format, the numeric base and the field width for the expressions that follow. Format settings do not affect the printing of strings.

VARIABLE	DESCRIPTION
x	The x coordinate (horizontal position) of the xy location. Range is the number of pixels across the CRT (left to right side) (0 to 639).
y	The y coordinate (vertical position) of the xy location. Range is the number of pixels across the CRT (top to bottom) (0 to 349).
0	Zero signifies leading zeros are added to fill the field width. The *0* is omitted for no leading zeros. Applies only to numeric expressions.
n	Specifies field width. *n* is the minimum number of digit spaces (decimal point counts as one). Applies only to numeric expressions. If *l* is D, *n* can be in the form of *b.a*, where *b* designates minimum number of digits before the decimal point and *a* designates the maximum number of digits after the decimal point.
l	Specifies the numeric base the data is printed in. Applies only to numeric expressions. Omit for a default of Signed Decimal or enter one of the following: B Binary Q Octal H Hexadecimal U Unsigned decimal D Signed decimal
expression	Expression to be printed. Can be a mathematical expression, a literal string or the contents of variables. Mathematical expressions are calculated and string functions are performed before the results are printed.

SEE ALSO **PRINT, PPRINT, PSCREEN**

EXAMPLES

```
U=44
T=37.83
STRING NOT
NOT="MOBILE PHONE DOES NOT RESPOND"

XYPRINT 5,5,%04,U," ",NOT
```

This **XYPRINT** command contains a format setting specifying leading zeros and a width of four characters. Signed decimal is the numeric base by default. The command prints `0044 MOBILE PHONE DOES NOT RESPOND` starting at point 5, 5 location on the CRT.

```
XYPRINT 319,174,U,%3," ",T
```

No leading zeros and a field width of 3 is specified after U is printed. The CRT displays `44 37.83` starting in the middle of the screen. The field width of 3 was expanded to fit T.

```
XYPRINT 0,0,%4B,X
```

No leading zeros, a field width of 4 and the Binary numeric base is specified. `101100` is printed in the upper left corner of the screen. The field width is enlarged to accommodate U.

```
XYPRINT 30,55,%7,$,Y
```

No leading zeros and a field width of 7 is specified. Starting at xy location (30, 50), `MOBILE PHONE DOES NOT RESPOND  37.83` is printed. The three blank character spaces are provided by the field width of 7.

```
XYPRINT 319,174,%3.1,T
```

No leading zeros and a minimum field width of 3 with a maximum of 1 digit after the decimal point is specified. The CRT displays `37.8` starting in the middle of the screen. The field width of 3 was expanded to fit T, counting the decimal point.

SECTION 4 - 1600S SPECIFIC TMAC COMMANDS

4-1 GENERAL

This Section lists FM/AM-1600S specific commands by Operation Mode. Many commands do not operate unless Test Set routing is configured for the operation mode of that command. **SETUP** commands establish routings for selected operation modes (see 4-2). **SCREEN** commands establish routings and display the operation screen for the selected operation mode (see 4-3).

Commands affect the Test Set when executed; however changes do not appear on the Operation Screens until the Screen is renewed using a **SCREEN** command.

The short form of the command is shown in uppercase letters and the long form is finished in lower case. When entering commands, it is not necessary to use a particular letter case. The TMAC compiler/interpreter is not case sensitive. Brackets ([]) indicate optional command items.

4-2 SETUP COMMANDS

SETUP commands establish the routings for various Operation Modes without displaying the Operation Screens on the CRT.

SETUP:

AF
Configures Test Set routing for AF Meter Operation.

ANLZ
Configures Test Set routing for Spectrum Analyzer Operation.

DISTortion
Configures Test Set routing for Distortion Meter Operation.

DUPlex
Configures Test Set routing for Duplex Operation.

DUPRX
Configures Test Set routing for Duplex Receiver Operation.

DUPTX
Configures Test Set routing for Duplex Transmitter Operation.

FUNC
Configures Test Set routing for AF Generator Operation.

GENerator
Configures Test Set routing for RF Generator Operation.

MONitor
Configures Test Set routing for Generator/Monitor Operation.

RECeiver
Configures Test Set routing for Receiver Operation.

SCOPe
Configures Test Set routing for Oscilloscope Operation.

SINAD
Configures Test Set routing for SINAD Meter Operation.

4-3 SCREEN COMMANDS

SCREEN commands renew and display Mode Operation Screen on the CRT. Many commands do not operate correctly unless the CRT displays the applicable Mode Operation Screen.

SCREEN:

AF
Displays Audio Frequency Meter Operation Screen.

ANLZ
Displays Spectrum Analyzer Operation Screen.

BER
Displays Bit Error Rate Meter Operation Screen.

CELL
Displays Forward Control Channel Screen of the AMPS Cell Site Monitor (AMPS option installed).

DEViation
Displays Deviation Meter (Peak) Operation Screen.

DISTortion
Displays Distortion Meter Operation Screen.

DMM
Displays Digital Multimeter Operation Screen.

DRMS
Displays Deviation Meter (RMS) Operation Screen.

DUPlex
Displays Duplex Operation Screen.

DUPRX
Displays Duplex Receiver Operation Screen.

DUPTX
Displays Duplex Transmitter Operation Screen.

FREQuency
Displays Frequency Error Meter Operation Screen.

FUNC
Displays AF Generator Operation Screen.

GENCELLular
Displays Main Menu of AMPS Mobile Simulator (AMPS option installed).

GENerator
Displays RF Generator Operation Screen.

GENRECC
Displays Reverse Control Channel Simulation Screen of AMPS Mobile Simulator (AMPS option installed).

SCREEN:

GENRVC
Displays Reverse Voice Channel Simulation Screen of AMPS Mobile Simulator (AMPS option installed).

MODulation
Displays Modulation Meter Operation Screen.

PM
Displays Phase Meter Operation Screen.

PMRMS
Displays Phase Meter (RMS) Operation Screen.

POWer
Displays Power Meter Operation Screen.

RECeiver
Displays Receiver Operation Screen.

SCOPe
Displays Oscilloscope Operation Screen.

SIGnal
Displays Signal Strength Meter Operation Screen.

SINAD
Displays SINAD Meter Operation Screen.

USER
Displays blank User Screen without changing routing. User Screen has no readings to update, decreasing run time when inside a macro. Following this command with other graphic, window and keypad commands creates customized menu screens (see 3-13).

4-4 AUXILIARY TEST SET COMMANDS

Auxiliary Test Set commands are used to communicate with an attached Auxiliary Test Set through the SCSI Connector. The Auxiliary Test Set holds certain options available with the FM/AM-1600S and may or may not be installed. Because of the vast command possibilities, the FM/AM-1600S does not check the Auxiliary Test Set command for correct string syntax.

AHIT?
Returns a 1 if there is input waiting from the Auxiliary Test Set.

AUX *"string"*
Issues commands, as strings, to the Auxiliary Test Set.

Example:
```
AUX "DELAY 200"        // The DELAY 200 command is passed to and
                       // executed by the Auxiliary Test Set.
```

AUX? *"string?"*
Issues queries, as strings, to the Auxiliary Test Set.

Example:
```
AUX? "PEEK? 1250"      // The PEEK? 1250 query is passed to the
                       // Auxiliary Test Set.  The Auxiliary Test
                       // Set executes the query and returns the result.
```

4-5 RF GENERATOR

4-5-1 RF GENERATOR COMMANDS

GENerator:

AF
Displays AF Level Meter on the RF Generator Operation Screen when followed by **SCREEN:GENerator** command.

CHANnel *n*
Sets Frequency to cellular channel *n* (1 to 1023) in the format selected using the **GEN:CHAN:FORM** command. RF Generator Operation Screen displays selected cellular channel when RF Generator is in Channel Mode (**GEN:MODE CHAN** command).

CHANnel:

FORMat:

AMPS:

FORward
Selects AMPS Forward channels.

REVerse
Selects AMPS Reverse channels.

ETACS:

FORward
Selects ETACS Forward channels.

REVerse
Selects ETACS Reverse channels.

NAMPS:

FORward
Selects NAMPS Forward channels.

REVerse
Selects NAMPS Reverse channels.

NT400:

FORward
Selects NT400 Forward channels.

REVerse
Selects NT400 Reverse channels.

FORMat?
Returns channel format setting (AMPS:FORWARD, AMPS:REVERSE, ETACS:FORWARD, ETACS:REVERSE, NAMPS:FORWARD, NAMPS:REVERSE, NT400:FORWARD or NT400:REVERSE).

GENerator:

DCS:

INVert *nnn*
Generates the three digit octal Digital Coded Squelch (DCS) code in inverted mode. Range of *nnn* is 000 to 777.

NORMal *nnn*
Generates the three digit octal DCS code. Range of *nnn* is 000 to 777. See 4-5-1G.

STOP
Stops generating the continuous DCS code.

```
Example: GEN:DCS:NORM 411       // Generates a continuous DCS code of 411.
         DELAY 200              // Waits for 200 ms before next command.
         GEN:DCS:STOP           // Stops generating the DCS code.
```

DIAL *"sequence"*
Generates *sequence* with a maximum of 16 digits (0 through 9 allowed) as 2805 Pulse code. See 4-5-1I.

DIAL:

FREQuency *f*
Sets 2805 Tone frequency to *f* Hz. Range of *f* is 0.0 to 40000.0.

FREQuency?
Returns the current 2805 Tone frequency in Hz.

```
Example: GEN:DIAL:FREQ 3000     // Sets 2805 Tone frequency to 3000 Hz.
         GEN:DIAL:FREQ?         // Returns 2805 Tone frequency setting.
         GEN:DIAL "5552424"     // Generates a 2805 Tone frequency message
                                // of 5552424.
```

DISTortion
Displays Distortion Meter on the RF Generator Operation Screen after the screen is updated by a **SCREEN:GENerator** command.

DMM
Displays Digital Multimeter on the RF Generator Operation Screen after the screen is updated by a **SCREEN:GENerator** command.

DSAT *n*
(NAMPS option installed) Generates the Digital Supervisory Audio Tone (DSAT) code specified by *n*. Range of *n* is 0 to 6.

DSAT:STOP
(NAMPS option installed) Stops generating the continuous DSAT code.

DST *n*
(NAMPS option installed) Generates the Digital Signal Tone (DST) code specified by *n*. Range of *n* is 0 to 6.

DST:STOP
(NAMPS option installed) Stops generating the continuous DST code.

GENerator:

DTMF *"sequence",mark,space*
Generates a DTMF coded *sequence*. *sequence* has a maximum of 16 digits. Range of optional *mark* and *space* time is from 25 to 9999 ms. Default for *mark* time is 74 ms. Default for *space* time is 67 ms. See 4-5-1D.

FREQuency *n* [*units*]
Sets RF Generator Frequency. Range of *n* is 250.0 kHz to 999.9999 MHz. Select HZ, KHZ, MHZ for *units*. Default for optional *units* is KHZ.

FREQuency?
Returns RF Generator Frequency in kHz (250.0 to 999999.9).

```
Example: GEN:FREQ 100 MHZ        // Set RF Generator Frequency to 100 MHz.
         GEN:FREQ?               // Query the RF Generator Frequency.
                                 // 100000 is returned (100000 kHz).
```

IMTS *"sequence"*
Generates *sequence* as a DCS IMTS code. *sequence* has a maximum of 16 digits (0 through 9 allowed).

LEVel *n* [*units*]
Sets RF Generator Level. Range of *n* is -137.0 to 0.0 dBm or 0.031 µV to 0.224 V. Select DBm, V, MV (mV) or UV (µV) for *units*. *units* is optional with the default being the current units. Specifying *units* does not change unit selection of Test Set.

LEVel:UNIT?
Returns the current units for the RF Generator Level.

LEVel?
Returns the RF Generator Level in the current units.

```
Example: GEN:LEV -65 DB          // Sets RF Generator Level to -65 dBm.
         GEN:LEV:UNIT?           // Queries the units for the RF Generator
                                 // Level.  DBM is returned.
         GEN:LEV?                // Queries the RF Generator Level.  -65
                                 // is returned.
         GEN:LEV -30             // Sets RF Generator Level to -30 dBm (units
                                 // default to dBm, the current units).
```

MODE *type*
Selects the RF Generator Mode (DIRect [Direct Mode] or CHANnel [Channel Mode]). Channel Mode displays cellular channel frequency according to **GEN:CHAN** commands.

MTS *"sequence"*
Generates *sequence* as a DCS MTS code. *sequence* has a maximum of 16 digits (0 through 9 allowed).

OUTput:

AUDio *b*
Routes AF Generator Output to the AUDIO OUT Connector if *b* is 1. Disconnects the AUDIO OUT Connector if *b* is 0.

GENerator:

OUTput:

DEMOD *b*
Routes AF Generator Output to the DEMOD OUT Connector if *b* is 1. Disconnects the DEMOD OUT Connector if *b* is 0.

POCSAG:

ALPHA:

LOWer *capcode*
Generates a lower case Alpha message for the *capcode* specified. Range of the *capcode* is 0 to 2097151.

NUMeric *capcode*
Generates an Alphanumeric message for the *capcode* specified. Range of the *capcode* is 0 to 2097151.

Example:

```
GEN:POCSAG:NUMeric 4000    // Generates an Alphanumeric message at a
                           // capcode of 4000 at the current rate.
```

SPECial *capcode*
Generates an Alpha special message for the *capcode* specified.

UPPer *capcode*
Generates an upper case Alpha message for the *capcode* specified. Range of the *capcode* is 0 to 2097151. See 4-5-1H.

BEEP *n, capcode*
Generates *n* Tone Beep POCSAG message for *capcode* specified. Select 1, 2, 3 or 4 for *n*. Range of the *capcode* is 0 to 2097151.

NUMeric *capcode*
Generates Numeric message for the *capcode* specified. Range of *capcode* is 0 to 2097151.

RATe
Sets POCSAG rate to high if *b* is 1, low if *b* is 0.

RATe?
Returns 1 if POCSAG rate is high, returns 0 if rate is low.

Example:

```
*DMC "POCSAG",BEGIN    // Defines a macro named POCSAG.
GEN:POCSAG:RATE 1      // Sets POCSAG rate to High.
FOR X=1 TO 5           // Starts FOR loop of X to loop 5 times.
GEN:POCSAG:BEEP 2,X+5030  // Generates a POCSAG Tone - 2 beeps
                          // message
                       // using the capcode specified by X.
DELAY 3000             // Delays 3 seconds (for POCSAG generation).
*WAI                   // Wait for the last command to be executed.
NEXT X                 // Loops X to the top of the loop until X=5.
GEN:POCSAG:RATE?       // Queries the POCSAG rate.  1 is returned
END                    // Ends the macro named POCSAG.
```

GENerator:

RCL *n*
Recalls RF Generator environment (routings and settings) stored in memory location *n*. Range of *n* is 1 to 9.

SINAD
Displays SINAD Meter on the RF Generator Operation Screen when followed by a **SCREEN:GENerator** command.

SPEAKer:SOURce *type*
Selects Test Set Speaker Input. Select OFF, FGEN (Function Generator), SINAD (SINAD/BER IN Connector) or EXTMOD (EXT MOD IN Connector) for *type*. This command takes effect when the RF Generator Operation Screen is updated.

Example:
```
GEN:SPEAK:SOUR EXTMOD // Routes EXT MOD IN Connector to the
                      // Test Set Speaker.
```

STORe *n*
Stores current RF Generator environment (routings and settings) in memory location *n*. Range of *n* is 1 to 9.

TONE *"sequence"*
Generates given *sequence* once the Audio code is selected using the **GENerator:TONE:TYPE** command. If selected Audio code is USER, characters contained in the *sequence* must be previously defined using the **GENerator:TONE:USER:DEFine** command. Valid *sequence* characters are the digits 0 through 9 and characters A, G, R and - (to signify a gap). Valid *sequence* characters for the USER code are the digits 0 through 9 and characters A through T.

TONE:

TYPE *code*
Selects the Audio code to generate. Enter one of the following for *code*:

CCIR	EEA	EIA (U.S).
ZVEI	DDZVEI	DZVEI
NATEL	EURO	TONE56
CCIRH	CCIRH4	USER

See 4-5-1E and F.

USER:DEFine *"id",freq,duration*
The *id* character is assigned a *freq* in Hz and a *duration* in ms. Range of *freq* is 0.0 to 9999.9. Range of *duration* is 20.0 to 9999.9. Valid characters for the *id* are the digits 0 through 9 and characters A through T. See 4-5-1E and 4-5-1F.

Example:
```
GEN:TONE:TYPE CCIRH    // Selects CCIRH as RF Generator Audio Code.
GEN:TONE "5553434"     // Generates 5552424 as CCIRH Audio Signal.
```

GENerator:

TREMote
Generates the Tone-remote sequence for the specified function tone frequency given by *f* in Hz. Select one of the following:

2050	1950	1850
1750	1650	1550
1450	1350	1250
1150	1050	

TREMote:STOP
Stops the Tone remote Guard Tone generated by a previous **GENerator:TREMote** command.

Example:
```
GEN:TREM 1350          // Generates a Tone Remote 1350 message.
GEN:TREM:STOP          // Stops the generating of the Tone Remote
                       // Guard Tone.
```

4-5-2 REMOTE RF GENERATOR EXAMPLES

A. GENERATING FM MODULATED RF SIGNALS

The following command sequence generates a 1 MHz RF signal FM modulated (5 kHz deviation) with a 1 kHz sine wave and routes the signal to the DEMOD OUT Connector:

```
SCREEN:GEN                      // Displays RF Generator Operation Screen.
GEN:FREQ 1000000                // Sets RF Generator Frequency to 1 MHz.
GEN:LEVEL 0 DB                  // Sets the RF Generator Level to 0 dBm.
GEN:OUTPUT:DEMOD 1              // Routes the RF Generator output to the
                                // DEMOD OUT Connector.
FGEN:GEN1:STATE 1               // Activates SOURCE 1 (AF Generator 1).
FGEN:GEN1:FREQ 1000             // Sets SOURCE 1 Frequency to 1000 Hz.
FGEN:GEN1:MODULATION:FM         // Sets SOURCE 1 Modulation to FM.
FGEN:GEN1:MODL 5                // Sets SOURCE 1 Deviation to 5 kHz.
FGEN:GEN1:SHAPE:SIN             // Sets SOURCE 1 Wave Shape to a sine wave.
```

B. GENERATING AM MODULATED RF SIGNALS

The following command sequence generates a 10 MHz RF signal, AM modulated (80% modulation), with a 2.5 kHz sine wave and routes the signal to the AUDIO OUT Connector:

```
SCREEN:GEN                      // Displays RF Generator Operation Screen.
GEN:FREQ 10 MHZ                 // Sets RF Generator Frequency to 10 MHz.
GEN:LEVEL 0 DB                  // Sets the RF Generator Level to 0.0 dBm.
GEN:OUTPUT:AUDIO 1              // Routes the RF Generator output to the
                                // AUDIO OUT Connector.
FGEN:GEN1:STATE 1               // Activates SOURCE 1 (AF Generator 1).
FGEN:GEN1:FREQ 2500             // Sets SOURCE 1 Frequency to 2500 Hz.
FGEN:GEN1:MODULATION:AM         // Sets SOURCE 1 Modulation to AM.
FGEN:GEN1:MODL 80               // Sets SOURCE 1 Modulation to 80%.
FGEN:GEN1:SHAPE:SIN             // Sets SOURCE 1 Wave Shape to a sine wave.
```

C. GENERATING EXTERNALLY MODULATED RF SIGNALS

The following command sequence generates a RF 15 MHz signal, AM modulated (80% modulation), with an external signal applied to the EXT MOD IN Connector and sends the signal to the AUDIO OUT Connector:

```
SCREEN:GEN                      // Displays RF Generator Operation Screen.
GEN:FREQ 15000                  // Sets RF Generator Frequency to 15 MHz.
GEN:LEV -20                     // Sets the RF Generator Level to -20 dBm.
GEN:OUT:AUD 1                   // Routes the RF Generator Output to the
                                // AUDIO OUT Connector.
FGEN:EXT:STATE 1                // Activates SOURCE EXT (for external
                                // modulation).
FGEN:EXT:MOD:AM                 // Sets SOURCE EXT Modulation to AM.
FGEN:EXT:MODL 80                // Sets SOURCE EXT Modulation Level to 80%.
```

D. GENERATING A DTMF CODED SIGNAL

The following command sequence generates a 450 MHz signal FM modulated with a DTMF Code. The modulation level is 4 kHz and the RF Output Level is -60 dBm.

```
SCREEN:GEN                      // Displays RF Generator Operation Screen.
*WAI                            // Waits for previous commands to execute.
GEN:FREQ 450 MHZ                // Sets RF Generator Frequency to 450 MHz.
GEN:LEV -60 DB                  // Sets the RF Generator Level to -60 dBm.
FGEN:GEN3:ENCODE DTMF           // Selects DTMF as the signaling format.
SCREEN:GEN                      // Renews RF Generator Operation Screen.
*WAI                            // Waits for previous commands to execute.
FGEN:GEN3:MOD:FM                // Sets SOURCE 3 Modulation to FM.
FGEN:GEN3:MODL 4                // Sets SOURCE 3 Modulation Level to 4 kHz.
GEN:DTMF "55523*#",80,70        // Generates 55523*# DTMF coded signal with
                                // 80 ms mark time and a 70 ms space time.
```

E. GENERATING AUDIO TWO TONE CODING

The following command sequence generates a 150 MHz signal FM modulated with an Audio Two Tone Code. The modulation level is 4 kHz and the RF Output Level is 0 dBm.

```
SCREEN:GEN                      // Displays RF Generator Operation Screen.
*WAI                            // Waits for previous commands to execute.
GEN:FREQ 150 MHZ                // Sets RF Generator Frequency to 150 MHz.
GEN:LEV 0 DB                    // Sets the RF Generator Level to 0 dBm.
FGEN:GEN3:ENCODE TONE           // Selects Audio as the signaling format.
SCREEN:GEN                      // Renews RF Generator Operation Screen.
*WAI                            // Waits for previous commands to execute.
FGEN:GEN3:MOD:FM                // Sets SOURCE 3 Modulation to FM.
FGEN:GEN3:MODL 4                // Sets SOURCE 3 Modulation Level to 4 kHz.
GEN:TONE:TYPE USER              // Selects User Defined for the Audio Code.
GEN:TONE:USER:DEF "0,0,100"     // Defines 0 Tone as 0 Hz, 100 ms.
GEN:TONE:USER:DEF "1,880,500"   // Defines 1 Tone as 880 Hz, 500 ms.
GEN:TONE:USER:DEF "2,2200,500"// Defines 2 Tone as 2200 Hz, 500 ms.
GEN:TONE 102                    // Generates the User Defined Audio Code
                                // defined in the 3 previous commands.
```

F. GENERATING A 5/6 AUDIO TONE SEQUENCE

The following command sequence generates a 162 MHz signal FM modulated with an Audio 5/6 Tone Code. The modulation level is 4 kHz and the RF Output Level is 0 dBm.

```
SCREEN:GEN                      // Displays RF Generator Operation Screen.
*WAI                            // Waits for previous commands to execute.
GEN:FREQ 162 MHZ                // Sets RF Generator Frequency to 162 MHz.
GEN:LEV 0 DB                    // Sets the RF Generator Level to 0 dBm.
FGEN:GEN3:ENCODE TONE           // Selects Audio as the signaling format.
SCREEN:GEN                      // Renews RF Generator Operation Screen.
*WAI                            // Waits for previous commands to execute.
FGEN:GEN3:MOD:FM                // Sets SOURCE 3 Modulation to FM.
FGEN:GEN3:MODL 4                // Sets SOURCE 3 Modulation Level to 4 kHz.
GEN:TONE:TYPE:USER              // Selects User Defined for the Audio Code.
GEN:TONE:USER:DEF 0,900,150     // Defines 0 Tone at 900 Hz and 150 ms.
GEN:TONE:USER:DEF 1,1100,80     // Defines 1 Tone at 1100 Hz and 80 ms.
GEN:TONE:USER:DEF 2,1200,80     // Defines 2 Tone at 1200 Hz and 80 ms.
GEN:TONE:USER:DEF 3,1300,80     // Defines 3 Tone at 1300 Hz and 80 ms.
GEN:TONE:USER:DEF 4,1400,80     // Defines 4 Tone at 1400 Hz and 80 ms.
GEN:TONE:USER:DEF A,0,40        // Defines A Tone at 0 Hz and 40 ms.
GEN:TONE 0A1234                 // Generates the User Defined Audio Code.
                                // defined in the 6 previous commands.
```

G. GENERATING DCS CODE

The following command sequence generates a 162.450 MHz signal FM modulated with a DCS Code of 456. The modulation level is 1 kHz and the RF Output Level is 0 dBm.

```
SCREEN:GEN                      // Displays RF Generator Operation Screen.
*WAI                            // Waits for previous commands to execute.
GEN:FREQ 162450                 // Sets 162.450 MHz RF Generator Frequency.
GEN:LEV 0 DB                    // Sets the RF Generator Level to 0 dBm.
FGEN:GEN3:ENCODE DIG            // Selects Digital as the signaling format.
SCREEN:GEN                      // Renews RF Generator Operation Screen.
*WAI                            // Waits for previous commands to execute.
FGEN:GEN3:MOD:FM                // Sets SOURCE 3 Modulation to FM.
FGEN:GEN3:MODL 1                // Sets SOURCE 3 Modulation Level to 1 kHz.
GEN:DCS:NORM 456                // Generates a 456 DCS Code.
GEN:DCS:STOP                    // Stops the generating of the DCS Code.
```

H. GENERATING POCSAG CODE

The following command sequence generates a 930 MHz signal FM modulated with a POCSAG Code. The modulation level is 4 kHz and the RF Output Level is 0 dBm.

```
SCREEN:GEN                          // Displays RF Generator Operation Screen.
*WAI                                // Waits for previous commands to execute.
GEN:FREQ 450 MHZ                    // Sets RF Generator Frequency to 450 MHz.
GEN:LEV 0 DB                        // Sets the RF Generator Level to 0 dBm.
FGEN:GEN3:ENCODE DIG                // Selects Digital as the signaling format.
SCREEN:GEN                          // Renews RF Generator Operation Screen.
*WAI                                // Waits for previous commands to execute.
FGEN:GEN3:MOD:FM                    // Sets SOURCE 3 Modulation to FM.
FGEN:GEN3:MODL 4                    // Sets SOURCE 3 Modulation Level to 4 kHz.
GEN:POCSAG:RATE 0                   // Sets POCSAG rate to Low.
FOR Y=1 TO 5                        // Starts loop.  Y to be looped 5 times.
   GEN:POCSAG:ALPHA:UPPER Y+5130     // Generates an Alpha upper POCSAG
                                     // message using the specified capcode.
   DELAY 3000                       // Delays 3 seconds (for POCSAG generation).
   *WAI                             // Wait for the last command to be executed.
NEXT Y                              // Loops to top of the loop until Y is 5.
```

I. GENERATING 2805 CODE

The following command sequence generates a 155 MHz signal FM modulated with a 2805 Tone (with frequency reset to 1500 Hz). The modulation level is 4 kHz and the RF Output Level is -60 dBm.

```
SCREEN:GEN                          // Displays RF Generator Operation Screen.
*WAI                                // Waits for previous commands to execute.
GEN:FREQ 155 MHZ                    // Sets RF Generator Frequency to 155 MHz.
GEN:LEV -60 DB                      // Sets the RF Generator Level to -60.0 dBm.
FGEN:GEN3:ENCODE RCC                // Selects RCC as the signaling format.
SCREEN:GEN                          // Renews RF Generator Operation Screen.
*WAI                                // Waits for previous commands to execute.
FGEN:GEN3:MOD:FM                    // Sets SOURCE 3 Modulation to FM.
FGEN:GEN3:MODL 4                    // Sets SOURCE 3 Modulation Level to 4 kHz.
GEN:DIAL:FREQ 1500                  // Sets 2805 Tone Frequency to 1500 Hz.
GEN:DIAL 5551234                    // Generates a 5551234 2805 Tone signal.
```

4-6 RECEIVER

4-6-1 RECEIVER COMMANDS

RECeive:

AGC:

AUTO
Sets Automatic Gain Control to automatic setting.

MANual *n*
Sets Automatic Gain Control to manual setting and sets level to *n*. Range of *n* is 0 to 255.

Example: `REC:AGC:MAN 130` `// Sets AGC setting to Manual and sets // Manual AGC setting to 130.`

USER:*xxx*
Sets Automatic Gain Control to User. Select one of the following for *xxx*: MEASure, SPeech, DATA, HIGH, TYPE1, TYPE2 or TYPE3.

Example: `REC:AGC:USER:HIGH` `// Sets AGC setting to User Defined High.`

CHANnel *n*
Sets Receive Frequency to cellular channel *n* (1 to 1023) in the format selected using the **REC:CHAN:FORM** command. The Receiver Operation Screen displays selected cellular channel when Receiver is in Channel Mode (**REC:MODE CHAN** command).

CHANnel:

FORMat:

AMPS:

FORward
Selects AMPS Forward channels.

REVerse
Selects AMPS Reverse channels.

ETACS:

FORward
Selects ETACS Forward channels.

REVerse
Selects ETACS Reverse channels.

NAMPS:

FORward
Selects NAMPS Forward channels.

REVerse
Selects NAMPS Reverse channels.

RECeive:

CHANnel:

FORMat:

NT400:

FORward
Selects NT400 Forward channels.

REVerse
Selects NT400 Reverse channels.

FORMat?
Returns channel format setting (AMPS:FORWARD, AMPS:REVERSE, ETACS:FORWARD, ETACS:REVERSE, NAMPS:FORWARD, NAMPS:REVERSE, NT400:FORWARD or NT400:REVERSE).

DCS:

INVert?
Returns the three octal digits in inverted mode, decoded from the Digital Coded Squelch (DCS) code. Returns -1 if none available or if invalid for inverted DCS.

NORMal?
Returns the three octal digits decoded from the DCS code. Returns -1 if none available or if invalid for normal DCS. See 4-6-1F.

STATe *b*
Enables DCS decoding if *b* is 1, disables if *b* is 0. **RECeive:DECode DIGital** and **RECeive:DIGital DCS** (or **DCSINV**) commands must be initiated prior to enabling DCS decoding. DCS decoding must be disabled after decoding is finished.

Example:
```
REC:DEC DIG            // Sets Receiver for decoding digital data.
REC:DIG DCS            // Prepares Receiver for decoding DCS.
REC:DCS:STAT 1         // Enables DCS decoding.
REC:DCS:NORM?          // Returns decoded Normal DCS digits.
REC:DCS:STAT 0         // Disables DCS decoding.
```

DECode *type*
Sets Receiver for decoding *type*. The setting for *type* is DTMF, TONE or DIGital.

DEVRms
Displays Deviation Meter (RMS) reading on Receiver Operation Screen when followed by **SCREEN:RECeive** command.

DIGital *type*
Sets digital *type*. Used with the **RECeive:DECode DIGital** command to prepare the Receiver for decoding. The setting for *type* is DCS, DCSINV, POCSAG, DSAT or DST.

DISTortion
Displays Distortion Meter reading on Receiver Operation Screen when followed by **SCREEN:RECeive** command.

RECeive:

DMM
Displays Digital Multimeter reading on Receiver Operation Screen when followed by **SCREEN:RECeive** command.

DSAT:

STATe *b*
(NAMPS option installed) Enables DSAT Decoding Function if *b* is 1, disables if *b* is 0. **RECeive:DECode DIGital** and **RECeive:DIGital DSAT** commands must be initiated prior to enabling DSAT decoding. DSAT decoding must be disabled after decoding is finished.

DSAT?
(NAMPS option installed) Returns DSAT reading. Returns -1 if none available or invalid for normal transmission.

DST:

STATe *b*
(NAMPS option installed) Enables DST Decoding Function if *b* is 1, disables if *b* is 0. **RECeive:DECode DIGital** and **RECeive:DIGital DST** commands must be initiated prior to enabling DST decoding. DST decoding must be disabled after decoding is finished.

DST?
(NAMPS option installed) Returns DST reading. Returns -1 if none available or invalid for normal transmission.

DTMF:

STATe *b*
Enables DTMF Decoding Function if *b* is 1, disables if *b* is 0. See 4-6-1D.

DTMF?
Returns string of decoded digits or -1 if nothing decoded.

```
Example: SCREEN:REC              // Displays Receiver Operation Screen.
         REC:DEC DTMF            // Sets Receiver for decoding DTMF signals.
         REC:DTMF:STAT 1         // Enables DTMF decoding.
         REC:DTMF?               // Returns decoded DTMF digits.
         REC:DTMF:STAT 0         // Disables DTMF decoding.
```

FIND:

FREQuency?
Returns frequency of first signal with amplitude larger than Find reference level. Returns 0 if no signal is found.

REFerence *n*
Sets Find reference level to *n* dBm (-110 to -5).

REFerence?
Returns Find reference level in dBm (-110 to -5).

RECeive:

```
Example: REC:FIND:REF -65        // Sets Find Reference Level to -65 dBm.
         REC:FIND:FREQ?          // Returns the first frequency (in kHz)
                                 // containing a signal greater than -65 dBm.
         REC:FIND:REF?           // Queries the current Find Reference Level
                                 // (in dBm).  Returns -65.
```

FREQuency *n* [*units*]
Sets Receiver Frequency. Range of *n* is 250.0 kHz to 999.9999 MHz. Select HZ, KHZ or MHZ for *units*. *units* is optional with a default of KHZ.

FREQuency?
Returns Receiver Frequency in kHz (250.0 to 999999.9).

```
Example: REC:FREQ 100 MHZ        // Sets Receiver Frequency to 100 MHz.
         REC:FREQ?               // Queries the Receiver Frequency.
                                 // Returns 100000 (100000 kHz).
```

INPut:

ANTenna
Selects ANTENNA IN Connector as the Receiver Input Source. Displays Signal Strength Meter reading on the Receiver Operation Screen.

ATTenuation *n*
Sets Input Attenuation to *n* dB. Select 0, 20 or 40.

TR
Selects T/R Connector as the Receiver Input Source. Displays Power Meter reading on the Receiver Operation Screen.

MODE *type*
Selects Receiver Mode (DIRect, CHANnel or SCAN [Frequency Scan]). Channel Mode displays the cellular channel frequency according to **REC:CHAN** commands.

```
Example: REC:INP:TR              // Selects TR Connector for Receiver Input.
         REC:INP:ATT 20          // Sets Receiver Input Attenuation to 20 dB.
         REC:MODE CHAN           // Selects Receiver Channel Mode.
```

MODMeter
Displays Modulation Meter reading on the Receiver Operation Screen when followed by a **SCREEN:RECeive** command and AM is selected modulation.

MODulation:

AM*n*
Selects Amplitude Modulation. Select 1 or 2 for *n*.

BFO
Selects Beat Frequency Oscillation.

FM*n*
Selects Frequency Modulation. Select 1, 2, 3 or 4 for *n*.

```
Example: REC:MOD:FM4             // Selects FM4 as Receiver Modulation.
```

RECeive:

MODulation:

LSB
Selects Lower Sideband Modulation.

PM
Selects Phase Modulation.

USB
Selects Upper Sideband Modulation.

USER:

FILTer *f*
Sets User Defined Modulation IF Filter to *f* kHz. Select 3, 30 or 300 for *f*.

MODulation:*type*
Selects User Defined Modulation *type* (FM, AM, USB, LSB, BFO, PM or DATA [FM]).

POST:

APASs
Selects All Pass Post Detection Filter for User Defined Modulation.

BPASs *fl,fh*
Selects Bandpass Post Detection Filter for the User Defined Modulation. Lower cutoff frequency is set to *fl* kHz with a range of 0.5 to 20. Higher cutoff frequency is set to *fh* kHz with a range of 0.1 to 30.

CWEight
Selects C-Weighted Post Detection Filter (C-message noise weighting curve response) for the User Defined Modulation. Refer to MIL-STD-188-200.

HPASs *f*
Selects High-Pass Post Detection Filter for the User Defined Modulation. Cutoff frequency is set to *f* kHz with a range of 0.5 to 20.

LPASs *f*
Selects Low-Pass Post Detection Filter for the User Defined Modulation. Cutoff frequency is set to *f* kHz with a range of 0.1 to 30.

Example:

```
REC:MOD:USER:FILT 30     // Selects the 30 kHz IF Filter.
REC:MOD:USER:MOD:DATA    // Selects User Defined FM DATA for Receiver
                         // Modulation.
REC:MOD:USER:POST:BPAS 10,20  // Selects a Bandpass Post Detection
                         // Filter with 10 kHz lower cutoff and 20 kHz
                         // higher cutoff frequencies.
REC:MOD?                 // Queries the Receiver Modulation.  USER is
                         // returned.
```

MODulation?
Returns Receiver Modulation (FM, AM, USB, LSB, BFO, PM, DATA or USER).

RECeive:

OUTput:

AUDio *b*
Routes Demodulated Audio to the AUDIO OUT Connector if *b* is 1. Disconnects the AUDIO OUT Connector if *b* is 0.

DEMOD *b*
Routes Demodulated Audio to the DEMOD OUT Connector if *b* is 1. Disconnects the DEMOD OUT Connector if *b* is 0.

SPEAKer *b*
Routes Demodulated Audio to the Speaker if *b* is 1. Disconnects Speaker if *b* is 0.

PMRms
Displays Phase Meter (RMS) reading when followed by a **SCREEN:RECeive** command.

POCSAG:

CAPcode?
Returns the received capcode or -1 if a capcode is not received.

MESSage?
Returns message from the decoded POCSAG signal or -1 if none available.

RATe *b*
Sets POCSAG rate to high if *b* is 1, low if *b* is 0.

RATe?
Returns 1 if POCSAG rate is high, 0 if rate is low.

STATe *b*
Enables POCSAG decoding if *b* is 1, disables if *b* is 0. **RECeive:DECode DIGital** and **RECeive:DIGital POCSAG** commands must be initiated prior to enabling POCSAG decoding. POCSAG decoding must be disabled after decoding is finished.

NOTE: Do not change screens with POCSAG enabled or Test Set may lock up.

TYPE?
Returns POCSAG Function Type. One of the following is returned:

TONE 1 BEEP	TONE 2 BEEPS	TONE 3 BEEPS
TONE 4 BEEPS	NUMERIC	ALPHANUMERIC
NO MESSAGE		

Example:
```
REC:DEC DIG            // Sets Receiver for decoding digital data.
REC:DIG POCSAG         // Prepares Receiver for decoding POCSAG.
REC:POCSAG:STAT 1      // Enables POCSAG decoding.
REC:POCSAG:RATE 1      // Selects High rate to decode.
REC:POCSAG:MESS?       // Queries for the decoded message.
REC:POCSAG:CAP?        // Queries for the decoded capcode.
REC:POCSAG:TYPE?       // Queries for the decoded message type.
REC:POCSAG:RATE?       // Queries for the current rate setting.
                       // 1 is returned (for High).
REC:POCSAG:STAT 0      // Disables POCSAG decoding.
```

RECeive:

RCL *n*
Recalls Receiver environment (routings and settings) stored in memory location *n*. Range of *n* is 1 to 9.

SCAN:

ABORt
Stops Receiver Scan Function.

CONTinue
Starts or if paused, continues Scan Function.

FREQuency?
Returns frequency currently being scanned in kHz.

INCrement *f*
Sets Receiver Scan increment to *f* kHz. Range of *f* is 0.0 to 999999.9.

PAUSe *t*
Sets Receiver Pause rate to *t* sec. Range of *t* is 0.0 to 99.9. Receiver Scan pause time is length of time Scan Function pauses at frequency with broken squelch. Scan Function stops permanently on squelch broken frequency if pause set to 0.

PAUSe?
Returns 1 if Scan Function is paused, 0 if not paused.

RATe *t*
Sets Receiver Scan rate to *t* sec. Range of *t* is 0.00 to 99.99. Receiver Scan rate is the time each frequency is scanned with squelch unbroken.

STARt *f*
Sets Receiver Scan starting frequency to *f* kHz. Range of *f* is 250.0 to 999999.9.

STOP *f*
Sets Receiver Scan stopping frequency to *f* kHz. Range of *f* is 250.0 to 999999.9.

Example:
```
REC:SCAN:STAR 1000     // Sets Receiver Scan starting frequency to
                       // 1 MHz
REC:SCAN:STOP 100000   // Sets Receiver Scan stopping frequency to
                       // 100 MHz
REC:SCAN:INC 250       // Sets Receiver Scan increment to 250 kHz.
REC:SCAN:RAT 1.5       // Sets Receiver Scan rate to 1.5 sec.
REC:SCAN:PAUS 10       // Sets Receiver Scan pause time to 10 sec.
REC:SCAN:CONT          // Starts Receiver Scan.
REC:SCAN:ABOR          // Stops Receiver Scan.
```

SINAD
Displays SINAD Meter reading when followed by a **SCREEN:RECeive** command.

SQUelch *n*
Sets squelch to *n*. Range of *n* is 0.0 to 1.0.

RECeive:

SQUelch?
Returns squelch level (0.0 to 1.0).

```
Example: SQU .3                 // Sets Squelch level to 3/10 of total
                                // allowable setting.
         SQU?                   // Queries Squelch level.  .3 is returned.
```

STORe *n*
Stores current Receiver environment (routings and settings) in memory location *n*. Range of *n* is 1 to 9.

TONE:

STATe *b*
Enables Audio Tone decoding if *b* is 1, disables if *b* is 0. **RECeive:DECode TONE** command must be initiated prior to enabling Audio Tone decoding. Audio Tone decoding must be disabled after decoding is finished.

TYPE *xxx*
Selects Audio Tone to be decoded. Choose one of the following for *xxx*:

CCIR	EEA	EIA
ZVEI	DDZVEI	DZVEI
NATEL	EURO	TONE56
CCIRH	CCIRH4	USER

See 4-6-1E and G.

TONE?
Returns the decoded Audio Tone sequence or -1 if not available.

```
Example: REC:DEC TONE           // Sets Receiver for decoding Audio Tones.
         REC:TONE:STAT 1        // Enables Audio Tone decoding.
         REC:TONE:TYPE CCIR     // Selects CCIR as the Audio Tone type.
         REC:TONE?              // Returns the decoded Audio Tone sequence.
         REC:TONE:STAT 0        // Disables Audio Tone decoding.
```

VOLume *n*
Sets volume to *n*. Range of *n* is 0.0 to 1.0.

VOLume?
Returns the volume level (0.0 to 1.0).

VOLume:

AUTO *b*
Enables Automatic Volume Control if *b* is 1, disables if *b* is 0.

AUTO?
Returns Automatic Volume Control state. 1 is returned if enabled, 0 if disabled.

CALibration:FMZ *n*
Zeros FM Deviation Meter. Displays the Receiver Operation Screen and selects FM*n* Modulation. Range of *n* is from 1 to 4.

4-6-2 REMOTE RECEIVER EXAMPLES

A. RECEIVING FM SIGNALS

The following command sequence receives a 96 MHz FM signal through the ANTENNA IN Connector and outputs the audio signal through the Test Set Speaker.

```
*DMC "REC_FM",BEGIN              // Define a macro named REC_FM.
SCREEN:REC                       // Display the Receiver Operation Screen.
REC:FREQ 96000                   // Set Receiver Frequency to 96 MHz.
REC:MODULATION:FM1               // Select FM1 for Receiver Modulation.
REC:INPUT:ANTENNA                // Select the ANTENNA IN Connector for
                                 // Receiver Input.
REC:INPUT:ATTENUATION 0          // Select 0 dB for the Receiver Input
                                 // Attenuation Level.
N=1,X=0                          // Set variables to initial values.
WHILE X=0                        // Start WHILE loop to loop as long as
                                 // squelch is not broken.
   REC:SQU N                     // Set squelch 0.02 lower than last setting.
   DELAY 25                      // Allow time for squelch to settle.
   N=N-0.02                      // Decrement variable N (use to set squelch).
   X=:MEAS:SQU?                  // Set X to 1 when squelch breaks (see 4-15).
WEND                             // End of WHILE loop.  X loops to the top as
                                 // long as X=0 (squelch unbroken).
REC:OUTPUT:SPEAKER 1             // Routes Receiver Output to the Test Set
                                 // Speaker.
END                              // End of macro REC_FM.
```

B. RECEIVING AM SIGNALS

The following command sequence receives a 1240 kHz AM signal through the ANTENNA IN Connector and outputs the audio signal through the DEMOD OUT Connector.

```
*DMC "REC_AM",BEGIN              // Define a macro named REC_AM.
SCREEN:REC                       // Display the Receiver Operation Screen.
REC:FREQ 1240                    // Set Receiver Frequency to 96 MHz.
REC:MODULATION:AM1               // Select AM1 for Receiver Modulation.
REC:INPUT:ANTENNA                // Select the ANTENNA IN Connector for
                                 // Receiver Input.
REC:INPUT:ATTENUATION 0          // Select 0 dB for the Receiver Input
                                 // Attenuation Level.
N=1,X=0                          // Set variables initially.
WHILE X=0                        // Start WHILE loop to loop as long as
                                 // squelch is not broken.
   REC:SQU N                     // Set squelch 0.02 lower than last setting.
   DELAY 25                      // Allow time for squelch to settle.
   N=N-0.02                      // Decrement variable N (use to set squelch).
   X=:MEAS:SQU?                  // Set X to 1 when squelch breaks (see 4-15).
WEND                             // End of WHILE loop.  X loops to the top as
                                 // long as X=0 (squelch unbroken).
REC:OUTPUT:DEMOD 1               // Route Receiver Output to the DEMOD OUT
                                 // Connector.
END                              // End of macro REC_AM.
```

C. RECEIVING SSB SIGNALS

The following command sequence receives a 14 MHz LSB signal through the T/R Connector and routes the demodulated signal to the AUDIO OUT Connector.

```
*DMC "REC_SSB",BEGIN          // Define a macro named REC_SSB.
SCREEN:REC                    // Display the Receiver Operation Screen.
REC:FREQ 14000                // Set Receiver Frequency to 14 MHz.
REC:MODULATION:LSB            // Select LSB for Receiver Modulation.
REC:INPUT:TR                  // Select the T/R Connector for Receiver
                              // Input.
REC:INPUT:ATTENUATION 0       // Select 0 dB for the Receiver Input
                              // Attenuation Level.
N=1,X=0                       // Set variables initially.
WHILE X=0                     // Start WHILE loop to loop as long as
                              // squelch is not broken.
   REC:SQU N                  // Set squelch 0.02 lower than last setting.
   DELAY 25                   // Allow time for squelch to settle.
   N=N-0.02                   // Decrement N (use to set squelch).
   X=:MEAS:SQU?               // Set X to 1 when squelch breaks (see 4-15).
WEND                          // End of WHILE loop.  X loops to the top as
                              // long as X=0 (squelch unbroken).
REC:OUTPUT:AUDIO 1            // Routes Receiver Output to the AUDIO OUT
                              // Connector.
END                           // End of macro REC_SSB.
```

D. DECODING DTMF CODED SIGNALS

The following command sequence receives a 450 MHz FM modulated DTMF signal and routes the demodulated signal to the AUDIO OUT Connector and the Test Set Speaker.

```
*DMC "REC_DTMF",BEGIN         // Define a macro named REC_DTMF.
SCREEN:REC                    // Display Receiver Operation Screen.
REC:FREQ 450000               // Set Receiver Frequency to 450 MHz.
REC:MODULATION:FM1            // Select FM1 for Receiver Modulation.
REC:INPUT:TR                  // Select T/R Connector for Receiver Input.
REC:INPUT:ATTENUATION 0       // Select 0 dB for Receiver Input
                              // Attenuation Level.
N=1,X=0                       // Set variables initially.
WHILE X=0                     // Start WHILE loop to loop as long as
                              // squelch is not broken.
   REC:SQU N                  // Set squelch 0.02 lower than last setting.
   DELAY 25                   // Allow time for squelch to settle.
   N=N-0.02                   // Decrement N (use to set squelch).
   X=:MEAS:SQU?               // Set X to 1 when squelch breaks (see 4-15).
WEND                          // End of WHILE loop.  X loops to the top as
                              // long as X=0 (squelch unbroken).
REC:OUTPUT:AUDIO 1            // Route the demodulated signal to the
                              // AUDIO OUT Connector.
REC:OUTPUT:SPEAKER 1          // Route Receiver Output to the Test Set
                              // Speaker.
REC:DEC DTMF                  // Set Receiver for decoding DTMF signals.
REC:DTMF:STAT 1               // Enable DTMF decoding.
REC:DTMF?                     // Return DTMF decoded digits.
END                           // End of macro REC_DTMF.
```

E. DECODING CCIR CODED SIGNALS

The following command sequence receives a 450 MHz FM modulated Audio signal and routes the demodulated signal to the AUDIO OUT Connector and the Test Set Speaker.

```
*DMC "REC_CCIR",BEGIN          // Define a macro named REC_CCIR.
SCREEN:REC                     // Display the Receiver Operation Screen.
REC:FREQ 450000                // Set Receiver Frequency to 450 MHz.
REC:MODULATION:FM1             // Select FM1 for Receiver Modulation.
REC:INPUT:ANT                  // Select the ANTENNA IN Connector for the
                               // Receiver Input.
REC:INPUT:ATTENUATION 0        // Select 0 dB for the Receiver Input
                               // Attenuation Level.
N=1,X=0                        // Set variables initially.
WHILE X=0                      // Start WHILE loop to loop as long as
                               // squelch is not broken.
   REC:SQU N                   // Set squelch 0.02 lower than last setting.
   DELAY 25                    // Allow time for squelch to settle.
   N=N-0.02                    // Decrement N (use to set squelch).
   X=:MEAS:SQU?                // Set X to 1 when squelch breaks (see 4-15).
WEND                           // End of WHILE loop.  X loops to the top as
                               // long as X=0 (squelch unbroken).
REC:OUTPUT:AUDIO 1             // Routes the demodulated signal to the
                               // AUDIO OUT Connector.
REC:OUTPUT:SPEAKER 1           // Routes Receiver Output to the Test Set
                               // Speaker.
REC:TONE:TYPE CCIR             // Selects CCIR for the Audio Code Type.
REC:DEC TONE                   // Sets Receiver for decoding Audio Tones.
REC:TONE:STAT 1                // Enables Audio decoding.
REC:TONE?                      // Returns the decoded Audio Tone sequence.
REC:TONE:STAT 0                // Disables Audio decoding.
END                            // End of macro REC_CCIR.
```

F. DECODING DCS CODED SIGNALS

The following command sequence receives a 450 MHz FM modulated DCS signal and decodes the signal until Soft Function key F1 is pressed. The decoded DCS digits are printed to the Host.

```
*DMC "REC_DCS",BEGIN          // Define a macro named REC_DCS.
VAR CODE                      // Defines variable to hold decoded DCS.
SCREEN:REC                    // Displays the Receiver Operation Screen.
REC:FREQ 450000               // Sets Receiver Frequency to 450 MHz.
REC:MODULATION:FM1            // Selects FM1 for Receiver Modulation.
REC:INPUT:ANT                 // Selects the ANTENNA IN Connector for the
                              // Receiver Input.
REC:INPUT:ATTENUATION 0       // Selects 0 dB for the Receiver Input
                              // Attenuation Level.
N=1,X=0                       // Set variables initially.
WHILE X=0                     // Start WHILE loop to loop as long as
                              // squelch is not broken.
   REC:SQU N                  // Set squelch 0.02 lower than last setting.
   DELAY 25                   // Allows time for squelch to settle.
   N=N-0.02                   // Decrement N (used to set squelch).
   X=:MEAS:SQU?               // Set X to 1 when squelch breaks (see 4-15).
WEND                          // End of WHILE loop.  X loops to the top as
                              // long as X=0 (squelch unbroken).
KEYPAD:CLAIM                  // Directs all Keyboard Input to TMAC so
                              // Operation Screen is not changed.
REC:DEC DIG                   // Sets Receiver for decoding digital data.
REC:DIG DCS                   // Prepares Receiver for decoding DCS.
REC:DCS:STAT 1                // Enable DCS decoding.
WHILE (SYSTEM:KEY? != F1)     // Loop until F1 Key is pressed.
  TPAUSE                      // Allow TMAC to share processor time with
                              // Test Set (see 3-12).
  CODE=REC:DCS:NORM?)         // Decode a Normal DCS digits, if received.
  IF (CODE != "-1")           // If DCS digits are decoded,
    PPRINT CODE               // print them to the Host.
  ENDIF                       // End of IF statement.
WEND                          // End of While loop.
REC:DCS:STAT 0                // Disable DCS decoding.
KEYPAD:UNCLAIM                // Release Keyboard for normal use.
END                           // End of macro REC_DCS.
```

G. DECODING AUDIO USER DEFINED CODED SIGNALS

The following command sequence receives a 450 MHz FM modulated Audio signal, decodes the signal using the RF Generator Audio User Defined Tones and routes the demodulated signal to the AUDIO OUT Connector and the Test Set Speaker.

```
*DMC "REC_AUDIO_USER"          // Define a macro named REC_AUDIO_USER.

                               // DEFINE THE TONES TO BE DECODED

SCREEN:GEN                     // Displays RF Generator Operation Screen.
GEN:TONE:TYPE:USER             // Selects User Defined for the Audio Code.
GEN:TONE:USER:DEF 0,900,150    // Defines 0 Tone at 900 Hz and 150 ms.
GEN:TONE:USER:DEF 1,1100,80    // Defines 1 Tone at 1100 Hz and 80 ms.
GEN:TONE:USER:DEF 2,1200,80    // Defines 2 Tone at 1200 Hz and 80 ms.
GEN:TONE:USER:DEF 3,1300,80    // Defines 3 Tone at 1300 Hz and 80 ms.
GEN:TONE:USER:DEF 4,1400,80    // Defines 4 Tone at 1400 Hz and 80 ms.
GEN:TONE:USER:DEF A,0,40       // Defines A Tone at 0 Hz and 40 ms.
                               //
                               // DECODE THE DEFINED TONES
                               //
SCREEN:REC                     // Displays the Receiver Operation Screen.
REC:FREQ 450000                // Sets Receiver Frequency to 450 MHz.
REC:MODULATION:FM1             // Selects FM1 for Receiver Modulation.
REC:INPUT:ANT                  // Selects the ANTENNA IN Connector for the
                               // Receiver Input.
REC:INPUT:ATTENUATION 0        // Selects 0 dB for the Receiver Input
                               // Attenuation Level.
N=1,X=0                        // Set variables initially.
WHILE X=0                      // Start WHILE loop to loop as long as
                               // squelch is not broken.
   REC:SQU N                   // Set squelch 0.02 lower than last setting.
   DELAY 25                    // Allows time for squelch to settle.
   N=N-0.02                    // Decrement variable N (used to set
                               // squelch).
   X=:MEAS:SQU?                // Set X to 1 when squelch breaks (see 4-15).
WEND                           // End of WHILE loop.  X loops to the top as
                               // long as X=0 (squelch unbroken).
REC:OUTPUT:AUDIO 1             // Routes the demodulated signal to the
                               // AUDIO OUT Connector.
REC:OUTPUT:SPEAKER 1           // Routes Receiver Output to the Test Set
                               // Speaker.
REC:TONE:TYPE USER             // Selects User Defined for the Audio Code
                               // Type.
REC:DEC TONE                   // Sets Receiver for decoding Audio Tones.
REC:TONE:STAT 1                // Enables Audio decoding.
REC:TONE?                      // Returns the decoded Audio Tone sequence.
REC:TONE:STAT 0                // Disables Audio decoding.
END                            // End of macro REC_AUDIO_USER.
```

H. DECODING POCSAG CODED SIGNALS

The following command sequence receives a 450 MHz FM modulated POCSAG signal, decodes the signal and prints the POCSAG messages to the Host until Soft Function Key F1 is pressed.

```
*DMC "REC_POCSAG",BEGIN          // Define a macro named REC_POCSAG.
STRING CODE                      // Define a string variable to hold decoded
                                 // POCSAG message.
SCREEN:REC                       // Displays the Receiver Operation Screen.
REC:FREQ 450000                  // Sets Receiver Frequency to 450 MHz.
REC:MODULATION:FM1               // Selects FM1 for the Receiver Modulation.
REC:INPUT:ANT                    // Selects the ANTENNA IN Connector for the
                                 // Receiver Input.
REC:INPUT:ATTENUATION 0          // Selects 0 dB for the Receiver Input
                                 // Attenuation Level.
N=1,X=0                          // Set variables initially.
WHILE X=0                        // Start WHILE loop to loop as long as
                                 // squelch is not broken.
   REC:SQU N                     // Set squelch 0.02 lower than last setting.
   DELAY 25                      // Allows time for squelch to settle.
   N=N-0.02                      // Decrement variable N (used to set
                                 // squelch).
   X=:MEAS:SQU?                  // Set X to 1 when squelch breaks (see 4-15).
WEND                             // End of WHILE loop.  X loops to the top as
                                 // long as X=0 (squelch unbroken).
REC:OUTPUT:SPEAKER 1             // Routes Receiver Output to the Test Set
                                 // Speaker.
REC:POCSAG:RAT 1                 // Sets POCSAG rate to be decoded to High.
KEYPAD:CLAIM                     // Directs all Keyboard Input to TMAC so
                                 // Operation Screen is not changed.
REC:DEC DIG                      // Sets Receiver for decoding digital data.
REC:DIG POCSAG                   // Prepares Receiver for decoding POCSAG.
REC:POCSAG:STAT 1                // Enable POCSAG decoding.
WHILE (SYSTEM:KEY? != F1)        // Loop until F1 Key is pressed.
  TPAUSE                         // Allow TMAC to share processor time with
                                 // Test Set (see 3-12).
  CODE=STR(REC:POCSAG:MESS?)     // Decode a POCSAG message, if received.
  IF (CODE != "-1")              // If a POCSAG message was decoded,
    PPRINT CODE                  // print message to the Host.
  ENDIF                          // End of IF statement.
WEND                             // End of While loop.
REC:POCSAG:STAT 0                // Disable POCSAG decoding.
KEYPAD:UNCLAIM                   // Release Keyboard for normal use.
END                              // End of macro REC_POCSAG.
```

I. RECEIVER SETUP FOR CELLULAR OPERATION

The following command sequence configures the Receiver Mode for receiving Cellular signals through the ANTENNA IN Connector.

```
SCREEN:REC                      // Displays the Receiver Operation Screen.
REC:MOD:USER:MOD:DATA           // Selects User Defined FM Data for the
                                // Receiver Modulation.
REC:MOD:USER:FILT 30            // Selects the 30 kHz IF Filter.
REC:MOD:USER:POST:LPAS 15       // Selects Low-Pass Post Detection Filter
                                // with a cutoff frequency of 15 kHz.
REC:AGC:USER:HIGH               // Selects User Defined High Speed for the
                                // AGC setting.
REC:INPUT:ANTENNA               // Selects the Antenna for Receiver Input.
M_DEV:RANG:UPP 10               // Sets Deviation Meter Range to 10 kHz.
M_AF:RES 1                      // Sets AF Meter Gate Time to .1 sec (1 Hz).
```

4-7 DUPLEX

4-7-1 DUPLEX GENERAL COMMANDS

DUPlex:

METER:

DISTortion
Displays Distortion Meter on Duplex Operation Screen.

MODMeter
Displays Modulation Meter on Duplex Operation Screen.

OFF
Disables Modulation, Distortion and SINAD Meters.

SINAD
Displays SINAD Meter on Duplex Operation Screen.

SPEAKer:SOURce *type*
Selects Test Set Speaker Input. Select OFF, FGEN (Function Generator), SINAD (SINAD/BER IN Connector) or EXTMOD (EXT MOD IN Connector) for *type*. Command takes effect when RF Generator Operation Screen is updated.

STORe *n*
Stores current Duplex environment (routings and settings) in memory location *n*. Range of *n* is 1 to 9.

RCL *n*
Recalls Duplex environment (routings and settings) stored in memory location *n*. Range of *n* is 1 to 9.

4-7-2 DUPLEX TRANSMITTER COMMANDS

DUPlex:

INPut:

AGC:

AUTO
Sets Automatic Gain Control to automatic setting.

MANual *n*
Sets Automatic Gain Control to manual setting and sets level to *n*. Range of *n* is 0 to 255.

USER:*XXX*
Sets Automatic Gain Control to User setting. Select one of the following for *XXX*: MEASure, SPeech, DATA, HIGH, TYPE1, TYPE2 or TYPE3.

Example: `DUP:INP:AGC:USER:DATA // Sets AGC setting to User Defined Data.`

DUPlex:

INPut:

ANTenna
Selects ANTENNA IN Connector as Transmitter Input Source. Displays Signal Strength Meter reading on the Duplex Transmitter Operation Screen.

ATTenuation *n*
Sets Input Attenuation to *n* dB. Select 0, 20 or 40.

CHANnel *n*
Sets Duplex Transmitter Frequency to cellular channel *n* (1 to 1023) in the format selected using the **DUPlex:INPut:CHANnel:FORMat** command. The Duplex Transmitter Operation Screen displays selected cellular channel when Duplex Transmitter is in Channel Mode (**DUPlex:INPut:MODE CHAN** command).

CHANnel:

FORMat:

AMPS:

FORward
Selects AMPS Forward channels.

REVerse
Selects AMPS Reverse channels.

ETACS

FORward
Selects ETACS Forward channels.

REVerse
Selects ETACS Reverse channels.

NAMPS:

FORward
Selects NAMPS Forward channels.

REVerse
Selects NAMPS Reverse channels.

NT400:

FORward
Selects NT400 Forward channels.

REVerse
Selects NT400 Reverse channels.

DUPlex:

INPut:

CHANnel:FORMat?
Returns channel format setting (AMPS:FORWARD, AMPS:REVERSE, ETACS:FORWARD, ETACS:REVERSE, NAMPS:FORWARD, NAMPS:REVERSE, NT400:FORWARD or NT400:REVERSE).

FIND:

FREQuency?
Returns frequency of first signal with amplitude larger than the Find reference level.

REFerence *n*
Sets Find reference level to *n* dB.

REFerence?
Returns Find reference level in dB.

Example:

```
DUP:INP:FIND:REF -55        // Sets Find Reference Level to -55 dBm.
DUP:INP:FIND:FREQ?          // Returns the lowest frequency (in kHz)
                            // containing a signal greater than -55 dBm.
DUP:INP:FIND:REF?           // Queries the current Find Reference Level
                            // (in dBm).  -55 is returned.
```

FREQuency *n* [*units*]
Sets Duplex Transmitter Frequency. Range of *n* is 250.0 kHz to 999.9999 MHz. Select HZ, KHZ, MHZ for *units*. *units* is optional with a default of KHZ.

FREQuency?
Returns RF Generator Frequency in kHz.

Example:

```
DUP:INP:FREQ 145 MHZ  // Set RF Generator Frequency to 145 MHz.
DUP:INP:FREQ?         // Query the RF Generator Frequency.
                      // 145000 is returned (145000 kHz).
```

METER:

DEVRms
Displays Deviation Meter (RMS) when followed by a **SCREEN:DUPlex** command.

DISTortion
Displays Distortion Meter when followed by a **SCREEN:DUPlex** command.

MODMeter
Displays Modulation Meter when followed by a **SCREEN:DUPlex** command.

PMRms
Displays Phase Meter (RMS) when followed by a **SCREEN:DUPlex** command.

SINAD
Displays SINAD Meter when followed by a **SCREEN:DUPlex** command.

DUPlex:

INPut:

MODE *type*
Selects the Duplex Transmitter Mode (DIRect or CHANnel). Channel Mode displays cellular channel frequency according to **DUP:INP:CHAN** commands.

MODulation:

AM*n*
Selects Amplitude Modulation. Select 1 or 2 for *n*.

Example: `DUP:INP:MOD:AM1  // Selects AM1 for Duplex Transmitter // Modulation.`

BFO
Selects Beat Frequency Oscillation.

FM*n*
Selects Frequency Modulation. Select 1, 2, 3 or 4 for *n*.

LSB
Selects Lower Side Band Modulation.

PM
Selects Phase Modulation.

USB
Selects Upper Side Band Modulation.

USER:

FILTer *f*
Sets User Defined Modulation IF Filter to *f* kHz. Select 3, 30 or 300 for *f*.

MODulation:*type*
Selects User Defined Modulation. Enter one of the following for *type*: FM, AM, USB, LSB, BFO, PM or DATA (FM DATA).

POST:

APASs
Selects All Pass Post Detection Filter for User Defined Modulation.

BPASs *fl,fh*
Selects Bandpass Post Detection Filter for User Defined Modulation. Lower cutoff frequency is set to *fl* kHz with range of 0.5 to 20. Higher cutoff frequency is set to *fh* kHz with a range of 0.1 to 30.

CWEight
Selects C-Weighted Post Detection Filter (C-message noise weighting curve response) for the User Defined Modulation. Refer to MIL-STD-188-200.

DUPlex:

INPut:

MODulation:

USER:

HPASs *f*
Selects High-Pass Post Detection Filter for User Defined Modulation. Cutoff frequency is set to *f* kHz with a range of 0.5 to 20.0.

LPASs *f*
Selects Low-Pass Post Detection Filter for User Defined Modulation. Cutoff frequency is set to *f* kHz with a range of 0.1 to 30.0.

MODulation?
Returns the current Duplex Transmitter Modulation (FM, AM, USB, LSB, BFO, PM, DATA or USER).

Example:
```
DUP:INP:MOD:USER:FILT 3          // Selects the 3 kHz IF Filter.
DUP:INP:MOD:USER:FM              // Selects User Defined FM for the
                                 // Duplex Transmitter Modulation.
DUP:INP:MOD:USER:POST:LPAS 15 // Selects a Low-Pass Post Detection
                                 // Filter with 15 kHz cutoff
                                 // frequency.
DUP:INP:MOD:USER:MOD?            // Queries the Duplex Transmitter
                                 // Modulation.  User is returned.
```

TO:

AUDIo *b*
Routes demodulated Receiver Input to the AUDIO OUT Connector if *b* is 1. Disconnects the AUDIO OUT Connector if *b* is 0.

DEMOD *b*
Routes demodulated Receiver Input to the DEMOD OUT Connector if *b* is 1. Disconnects the DEMOD OUT Connector if *b* is 0.

SPEAKer *b*
Routes demodulated Receiver Input to the Test Set Speaker if *b* is 1. Disconnects the Test Set Speaker if *b* is 0.

TR
Selects T/R Connector as Duplex Transmitter Input Source. Displays Power Meter on the Duplex Transmitter Operation Screen.

VOLume:

AUTO *b*
Enables Automatic Volume Control if *b* is 1, disables if *b* is 0.

AUTO?
Returns Automatic Volume Control state. 1 is returned if enabled, 0 if disabled.

4-7-3 DUPLEX RECEIVER COMMANDS

DUPlex:

OUTput:

AUDio *b*
Routes AF Generator Output to the AUDIO OUT Connector if *b* is 1. Disconnects the AUDIO OUT Connector if *b* is 0.

CHANnel *n*
Sets Duplex Receiver Frequency to cellular channel *n* (1 to 1023) in the format selected using the **DUPlex:OUTut:CHANnel:FORMat** command. The Duplex Receiver Operation Screen displays selected cellular channel when Duplex Receiver is in Channel Mode (**DUPlex:OUTput:MODE CHAN** command).

CHANnel:

FORMat:

AMPS:

FORward
Selects AMPS Forward channels.

REVerse
Selects AMPS Reverse channels.

ETACS

FORward
Selects ETACS Forward channels.

REVerse
Selects ETACS Reverse channels.

NAMPS:

FORward
Selects NAMPS Forward channels.

REVerse
Selects NAMPS Reverse channels.

NT400:

FORward
Selects NT400 Forward channels.

REVerse
Selects NT400 Reverse channels.

DUPlex:

OUTput:

CHANnel:FORMat?
Returns channel format setting (AMPS:FORWARD, AMPS:REVERSE, ETACS:FORWARD, ETACS:REVERSE, NAMPS:FORWARD, NAMPS:REVERSE, NT400:FORWARD or NT400:REVERSE).

DEMOD *b*
Routes AF Generator Output to the DEMOD OUT Connector if *b* is 1. Disconnects the DEMOD OUT Connector if *b* is 0.

DUPlex
Routes RF Generator Output to the DUPLEX OUT Connector and disconnects the T/R Connector.

FREQuency *n* [*units*]
Sets Duplex Receiver Frequency. Range of *n* is 250.0 kHz to 999.9999 MHz. Select HZ, KHZ, MHZ for *units*. *units* is optional with a default of KHZ.

FREQuency?
Returns Duplex Receiver Frequency in kHz.

```
Example: DUP:OUT:FREQ 900        // Set RF Generator Frequency to 900 kHz.
         DUP:OUT:FREQ?           // Query the RF Generator Frequency.
                                 // 900 is returned (900 kHz).
```

LEVel:

DBm *n*
Sets RF Level to *n* dBm. Range of *n* is 0.0 to -137.0

DBm?
Returns RF Level in dBm.

```
Example:DUP:OUT:LEV:DB -40  // Sets Duplex Receiver Output Level to
                            // -40 dBm.
DUP:OUT:DB?                 // Queries Duplex Receiver Output Level.
                            // -40 is returned.
```

DUPlex:

OUTput:

METER:

NOTE: **DUPlex:METER** commands were created to avoid meter operation and display conflicts due to measurement signal routings and overlapping applications.

AF
Displays AF Meter on the Duplex Receiver Operation Screen.

DISTortion
Displays Distortion Meter on the Duplex Receiver Operation Screen.

DMM
Displays Digital Multimeter on the Duplex Receiver Operation Screen.

SINAD
Displays SINAD Meter on the Duplex Receiver Operation Screen.

MODE *type*
Selects the Duplex Receiver Mode (DIRect [Direct Mode] or CHANnel [Channel Mode]). Channel Mode displays cellular channel frequency according to **DUP:OUT:CHAN** commands.

OFFSet *f*
Sets Offset Frequency to *f* kHz. Range of *f* is -999749.9 to 999749.9.

OFFSet?
Returns Offset Frequency in kHz.

TR
Routes Duplex Receiver Output to the T/R Connector and disconnects DUPLEX OUT Connector.

4-7-4 REMOTE DUPLEX EXAMPLES

A. GENERATING AND RECEIVING FM SIGNALS

The following sequence of commands generates a 160 MHz FM modulated signal routed to the DUPLEX OUT Connector and receives a 165 MHz FM modulated signal through the ANTENNA IN Connector:

```
SCREEN:DUPLEX                 // Displays the Duplex Operation Screen
DUP:OUT:FREQ 160000           // Sets 160 MHz Duplex Receiver Frequency.
DUP:OUT:DUP                   // Routes the DUPLEX OUT Connector for
                              // the Duplex Receiver Output.
DUP:OUT:LEV:DB -30            // Sets Duplex Receiver Output Level to
                              // -30 dBm.
FGEN:GEN1:STATE 1             // Activates SOURCE 1 (AF Generator 1).
FGEN:GEN1:FREQ 2000           // Sets SOURCE 1 AF frequency to 2000 Hz.
FGEN:GEN1:MOD:FM              // Sets SOURCE 1 Modulation to FM.
FGEN:GEN1:MODL 5              // Sets SOURCE 1 Modulation Level to 5 kHz.
FGEN:GEN1:SHAPE:SIN           // Sets SOURCE 1 Wave Shape to a sine wave.
DUP:INP:FREQ 165000           // Sets Duplex Transmitter Frequency to
                              // 165 MHz.
DUP:INP:ANT                   // Selects ANTENNA IN Connector for Duplex
                              // Transmitter Input.
DUP:INP:ATT 0                 // Sets Duplex Transmitter Input Attenuation
                              // Level to 0 dB.
DUP:INP:MOD:FM1               // Selects FM1 for the Duplex Transmitter
                              // Modulation type.
```

B. GENERATING AND RECEIVING AM SIGNALS

The following sequence of commands generates a 1.4 MHz AM signal (50% modulation) modulated with Input from the MIC/ACC IN/OUT Connector and receives a 1 MHz AM modulated signal through the ANTENNA IN Connector:

```
SCREEN:DUPLEX                 // Displays the Duplex Operation Screen
DUP:OUT:FREQ 140000           // Sets 140 MHz Duplex Receiver Frequency.
DUP:OUT:DUP                   // Selects the DUPLEX OUT Connector for
                              // the Duplex Receiver Output.
DUP:OUT:LEV:DB -30            // Sets Duplex Receiver Output Level to
                              // -30 dBm.
FGEN:MIC:STATE 1              // Activates SOURCE 1 (AF Generator 1).
FGEN:MIC:MOD:AM1              // Sets SOURCE 1 Modulation to AM1.
FGEN:MIC:MODL 50              // Sets SOURCE 1 Modulation Level to 50%.
PTT:STATE 1                   // Enables MIC/ACC Connector.
DUP:INP:FREQ 1 MHZ            // Sets Duplex Transmitter Frequency to
                              // 1 MHz.
DUP:INP:ANT                   // Selects ANTENNA IN Connector for Duplex
                              // Transmitter Input.
DUP:INP:ATT 0                 // Sets Duplex Transmitter Input Attenuation
                              // Level to 0 dB.
DUP:INP:MOD:AM1               // Selects AM1 for the Duplex Transmitter
                              // Modulation type.
```

4-8 AF GENERATOR

4-8-1 AF GENERATOR COMMANDS

FGEN:

DATA:
(Data Generator is used for BER Meter tests.)

MODulation:*type*
Sets Data Generator Modulation to *type*. Select OFF, AM, FM or PM.

MODulation?
Returns Data Generator Modulation.

MODL *n*
Sets Data Generator to *n* Modulation level. Range of *n* for AM is 0.0 to 100.0 (%), FM is 0.0 to 25.0 (kHz) and PM is 0.0 to 10.0 (radians).

MODL?
Returns Data Modulation level setting.

STATe *b*
Sets Data Modulation on if *b* is 1 or off if *b* is 0.

Example:
```
FGEN:DATA:MOD:FM      // Sets Data Generator Modulation to FM.
FGEN:DATA:MODL 4      // Sets Data Generator Modulation Level to
                      // 4 kHz.
FGEN:DATA:STAT 1      // Enables Data Generator.
FGEN:DATA:MOD?        // Returns Data Generator Modulation (FM).
FGEN:DATA:MODL?       // Returns Data Generator Modulation Level.
                      // 4 (kHz) is returned.
```

EXT:
(External modulation is input through EXT MOD IN Connector.)

LEVel *n*
Sets External Modulation Proportional output level to *n*%. Range of *n* is 0 to 100. Sources are adjusted so that proportional output levels of sources combine to equal Function Generator Output Level. Command has no effect if Proportional Mode off.

LEVel?
Returns External Modulation Proportional output level setting in %.

MODL *n*
Sets External Modulation level to *n*. Range of *n* for AM is 0 to 100 (%), FM is 0.0 to 10.0 (kHz) and PM is 0.0 to 10.0 (radians).

MODL?
Returns External Modulation level setting.

MODulation: *type*
Sets External Modulation to *type*. Select OFF, AM, FM or PM.

NOTE: Do not set different AF Generators to FM and PM simultaneously because hardware limitations may cause ambiguous readings.

FGEN:

EXT:

MODulation?
Returns External Modulation type (OFF, AM, FM or PM).

STATe *b*
Sets External Modulation on if *b* is 1 or off if *b* is 0.

Example:
```
FGEN:EXT:MOD:AM        // Sets External (EXT MOD) Modulation Type
                       // to AM.
FGEN:EXT:MODL 50       // Sets External Modulation Level to 50%.
FGEN:PROP 1            // Selects Proportional Mode for the AF
                       // Generator.
FGEN:EXT:LEV 30        // Sets External Output Level to 30%.
FGEN:EXT:STAT 1        // Enables External (EXT MOD) Modulation.
FGEN:EXT:MOD?          // Queries External (EXT MOD) Modulation
                       // Type.  AM is returned.
FGEN:EXT:MODL?         // Queries External (EXT MOD) Modulation
                       // Level.  50% is returned.
FGEN:EXT:LEV?          // Queries External (EXT MOD) Output Level.
                       // 30 is returned.
```

Commands for Generator 1 and Generator 2 are identical and are listed as **FGEN:GEN***x*. Parameter *x* is specified as 1 or 2 when the command is entered.

GEN*x*:

FREQuency *f*
Sets AF Generator to *f* Hz. Range of *f* is 0.0 to 40000.0.

FREQuency?
Returns AF Generator in Hz.

LEVel *n*
Sets Proportional output level of AF Generator to *n*%. Range of *n* is 0 to 100. Sources are adjusted so that proportional output levels of sources combine to equal Function Generator Output Level. Command has no effect if Proportional Mode off.

LEVel?
Returns Proportional output level setting of AF Generator in %.

MODL *n*
Sets Modulation level to *n*. Range of *n* for AM is 0 to 100 (%), FM is 0.0 to 100.0 (kHz) and PM is 0.0 to 10.0 (radians).

MODL?
Returns AF Generator Modulation level setting.

MODulation:*type*
Sets AF Generator Modulation to *type*. Select OFF, AM, FM or PM for *type*.

NOTE: Do not set different AF Generators to FM and PM simultaneously because hardware limitations may cause ambiguous readings.

FGEN:

GEN*x*:

MODulation?
Returns AF Generator Modulation Type.

SHAPE:*type*
Sets AF Generator wave shape to *type*. Select from SIN (Sine), SQU (Square Wave), RAMP or TRI (Triangle).

SHAPE:

DC *n*
Sets wave shape to DC at *n* level. Select -1, 0 or 1 for *n*.

PULse:DCYCLe 50
Sets AF Generator wave shape to Pulse with 50% duty cycle.

STATe *b*
AF Generator is enabled if *b* is 1, disabled if *b* is 0.

Example:

```
                                // AF Generator setup as RF Generator
                                // Modulation Source (AF Generator 1 and 2).
FGEN:GEN1:FREQ 1000             // Sets AF Generator 1 frequency to 1000 Hz.
FGEN:GEN1:MOD:FM                // Selects FM for AF Generator 1 Modulation
                                // Type.
FGEN:GEN1:MODL 5                // Sets AF Generator 1 Modulation Level to
                                // 5 kHz.
FGEN:GEN1:SHAPE:SIN             // Selects a sine wave for AF Generator 1
                                // Wave Form.
FGEN:GEN1:STAT 1                // Enables AF Generator 1.
FGEN:GEN2:FREQ 2500             // Sets AF Generator 2 frequency to 2500 Hz.
FGEN:GEN2:MOD:AM                // Selects AM for AF Generator 2 Modulation
                                // Type.
FGEN:GEN2:MODL 25               // Sets AF Generator 2 Modulation Level to
                                // 25%.
FGEN:GEN2:SHAPE:SIN             // Selects a sine wave for AF Generator 2
                                // Wave Form.
FGEN:GEN2:STAT 1                // Enables AF Generator 2.
```

Example:

```
                                // AF Generator setup as AF Signal Generator
FGEN:GEN1:FREQ 1000             // Sets AF Generator 1 frequency to 1000 Hz.
FGEN:OUT:PROP 1                 // Enables AF Generator Proportional Mode.
FGEN:GEN1:LEV 50                // Sets AF Generator 1 Proportion Level to
                                // 50%.
FGEN:GEN1:SHAPE:SIN             // Selects a sine wave for AF Generator 1
                                // Wave Form.
FGEN:GEN1:STAT 1                // Enables AF Generator 1.
FGEN:GEN2:FREQ 2500             // Sets AF Generator 2 frequency to 2500 Hz.
FGEN:GEN2:LEV 100               // Sets AF Generator 2 Proportion Level to
                                // 100%.
FGEN:GEN2:SHAPE:SIN             // Selects a sine wave for AF Generator 2
                                // Wave Form.
FGEN:GEN2:STAT 1                // Enables AF Generator 2.
```

FGEN:

GEN3:

DIGital *type*
Sets digital *type*. Used with the **FGEN:ENCode DIGital** command to prepare Generator 3 encoding. The setting for *type* is DCS, DCSINV, POCSAG, DSAT or DST.

ENCode *type*
Selects a signalling format to encode. Select DTMF, TONE, DIGital or RCC for *type*. Must be followed with a **SETUP:GEN** or **SCREEN:GEN** command.

MODL *n*
Sets AF Generator 3 Modulation level. Range of *n* for AM is 0 to 100.0 (%), PM is 0.0 to 10.0 (radians), FM is 0.0 to 100.0 (kHz)-Tone or RCC signalling format, FM is 0.0 to 10.0 (kHz)-DTMF signalling format or FM is 0.0 to 25.0 (kHz)-Digital signalling format.

MODL?
Returns AF Generator 3 Modulation level setting.

MODulation:*type*
Sets AF Generator 3 Modulation to *type*. Select OFF, AM, FM or PM.

NOTE: Do not set different AF Generators to FM and PM simultaneously because hardware limitations may cause ambiguous readings.

MODulation?
Returns AF Generator 3 Modulation type.

Example:
```
FGEN:GEN3:MOD:FM // Selects FM for AF Generator 3 Modulation type.
FGEN:GEN3:MODL 4 // Sets AF Generator 3 Modulation Level to 4 kHz.
FGEN:GEN3:STAT 1 // Enables AF Generator 3.
```

MIC:
(External audio modulation is input through MIC/ACC Connector.)

LEVel *n*
Sets Proportional output level of MIC/ACC Connector input to *n*%. Range of *n* is 0 to 100. Sources are adjusted so that proportional output levels of sources combine to equal Function Generator Output Level. Command has no effect if Proportional Mode off.

LEVel?
Returns Proportional output level setting for MIC/ACC Connector input.

MODL *n*
Sets MIC/ACC Connector input Modulation level to *n*. Range of *n* for AM is 0 to 100 (%), FM is 0.0 to 25.0 (kHz) and PM is 0.0 to 10.0 (radians).

MODL?
Returns MIC/ACC Connector input Modulation level setting.

MODulation:*type*
Sets MIC/ACC Connector input Modulation to *type*. Select OFF, AM, FM or PM.

NOTE: Do not set different AF Generators to FM and PM simultaneously because hardware limitations may cause ambiguous readings.

FGEN:

MIC:

MODulation?
Returns MIC/ACC Connector input Modulation type.

STATe *b*
Enables external modulation through MIC/ACC Connector if *b* is 1, disables if *b* is 0.

PTT:STATe *b*
Sets the push to talk pin on the MIC/ACC Connector high (keys the mic) if *b* is 1, sets the push to talk pin low if *b* is 0.

Example:
```
FGEN:MIC:MOD:AM       // Sets External (MIC/ACC) Modulation Type
                      // to AM.
FGEN:MIC:MODL 25      // Sets External (MIC/ACC) Modulation Level
                      // to 25%.
FGEN:OUT:PROP 1       // Selects Proportional Mode for the AF
                      // Generator.
FGEN:MIC:LEV 50       // Sets External (MIC/ACC) Output Level
                      // to 50%
FGEN:MIC:STAT 1       // Enables External (MIC/ACC) Modulation.
FGEN:MIC:MOD?         // Queries External (MIC/ACC) Modulation
                      // Type.  AM is returned.
FGEN:MIC:MODL?        // Queries External (MIC/ACC) Modulation
                      // Level.  25% is returned.
FGEN:MIC:LEV?         // Queries External (MIC/ACC) Output Level.
                      // 50 is returned.
```

FGEN:

OUTput:

AUDio *b*
Routes AF Generator Output to the AUDIO OUT Connector if *b* is 1, disconnects AUDIO OUT Connector from Output if *b* is 0.

AUDio?
Returns 1 if AF Generator Output is routed to the AUDIO OUT Connector, 0 if AUDIO OUT Connector is disconnected.

DEMOD *b*
Routes AF Generator Output to DEMOD OUT Connector if *b* is 1, disconnects DEMOD OUT Connector from Output if *b* is 0.

DEMOD?
Returns 1 if AF Generator Output is routed to DEMOD OUT Connector, 0 if DEMOD OUT Connector is disconnected.

LEVel *v*
Sets AF Generator output level to *v* volts. Range for *v* is 0.0000 to 3.1000.

FGEN:

OUTput:

LEVel?
Returns AF Generator output level in volts.

SPEAKer *b*
Routes AF Generator to Speaker if *b* is 1, disconnects Speaker from Output if *b* is 0.

SPEAKer?
Returns 1 if AF Generator Output is routed to Speaker, 0 if Speaker is disconnected.

PROPortional *b*
Selects Proportional mode for AF Generator if *b* is 1. Takes AF Generator out of Proportional mode if *b* is 0.

PROPortional?
Returns 1 if AF Generator is in Proportional mode; 0 otherwise.

Example:
```
FGEN:GEN1:FREQ 500      // Sets AF Generator 1 frequency to 500 Hz.
FGEN:OUT:PROP 1         // Enables Proportional Mode of AF Generator.
FGEN:GEN1:LEV 100       // Sets AF Generator 1 Proportion Level to
                        // 100%.
FGEN:GEN1:SHAPE:RAMP    // Selects ramp for the AF Generator 1
                        // Wave Form.
FGEN:GEN1:STAT 1        // Enables AF Generator 1.
FGEN:GEN2:FREQ 2000     // Sets AF Generator 2 frequency to 2000 Hz.
FGEN:GEN2:LEV 100       // Sets AF Generator 2 Proportion Level to
                        // 100%.
FGEN:GEN2:SHAPE:SIN     // Selects a sine wave for AF Generator 2
                        // Wave Form.
FGEN:GEN2:STAT 1        // Enables AF Generator 2.
FGEN:OUT:LEV 1.5        // Sets AF Generator Output Level to 1.5 V.
FGEN:OUT:AUDIO 1        // Routes AF Generator Output to AUDIO OUT
                        // Connector.
```

RCL *n*
Recalls AF Generator environment (routings and settings) stored in memory location *n*. Range of *n* is 1 to 9.

STORe *n*
Stores current AF Generator environment (routings and settings) in memory location *n*. Range of *n* is 1 to 9.

4-8-2 REMOTE AF GENERATOR EXAMPLES

The following command sequence generates a 1 kHz sine wave, with 1 V level, and routes the Output to the DEMOD OUT Connector:

```
SCREEN:FUNC                 // Displays the AF Generator Operation Screen.
FGEN:GEN1:STATE 1           // Enables AF Generator 1.
FGEN:GEN1:FREQ 1000         // Sets AF Generator 1 frequency to 1000 Hz.
FGEN:GEN1:SHAPE:SIN         // Selects a sine wave for the AF Generator 1
                            // wave form.
FGEN:LEVEL 1                // Sets AF Generator 1 Output Level to 1 V.
FGEN:OUTPUT:DEMOD 1         // Routes AF Generator Output to the DEMOD OUT
                            // Connector.
```

The following command sequence generates the sum of a 1 kHz and 2.5 kHz sine wave in equal proportions and routes this Output to the AUDIO OUT Connector:

```
SCREEN:FUNC                 // Displays the AF Generator Operation Screen.
FGEN:PROPORTIONAL 1         // Selects Proportional Mode for the AF Generator.
FGEN:OUTPUT:LEVEL 1         // Sets AF Generator 1 Output Level to 1 V.
FGEN:GEN1:STATE 1           // Enables AF Generator 1.
FGEN:GEN1:FREQ 1000         // Sets AF Generator 1 frequency to 1000 Hz.
FGEN:GEN1:SHAPE:SIN         // Selects a sine wave for the AF Generator 1
                            // wave form.
FGEN:GEN1:LEVEL 100         // Sets AF Generator 1 Proportional Level to 100%.
FGEN:GEN2:STATE 1           // Enables AF Generator 2.
FGEN:GEN2:FREQ 2500         // Sets AF Generator 2 frequency to 2500 Hz.
FGEN:GEN2:SHAPE:SIN         // Selects a sine wave for the AF Generator 2
                            // wave form.
GEN:GEN2:LEVEL 100          // Sets AF Generator 2 Proportional Level to 100%.
FGEN:OUTPUT:AUDIO 1         // Routes AF Generator Output to the AUDIO OUT
                            // Connector.
```

4-9 OSCILLOSCOPE

4-9-1 OSCILLOSCOPE COMMANDS

SCOPe:

ARM
Arms Oscilloscope. Sets Oscilloscope for one sweep. Command ignored unless Trigger is set to One Shot (see **SCOPe:TRIGger** commands).

AVErage *n*
Selects Oscilloscope Average Mode using *n* samples. Range of *n* is 1 to 100 (default is 100).

COMPare *n*
Selects Compare Mode for Oscilloscope. Trace stored at *n* memory location is compared to the current Live Trace. If *n* is not specified, the lowest numbered memory location with a stored Trace is used.

Example:
```
SCOP:COMP 2   // Displays the Trace stored in memory location 2 on
              // the CRT with the current live Trace.  All
              // Oscilloscope settings affect the live Trace only.
```

COUPling *type*
Selects *type* for External coupling. Select AC, DC or GROund.

FULL
Selects a full size Oscilloscope display for the RF Generator, Receiver and Duplex Operation Screens.

HORIZontal *n*
Sets Horizontal Time Offset to *n* major divisions. *n* is an integer from -12 to 12. -12 to -1 are major divisions before the trigger. 1 to 12 are major divisions after the trigger.

INPut:

FILTer:

CWEight:STATe *b*
Enables Internal C-Weight filter if *b* is 1, disables if *b* is 0.

HPASs:

FREQuency *f*
Sets Internal High-Pass frequency to *f* kHz. Range of *f* is 0.2 to 100.

STATe *b*
Enables Internal High-Pass filter if *b* is 1, disables if *b* is 0.

LPASs:

FREQuency *f*
Sets Internal Low-Pass frequency to *f* kHz. Range of *f* is 0.2 to 50.

STATe *b*
Enables Internal Low-Pass filter if *b* is 1, disables if *b* is 0.

SCOPe:

INPut:

FILTer:

NOTch:

FREQuency *f*
Sets Notch center frequency to *f* kHz. Range of *f* is 0.5 to 1.5.

STATe *b*
Enables Internal Notch Filter if *b* is 1, disables if *b* is 0.

Example:

```
SCOP:INP:FILT:NOT:FREQ 1.5  // Sets Oscilloscope Input Notch Filter
                            // center frequency to 1.5 kHz.
SCOP:INP:FILT:NOT:STAT 1    // Enables Internal Notch Filter.
```

INTernal *type*
Selects *type* as Internal Oscilloscope Input. Select from:

IF (Rcvr IF)	DEMOD (Demod Audio)	POWer (RF Pwr Lvl)
SINAD (SINAD/BER)	FUNCtion (Func Gen)	XAUDIO (Ext Mod)

LEVel *n*
Sets Trigger level to *n*. Range of *n* is 0 to 255 with 0 corresponding to the bottom of the Oscilloscope Display and 255 corresponding to the top.

LIVe *b*
Selects Live Trace Mode for the Oscilloscope. If *b* is 1, a **SCREEN:SCOPE** command is performed. If *b* is 0, no **SCREEN:SCOPE** command is performed. Default of optional *b* is 1.

MARKer:

AOFF
Disables both Markers.

DELTA:

AMPLitude?
Returns voltage difference of the Trace Marker 1 and Trace Marker 2 crossings in volts. Valid only for Oscilloscope Inputs AC, DC and GND.

POINt?
Returns the difference of Marker positions in graticules with 100 graticules equal to the Oscilloscope display width.

TIME?
Returns the difference of the two Marker positions in ms.

TRACK *b*
Enables Marker Tracking if *b* is 1, disables Marker Tracking if *b* is 0. Tracking feature keeps Markers a constant distance apart.

Commands for Marker 1 and Marker 2 are identical and are listed as **SCOPe:MARKER***x* commands. Parameter *x* is specified as 1 or 2 when the command is entered.

SCOPe:

MARKER*x*:

AMPLitude?
Returns voltage of live Trace at Marker *x* crossing in volts. Valid only for Oscilloscope Inputs AC, DC and GND.

POINt *n*
Sets Marker *x* position to *n* graticules. Range of *n* is 1 to 100 with 100 graticules equal to Oscilloscope display width.

POINt?
Returns Marker *x* position in graticules.

STATe *b*
Enables Marker *x* if *b* is 1, disables Marker *x* if *b* is 0.

NOTE: Both Markers are displayed when one or both are active.

STATe?
Returns 1 if Marker *x* is active, 0 if Marker *x* is not active.

TIME?
Returns Marker *x* position in ms from left edge of Oscilloscope display.

Example:
```
SCOP:MARKER1:STAT 1     // Enables Marker 1.
SCOP:MARKER2:STAT 1     // Enables Marker 2.
SCOP:MARKER1:POIN 20    // Positions Marker 1 two major divisions
                        // from left edge of CRT.
SCOP:MARKER2:POIN 40    // Positions Marker 2 four major divisions
                        // from left edge of CRT.
SCOP:MARK:TRACK 1       // Enables Marker Tracking.
SCOP:MARKER1:POIN 30    // Positions Marker 1 three divisions from
                        // left edge of CRT (and Marker 2 five
                        // divisions from left edge of CRT).
SCOP:MARK:DELTA:AMPL?   // Queries amplitude difference of Trace
                        // Marker crossings in volts.
SCOP:MARK:DELTA:POIN?   // Queries position difference of markers
                        // in graticules.
SCOP:MARK:DELTA:TIME?   // Queries position difference of markers
                        // in ms.
SCOP:MARK:TRACK 0       // Disables Marker Tracking.
SCOP:MARKER2:AMPL?      // Queries the amplitude of the Trace at
                        // Marker 2 position.
SCOP:MARKER2:POIN?      // Queries Marker 2 position in graticules.
                        // 50 is returned.
SCOP:MARKER2:TIME?      // Queries Marker 2 position in ms.
SCOP:MARK:AOFF          // Disables both Markers.
```

SCOPe:

QTR
Selects a 1/4 size Oscilloscope display for the RF Generator, Receiver and Duplex Operation Screens.

RCL *n*
Recalls Oscilloscope Trace and environment (routings and settings) stored in memory location *n*. Range of n is 1 to 9.

NOTE: Recalled Trace reflects the stored Trace parameters. Changing current settings does not alter the recalled Trace.

Example:
```
SCOP:RCL 4          // Recalls Oscilloscope trace and environment
                    // stored in memory location 4.
  SCOP:SCAL 500     // Change current scale to 500 mV/div.
                    // Recalled Trace does not change position.
```

SCALe *n*
Sets Oscilloscope scale to *n* mV/div for AC, DC or GND Input. Select from:

1	2	5
10	20	50
100	200	500
1000	2000	5000
10000	20000	50000
100000		

Sets Oscilloscope scale to *n* kHz/div for Demod Audio Input with FM. Select from:

2	4	10
20		

Sets Oscilloscope scale to *n* mV/div for Func Gen or Ext Mod Input. Select from:

500	1000	2500

SCALe?
Returns Oscilloscope scale in mV/div if input is AC, DC, GND, SINAD/BER, Func Gen or Ext Mod. Returns Oscilloscope scale in kHz/div if input is Demod Audio. Returns Oscilloscope scale in W if input is RF Pwr Lvl.

SOURce EXTernal
Routes Oscilloscope Input from the SCOPE IN Connector.

SOURce INTernal
Disconnects SCOPE IN Connector and selects Internal Oscilloscope Input.

STATe *b*
Displays an Oscilloscope display in the RF Generator, Receiver and Duplex Operation Screens if *b* is 1. Oscilloscope display is not displayed if *b* is 0.

STORe *n*
Stores current Oscilloscope Trace and environment (routings and settings) in memory location *n*. Range of *n* is 1 to 9.

SCOPe:

SWEep *t*
Sets Oscilloscope sweep rate to *t* µs/div. Select from:

1	2	5
10	20	50
100	200	500
1000	2000	5000
10000	20000	50000
100000		

SWEep?
Returns Oscilloscope sweep rate in µs/div.

```
Example: SCOP:INT FUNC          // Selects AF Generator as the Oscilloscope
                                // Input.
         SCOP:LIV               // Selects Live for Oscilloscope Mode.
         SCOP:TRIG:AUTO         // Selects Auto for Oscilloscope Trigger
                                // Mode.
```

For the following **SCOPe:TRACE** commands, the Oscilloscope display is divided into 400 positions horizontally (0 signifying the left edge of the display, 399 signifying the right edge of the display) and 255 values vertically (0 signifying the bottom of the display, 255 signifying the top of the display):

TRACE:

DATA *n,offset,value,value,...,value*
Replaces points of a stored Trace *n* with specified *values* starting after *offset*. Optional *Offset* is given in number of positions from the left edge of the display with a default of 0. Range of *n* is 1 to 9. Multiple *values* are separated by commas. Only one *value* can be displayed for each horizontal position of the Trace. Intended for remote GPIB or RS-232 use only.

DATA? *n,offset,points*
Sends the specified number of data points of Trace *n* to the Host. Trace *n* can be the live Trace or a Trace stored in memory. The affected *points* start after the optional *offset* with an *offset* default of 0. Range of *offset* is 0 to 399. Range of *n* is 0 to 9 with 0 signifying the Live Trace. Range of the number of *points* is 1 to 400 with a default of 400. Intended for remote GPIB or RS-232 use only.

GET *name,n*
Assigns values of trace *n* (in graticules) to a declared array. Parameter *name* is the name of the declared array. Range of *n* is 0 to 9, 0 signifying the live trace and 1 to 9 signifying stored traces. If array is less than 400 values in length (stored trace length), the array values are assigned for the length of the array, leaving the rest of the stored trace intact.

GET? *n,offset*
Returns value of a point in Trace *n* located the *offset* number of positions from the left edge of the Display. Trace *n* can be the live Trace or a Trace stored in memory. Range of *n* is 0 to 9 with 0 signifying the Live Trace. Range of *offset* is 0 to 399.

SCOPe:

TRACE:

MAX? *n,offset,points*
Returns the maximum point of Trace *n* within the specified number of *points* starting after the given *offset.* Result is returned in x,y format with x being the number of positions from the left edge and y being the number of values from the bottom. Trace *n* can be the live Trace or a Trace stored in memory. Range of *n* is 0 to 9 with 0 signifying the Live Trace. Range of the optional *offset* is 0 to 399 with a default of 0. Range of optional number of *points* is 1 to 400 with a default of 400. Intended for remote GPIB or RS-232 use only.

MIN? *n,offset,points*
Returns the minimum point of Trace *n* within the specified number of *points* starting after the given *offset.* Result is returned in x,y format with x being the number of positions from the left edge and y being the number of values from the bottom. Trace *n* can be the Live Trace or a Trace stored in memory. Range of *n* is 0 to 9 with 0 signifying the Live Trace. Range of the optional *offset* is 0 to 399 with a default of 0. Range of optional number of *points* is 1 to 400 with a default of 400. Intended for remote GPIB or RS-232 use only.

PUT *name,n*
Assigns values of an array to trace *n* (in graticules). Parameter *name* is the array name. Range of *n* is 1 to 9, signifying stored traces. If array is less than 400 values in length (stored trace length), the array values are assigned for the length of the array, leaving the rest of the stored trace intact.

Example:
```
SCOP:TRACE:DATA 3,40,128      // Changes a single Trace value,
                              // located one division from the
                              // left edge, to the midway point
                              // (on the axis).
SCOP:TRACE:GET? 0,200         // Queries the value of the live
                              // Trace at the point 200 (center
                              // of oscilloscope display).
SCOP:TRACE:DATA? 3,200,200    // Returns the Trace values of the
                              // right half of the Trace stored
                              // in memory location 3.
SCOP:TRACE:MAX? 0,200         // Queries the maximum Trace value
                              // of the left half of the live
                              // Trace.
SCOP:TRACE:MIN? 0,40,80       // Queries the minimum Trace value
                              // of the live Trace for the second
                              // and third divisions.
```

SCOPe:

TRIGger:

AUTO
Sets Trigger to Auto Sweep.

IMMediate
Triggers Oscilloscope as soon as command is interpreted or sequenced in a macro.

NORM
Sets Trigger to Normal Sweep.

ONE
Sets Trigger to One Shot. Causes oscilloscope to sweep once when trigger is received and after setting arm function (**SCOPe:ARM** command).

SOURce *type*
Selects source to trigger on. Select EXTernal (for external trigger, AC or DC Input only), INTernal (for internal trigger, AC or DC Input only) or BUS (triggered by *TRG or IEEE-488.1 GET [Get Execute Trigger] command).

VERTical *n*
Sets Vertical Offset voltage to *n* graticules. Range of n is 0 to 255. Setting is not linear. For AC, DC or GND, middle setting is approximately 150 to 170. For other Inputs, middle setting is approximately 160 to 180.

4-9-2 REMOTE OSCILLOSCOPE EXAMPLES

The following example is a macro that centers a trace in the middle of the CRT display. If the trace centered has AC, DC or GND Input, GND Input should be selected before executing this macro. If the trace centered has other Input, RF Pwr Lvl Input should be selected before executing this macro. The Oscilloscope Operation Screen must be displayed for this macro to operate.

```
*DMC "Center",BEGIN          // Defines macro names Center.
VAR HOLD,CENTER_LINE,X       // Declares variables needed.
CENTER_LINE = 0              // Initializes Center_line variable.
FOR X = 0 TO 255             // Loops for every possible vertical trace
                             // position.
  SCOPE:VERT X               // Moves trace to vertical position x.
  DELAY 50                   // Pause allows SCOPE:VERT command to finish.
  HOLD = SCOP:TRACE:GET? 0,200          // Trace's actual vertical
                                        // position is stored in HOLD.
  IF (HOLD>=(128-3)) AND (HOLD<=(128+3)) // If HOLD is close to center
                                        // of CRT display,
    CENTER_LINE = X                     // store x value into
                                        // CENTER_LINE.
  ENDIF                      // End of IF statement.
NEXT X                       // End of FOR loop.
SCOPE:VERT CENTER_LINE       // Trace is moved to point previously found
                             // that centers trace.
END                          // End of macro Center.
```

The following example is a macro that finds the maximum of the live trace starting from a given offset. Range of the offset is 0 to 400. This macro returns the horizontal position of the maximum point. If there are multiple maximum points, the first one is returned. With the **RETURN** command included, this macro is executed as a function (ex. X = FINDMAX 0).

```
*DMC "FINDMAX",BEGIN         // Define macro named FINDMAX
VAR NEW,OLD=0,OLD_X=0        // Declare and initialize variables: NEW hold
                             // latest amplitude reading, OLD holds
                             // highest amplitude reading yet found and
                             // OLD_X holds position of OLD.
FOR X=$1 TO 400              // Loop from offset given to end of CRT.
  NEW=SCOPE:TRACE:GET? 0,X   // Read 1 point and store in NEW.
  IF NEW > OLD+3             // If current amplitude is greater than last
                             // max,
    OLD = NEW                // store new max into old max and,
    OLD_X = X                // store new max location into old max
                             // location.
  ENDIF                      // End of IF statement.
NEXT X                       // End of looping.
RETURN OLD_X                 // Return location of maximum value found.
END                          // End of FINDMAX macro.
```

The following command sequence compares a stored Trace (1 kHz sine wave) with a live Trace (2 kHz square wave). This example assumes the previous macro "Center" already resides in Test Set memory.

```
                              // AF Generator is setup to create first
                              // signal.

FGEN:OUTPUT:LEVEL 1           // Sets AF Generator Output Level to 1 V.
FGEN:GEN1:STATE 1             // Enables AF Generator 1.
FGEN:GEN1:FREQ 1000           // Sets AF Generator 1 frequency to 1000 Hz.
FGEN:GEN1:SHAPE:SIN           // Selects a sine wave for AF Generator 1
                              // Wave Form.

                              // Oscilloscope parameters are set and first
                              // Trace is stored.

SCREEN:SCOPE                  // Displays Oscilloscope Operation Screen.
SCOPE:SOURCE INTERNAL         // Sets Oscilloscope to be triggered
                              // internally.
SCOPE:TRIGGER:NORM            // Selects Normal as Oscilloscope Trigger
                              // Mode.
                              // Sets Trigger Level to the midpoint.
SCOPE:LIVE                    // Sets Oscilloscope Mode to live.
SCOPE:INTERNAL POWER          // Sets Oscilloscope Input to RF Pwr Lvl.
Center                        // Executes macro Center which centers trace
                              // on CRT (see example above).
SCOPE:INTERNAL FUNC           // Selects AF Generator as Oscilloscope
                              // Input.
SCOPE:SCALE 1000              // Sets Scale to 1 volt/div.
SCOPE:SWEEP 200               // Sets Sweep rate to 200 µV/div.
SCOPE:STORE 1                 // Stores Trace in memory location 1.

                              // Second signal is created.

FGEN:GEN1:FREQ 2000           // Changes AF Generator 1 frequency to 2 kHz.
FGEN:GEN1:SHAPE:SQU           // Changes AF Generator 1 Wave Shape to a
                              // square wave.

                              // The first and second signals are
                              // compared.

SCOPE:COMPARE 1               // Selects Compare mode for Oscilloscope.
                              // Trace stored in memory location 1 is
                              // displayed on the Oscilloscope display
                              // with the current live Trace.
```

The following command sequence displays the sum of a 1 kHz and 2.5 kHz sine wave on the Oscilloscope display and returns the period of the signals. This example assumes the macro FINDMAX (see second example) is already loaded into Test Set memory.

```
                                    // Setup AF Generator 1 and 2 to generate
                                    // the signal to be measured.
FGEN:PROPORTIONAL 1                 // Enable AF Generator Proportional Mode.
FGEN:OUTPUT:LEVEL 1                 // Set AF Generator Output Level to 1 V.
FGEN:GEN1:STATE 1                   // Enable AF Generator 1.
FGEN:GEN1:FREQ 1000                 // Set AF Generator 1 Frequency to 1000 Hz.
FGEN:GEN1:SHAPE:SIN                 // Select a sine wave for the AF
                                    // Generator 1 Wave Form.
FGEN:GEN1:LEVEL 100                 // Set AF Generator 1 Proportional Level
                                    // to 100%.
FGEN:GEN2:STATE 1                   // Enable AF Generator 2.
FGEN:GEN2:FREQ 2500                 // Set AF Generator 2 Frequency to 2500 Hz.
FGEN:GEN2:SHAPE:SIN                 // Select a sine wave for the AF
                                    // Generator 1 Wave Form.
FGEN:GEN2:LEVEL 100                 // Set AF Generator 2 Proportional Level
                                    // to 100%.


                                    // MEAS_PER macro measures the period of
                                    // the AF Generator signal.
*DMC "MEAS_PER",BEGIN               // Define macro named MEAS_PER.
VAR MAX1,MAX2                       // Declare variables: MAX1 holds location
                                    // of first maximum, MAX2 holds location
                                    // of second maximum.
SCREEN:SCOPE                        // Display the Oscilloscope Screen.
SCOPE:SOURCE INTERNAL               // Select Internal Oscilloscope Input.
SCOPE:INTERNAL FUNC                 // Select the AF Generator as the
                                    // Oscilloscope Input.
SCOPE:LIVE                          // Select Live Mode for the Oscilloscope.
SCOPE:SCALE 500                     // Set Oscilloscope Scale to 0.5 V/div.
SCOPE:SWEEP 500                     // Set Oscilloscope Sweep Rate to
                                    // 500 µs/div.
SCOPE:TRIGGER:ONE                   // Select One Shot Trigger Mode.  Hold
                                    // trace still.
DELAY 600                           // Allow time to set for arming function.
SCOPE:ARM                           // Display and hold trace when triggered.
MAX1 = FINDMAX 0                    // MAX! is set to first maximum using
                                    // macro of second example.
MAX2 = FINDMAX (MAX1+5)             // MAX2 is set to second maximum using
                                    // macro of second example.
SCOPE:MARKER1:STATE 1               // Enables Marker 1.
SCOPE:MARKER1:POINT FLOOR(MAX1/4)   // Position Marker 1 on the first maximum.
SCOPE:MARKER2:STATE 1               // Enables Marker 2.
SCOPE:MARKER2:POINT FLOOR(MAX2/4)   // Positions Marker 2 on the second
                                    // maximum.
T=SCOPE:MARKER:DELTA:TIME?          // Query the difference in Marker
                                    // positions in ms.
PPRINT "PERIOD IS ",1000/T          // Figure and print period to Host.
END                                 // End macro MEAS_PER.
```

4-10 SPECTRUM ANALYZER

4-10-1 SPECTRUM ANALYZER COMMANDS

ANLZ:

AVErage *n*
Selects Average Mode for Analyzer using *n* samples. Range of *n* is 1 to 100. Default is 100.

CHANnel *n*
Sets RF Frequency to cellular channel *n* (1 to 1023) in the format selected using the **ANLZ:CHANnel:FORMat** command. The Spectrum Analyzer Operation Screen displays selected cellular channel when Analyzer is in Channel Mode (**ANLZ:MODE CHAN** command).

CHANnel:

FORMat:

AMPS:

FORward
Selects AMPS Forward channels.

REVerse
Selects AMPS Reverse channels.

ETACS:

FORward
Selects ETACS Forward channels.

REVerse
Selects ETACS Reverse channels.

NAMPS:

FORward
Selects NAMPS Forward channels.

REVerse
Selects NAMPS Reverse channels.

NT400:

FORward
Selects NT400 Forward channels.

REVerse
Selects NT400 Reverse channels.

FORMat?
Returns channel format setting (AMPS:FORWARD, AMPS:REVERSE, ETACS:FORWARD, ETACS:REVERSE, NAMPS:FORWARD, NAMPS:REVERSE, NT400:FORWARD or NT400:REVERSE).

ANLZ:

COMPare *n*
Selects Compare Mode for Analyzer. Trace stored at *n* memory location is compared with Live Trace.

```
Example: ANLZ:COMP 1          // Selects Compare mode for Analyzer.
                              // Trace stored in memory location 1 is
                              // displayed on the Analyzer display
                              // with the current live Trace.
```

FIND:

FREQuency?
Returns frequency of first signal with amplitude larger than Find reference level. Returns 0 if no signal is found.

REFerence *n*
Sets Find reference level to *n* dB.

REFerence?
Returns Find reference level in dB.

```
Example: ANLZ:FIND:REF -65    // Sets Find Reference Level to -65 dBm.
         ANLZ:FIND:FREQ?      // Returns the lowest frequency (in kHz)
                              // containing a signal greater than -65 dBm.
         ANLZ:FIND:REF?       // Queries the current Find Reference Level
                              // (in dBm).  -65 is returned.
```

FREQuency *f*
Sets Analyzer Frequency to *f* kHz. Range for *f* is 250.0 to 999999.9.

FREQuency?
Returns Analyzer Frequency in kHz.

```
Example: ANLZ:FREQ 135 MHZ    // Set Analyzer Frequency to 135 MHz.
         ANLZ:FREQ?           // Query the Analyzer Frequency.
                              // 135000 is returned (135000 kHz).
```

FULL
Selects a full size Analyzer display for the RF Generator, Receiver and Duplex Operation Screens.

INPut:

ANTenna
Selects ANTENNA IN Connector for the Analyzer Input.

ATTenuation *n*
Sets Analyzer Input Attenuation to *n* dB. Select 0, 20 or 40.

ATTenuation?
Returns Analyzer Input attenuation in dB.

TR
Selects T/R Connector for the Analyzer Input.

ANLZ:

INPut?
Returns Analyzer Input setting.

```
Example: ANLZ:INP:TR            // Selects T/R Connector for Analyzer Input.
         ANLZ:INP:ATT 20        // Sets Analyzer Input Attenuation to 20 dB.
         ANLZ:INP?              // Queries Analyzer Input Connector.  TR is
                                // returned.
         ANLZ:INP:ATT?          // Queries Analyzer Input Attenuation.
                                // 20 is returned.
```

LIVe
Selects Live Trace mode for the Spectrum Analyzer.

MARKer:

AOFF
Disables both Markers.

DELTA:

AMPLitude?
Returns amplitude difference between the Trace Marker 1 and Trace Marker 2 crossings in dB.

FREQuency?
Returns the difference between the two Marker positions in MHz.

POINt?
Returns the difference between the Marker positions in graticules with 100 graticules equal to the Analyzer display width.

TRACK *b*
Enables Marker Tracking Feature if *b* is 1, disables Marker Tracking Feature if *b* is 0. Tracking feature keeps Markers a constant distance apart.

Commands for Marker 1 and Marker 2 are identical and are listed as **ANLZ:MARKER***x* commands. Parameter *x* is specified as 1 or 2 when the command is entered.

MARKER*x***:**

AMPLitude?
Returns amplitude of the Trace at the Marker *x* crossing. Range and units depend on the current scale settings (**ANLZ:SCALe** commands).

FREQuency?
Returns Marker *x* position in kHz (250 to 999999.9).

POINt *n*
Sets Marker *x* position to *n* graticules. Range of *n* is 1 to 100 with 100 graticules equal to the Analyzer display width.

POINt?
Returns Marker *x* position in graticules.

ANLZ:

MARKER:

STATe *b*
Enables Marker x if *b* is 1, disables Marker x if *b* is 0.

Example:
```
ANLZ:MARKER1:STAT 1    // Enables Marker 1.
ANLZ:MARKER2:STAT 2    // Enables Marker 2.
ANLZ:MARKER1:POIN 10   // Positions Marker 1 one divisions from the
                       // left edge of the CRT.
ANLZ:MARKER2:POIN 30   // Positions Marker 2 three divisions from
                       // the left edge of the CRT.
ANLZ:MARK:TRACK        // Enables Marker Tracking.
ANLZ:MARKER1:POIN 60   // Positions Marker 1 six divisions from
                       // the left edge of the CRT (and Marker 2
                       // eight divisions from the left edge of
                       // the CRT).
ANLZ:MARK:DELTA:AMPL?  // Queries amplitude difference of Trace
                       // Marker crossings in volts.
ANLZ:MARK:DELTA:POIN?  // Queries position difference of markers in
                       // graticules.
ANLZ:MARK:DELTA:FREQ?  // Queries position difference of markers
                       // in kHz.
ANLZ:MARK:TRACK 1      // Disables Marker Tracking.
ANLZ:MARKER2:AMPL?     // Queries the amplitude of the Trace at the
                       // Marker 2 position.
ANLZ:MARKER2:POIN?     // Queries Marker 2 position in graticules.
                       // 80 is returned.
ANLZ:MARKER2:FREQ?     // Queries Marker 2 position in kHz.
ANLZ:MARK:AOFF         // Disables both Markers.
```

MODE *type*
Selects Analyzer RF Mode (DIRect [Direct Mode] or CHANnel [Channel Mode]). Channel Mode displays cellular channel frequency according to **ANLZ:CHANnel** commands.

NORMalize
Normalizes the Analyzer Trace to match the RF Generator Output.

PEAK
Selects Peak Hold Feature for the Analyzer.

QTR
Selects a 1/4 size Analyzer display for the RF Generator, Receiver and Duplex Operation Screens.

RCL *n*
Recalls Analyzer Trace and parameters stored in memory location *n*. Range of *n* is 1 to 9.

RLEVel?
Returns offset value, in dB, used in 2 dB/div scale.

ANLZ:

SCALe *n*
Sets Analyzer Units/Division Factor to *n* dB. Select 2 or 10.

SCALe:

UNIT:*type*
Sets Analyzer Scale Units to *type*. For T/R Analyzer Input, select DBM (dBm) or DBW (dBW). For ANTENNA Analyzer Input, select:

DBM (dBm)	DBMV (dBmV)	DBUV (dBµV)
DBV (dBV)	DBUW (dBµW)	

If DBW is selected, the T/R Connector is selected for the Analyzer Input. If DBV, DBMV, DBUV or DBUW is selected, the ANTENNA IN Connector is selected for the Analyzer Input.

UNIT?
Returns Analyzer Scale Units.

SCALe?
Returns Analyzer Scale in dB.

SCAN *n*
Sets Analyzer Scan Width to *n* kHz. Select 0 for zero scan or one of the following:

1	2	5
10	20	50
100	200	500
1000	2000	5000
10000	20000	50000
100000		

SCAN?
Returns Analyzer Scan Width in kHz.

STATE *b*
Displays Analyzer display in the RF Generator, Receiver and Duplex Operation Screens if *b* is 1. Analyzer display is not displayed if *b* is 0.

STORe *n*
Stores the current Analyzer Trace and environment (routings and settings) in memory location *n*. Range of *n* is 1 to 9.

TOP?
Returns the top of screen value in current units. Spectrum Analyzer Operation Screen must be displayed.

ANLZ:

```
Example:  ANLZ:SCAL 2            // Selects 2 dB/div for Analyzer
                                 // Units/Division Factor.
          ANLZ:SCAN 100          // Sets Analyzer Scan Width to 100 kHz.
          ANLZ:SCAL:UNIT:DBM     // Selects dBm for Analyzer Scale Units.
          ANLZ:TOP?              // Queries the amplitude value for the top
                                 // of the Analyzer display (in the current
                                 // units).
          ANLZ:SCAN?             // Queries Analyzer Scan Width in kHz.  100
                                 // is returned.
          ANLZ:SCAL:UNIT?        // Queries Analyzer Scale Units.
```

For the following **ANLZ:TRACE** commands, the Analyzer display is divided into 400 positions horizontally (0 signifying the left edge of the display, 399 signifying the right edge of the display) and 255 values vertically (0 signifying the bottom of the display, 255 signifying the top of the display):

TRACE:

DATA *n,offset,value,value,...,value*
Replaces points of a stored Trace *n* with specified *values* starting after *offset*. Optional *Offset* is given in number of positions from the left edge of the display with a default of 0. Range of *n* is 1 to 9. Multiple *values* are separated by commas. Only one *value* can be displayed for each horizontal position of the Trace. Intended for remote GPIB or RS-232 use only.

DATA? *n,offset,points*
Sends the specified number of data points of Trace *n* to the Host. Trace *n* can be the live Trace or a Trace stored in memory. The affected *points* start after the optional *offset* with an *offset* default of 0. Range of *offset* is 0 to 399. Range of *n* is 0 to 9 with 0 signifying the Live Trace. Range of the number of *points* is 1 to 400 with a default of 400. Intended for remote GPIB or RS-232 use only.

GET *name,n*
Assigns values of trace *n* (in graticules) to a declared array. Parameter *name* is the name of the declared array. Range of *n* is 0 to 9, 0 signifying the live trace and 1 to 9 signifying stored traces. If array is less than 400 values in length (stored trace length), the array values are assigned for the length of the array, leaving the rest of the stored trace intact.

GET? *n,offset*
Returns value of a point in Trace *n* located the *offset* number of positions from the left edge of the Display. Trace *n* can be the live Trace or a Trace stored in memory. Range of *n* is 0 to 9 with 0 signifying the Live Trace. Range of the *offset* is 0 to 399.

MAX? *n,offset,points*
Returns the maximum point of Trace *n* within the specified number of *points* starting after the given *offset*. Result is returned in x,y format with x being the number of positions from the left edge and y being the number of values from the bottom. Trace *n* can be the live Trace or a Trace stored in memory. Range of *n* is 0 to 9 with 0 signifying the Live Trace. Range of the optional *offset* is 0 to 399 with a default of 0. Range of optional number of *points* is 1 to 400 with a default of 400. Intended for remote GPIB or RS-232 use only.

ANLZ:

TRACE:

MIN? *n,offset,points*
Returns the minimum point of Trace *n* within the specified number of *points* starting after the given *offset*. Result is returned in x,y format with x being the number of positions from the left edge and y being the number of values from the bottom. Trace *n* can be the live Trace or a Trace stored in memory. Range of *n* is 0 to 9 with 0 signifying the Live Trace. Range of the optional *offset* is 0 to 399 with a default of 0. Range of optional number of *points* is 1 to 400 with a default of 400 Intended for remote GPIB and RS-232 use only.

PUT *name,n*
Assigns values of an array to trace *n* (in graticules). Parameter *name* is the array name. Range of *n* is 1 to 9, signifying stored traces. If array is less than 400 values in length (stored trace length), the array values are assigned for the length of the array, leaving the rest of the stored trace intact.

Example:
```
ANLZ:TRACE:DATA 2,80,128        // Changes a single Trace value,
                                // located 2 divisions from the left
                                // edge, to the midway point (on the
                                // axis).
ANLZ:TRACE:GET? 0,240           // Queries the value of the live
                                // Trace at the point 240 (six
                                // divisions from the left edge of
                                // the display).
ANLZ:TRACE:DATA? 3,200,200      // Returns the Trace values of the
                                // right half of the Trace stored
                                // in memory location 3 (200 values).
ANLZ:TRACE:MAX? 0,200           // Queries the maximum Trace value of
                                // the left half of the live Trace.
ANLZ:TRACE:MIN? 0,40,80         // Queries the minimum Trace value
                                // of the live Trace for the second
                                // and third divisions.
```

TRACK:

BWIDth *f*
Sets Tracking Generator Bandwidth to *f* kHz. Select 0.3, 3, 30, 300 or 3000.

BWIDth?
Returns Tracking Generator Bandwidth in kHz.

LEVel *n*
Sets Tracking Generator Level to *n* dB. Range of *n* is 0.0 to -127.0.

LEVel?
Returns Tracking Generator Level in dB.

ANLZ:

TRACK:

OUTput:

TR
Selects T/R Connector as Tracking Generator Output Connector.

DUPlex
Selects DUPLEX OUT Connector as Tracking Generator Output Connector.

OUTput?
Returns current Tracking Generator Output Connector.

RESolution:

HIGH
Selects high for Tracking Generator Resolution.

LOW
Selects low for Tracking Generator Resolution.

MED
Selects medium for Tracking Generator Resolution.

RESolution?
Returns HIGH if Tracking Resolution is high, LOW if Tracking Resolution is low or MED if Tracking Resolution is medium.

STATe *b*
Enables Tracking Generator if *b* is 1, disables Tracking Generator if *b* is 0.

STATe?
Returns 1 if Tracking Generator is active; 0 if inactive.

Example:
```
ANLZ:TRACK:LEV -10      // Sets Analyzer Tracking Generator Level
                        // to -10 dBm.
ANLZ:TRACK:RES:HIGH     // Sets Analyzer Tracking Generator
                        // Resolution to high.
ANLZ:TRACK:BWID 30      // Sets Analyzer Bandwidth to 30 kHz.
ANLZ:TRACK:STAT 1       // Enables the Analyzer Tracking Generator.
ANLZ:TRACK:LEV?         // Queries the Analyzer Tracking Generator
                        // Level.  -10 is returned.
ANLZ:TRACK:RES?         // Queries the Analyzer Tracking Generator
                        // Resolution.  HIGH is returned.
ANLZ:TRACK:BWID?        // Queries the Analyzer Bandwidth in kHz.
                        // 30 is returned.
ANLZ:TRACK:STAT?        // Queries the current status of the
                        // Analyzer Tracking Generator.  1 is
                        // returned.
```

4-10-2 REMOTE SPECTRUM ANALYZER EXAMPLE

The following command sequence measures a 96 MHz signal received at the ANTENNA IN Connector and returns the amplitude in dB:

```
SCREEN:ANLZ                    // Displays the Analyzer Operation Screen.
ANLZ:SCAN 20                   // Sets the Analyzer Scan Width to 20 kHz.
ANLZ:FREQ 96000                // Sets Analyzer RF Frequency to 96 MHz.
ANLZ:LIVE                      // Selects the Live Analyzer Mode.
ANLZ:SCALE 10                  // Selects the 10 Units/Division Factor.
ANLZ:SCALE:UNIT:DBM            // Selects dBm for the Analyzer Scale Units.
ANLZ:INPUT:ANTENNA             // Selects the ANTENNA IN Connector for the
                               // Analyzer Input.
ANLZ:INPUT:ATTENUATION 0       // Sets Analyzer Input Attenuation to 0 dB.
ANLZ:MARKER1:STATE 1           // Enables Marker 1.
ANLZ:MARKER1:POINT 50          // Positions Marker 1 at the Analyzer RF
                               // Frequency.
ANLZ:MARKER1:AMPLITUDE?        // Queries the amplitude of the Trace at the
                               // Marker 1 position.
```

4-11 METER COMMANDS

4-11-1 AF METER COMMANDS

M_AF:

ALARM *b*
Enables Alarm if *b* is 1, disables if *b* is 0. Enabled Alarm sounds when Upper or Lower Limit is surpassed.

FILTer:

HPASs:

FREQuency *f*
Sets High-Pass cutoff frequency to *f* kHz. Range of *f* is 0.5 to 20.0.

FREQuency?
Returns High-Pass cutoff frequency in kHz.

STATe *b*
Enables High-Pass Filter if *b* is 1, disables if *b* is 0.

STATe?
Returns 1 if High-Pass Filter is enabled, 0 if disabled.

LPASs:

FREQuency *f*
Set Low-Pass cutoff frequency to *f* kHz. Range of *f* is 0.1 to 30.0.

FREQuency?
Returns Low-Pass cutoff frequency in kHz.

STATe *b*
Enables Low-Pass Filter if *b* is 1, disables if *b* is 0.

STATe?
Returns 1 if Low-Pass Filter is enabled, 0 if disabled.

Example:

```
M_AF:FILT:LPAS:FREQ 5      // Sets Low-Pass cutoff frequency to 5 kHz.
M_AF:FILT:LPAS:STAT 1      // Enables Low-Pass Filter.
M_AF:FILT:LPAS:FREQ?       // Queries the Low-Pass cutoff frequency in
                           // kHz.  5 is returned.
M_AF:FILT:LPAS:STAT?       // Queries the Low-Pass Filter status.
                           // 1 is returned.
```

INPut:*type*
Selects the Audio Frequency Meter Input. Select from the following for *type*:

XAUDIO (Ext Mod)	DEMOD (Demod Audio)	FGEN (Func Gen Out)
SINAD (SINAD/BER)	POWer (RF Power)	

M_AF:

LL:

LEVel *f*
Sets Lower Limit to *f* kHz. Range of *f* is 0.0000 to 0.2000 for an upper range of 0.2. Range of *f* is 0.000 to 200.000 for upper ranges of 2, 20 and 200.

STATe *b*
Enables Audio Frequency Meter Lower Limit if *b* is 1, disables if *b* is 0. When enabled, exceeding the Lower Limit sets bit 3 of the Instrument Status Register to one (activating bit 13 of the Questionable Status Register and bit 3 of the Status Byte) (see 3-16). When the Lower Limit and Alarm (**M_AF:ALARM 1**) are enabled, exceeding the Lower Limit activates an audio alarm.

PEAK?
Returns Audio Frequency Meter Peak reading in Hz (0.0 to 200000.0).

PH *b*
Enables Peak Hold Feature when *b* is 1, disables Peak Hold Feature when *b* is 0.

RANGe:

AUTO
Sets frequency range to Autorange.

UPPer *f*
Sets frequency range to *f* kHz. Select 0.2, 2, 20 or 200.

Example:
```
M_AF:INP:SINAD          // Selects SINAD/BER IN Connector for AF
                        // Meter Input.
M_AF:RANG:UPP 20        // Sets AF Meter Range to 20 kHz.
M_AF:PH 1               // Enables AF Meter Peak Hold.
M_AF:LL:LEV 55.5        // Sets a Lower Limit of 55.5 kHz.
M_AF:LL:STAT 1          // Enables Lower Limit.
M_AF:ALARM 1            // Enables Alarm.
```

RCL *n*
Recalls Audio Frequency Meter environment (routings and settings) stored in memory location *n*. Range of *n* is 1 to 9.

RESolution *n*
Sets Audio Frequency Meter Resolution (Gate Time) to *n* Hz. Select 0.1 (Gate Time of 10 s) or 1 (Gate Time of 1 s).

STORe *n*
Stores current Audio Frequency Meter environment (routings and settings) in memory location *n*. Range of *n* is 1 to 9.

M_AF:

UL:

LEVel *f*
Sets Upper Limit to *f* kHz. Range of *f* is 0.0000 to 0.2000 for an upper range of 0.2. Range of *f* is 0.000 to 200.000 for upper ranges of 2, 20 and 200.

STATe *b*
Enables Audio Frequency Meter Upper Limit if *b* is 1, disables if *b* is 0. When enabled, exceeding the Upper Limit sets bit 2 of the Instrument Status Register to one (activating bit 13 of the Questionable Status Register and bit 3 of the Status Byte) (see 3-16). When the Upper Limit and Alarm (**M_AF:ALARM 1**) are enabled, exceeding the Upper Limit activates an audio alarm.

M_AF?
Returns an Audio Frequency Meter reading in Hz (0.0 to 200000.0). See also **MEASure:AUDio?** query (4-15).

NOTE: Reading is invalid if squelch is not broken for some internal routings.

Example:
```
SCREEN:AF              // Displays AF Meter Operation Screen.
M_AF:RANG:AUTO         // Sets AF Meter Range to Autorange.
M_AF:UL:LEV 100        // Sets AF Meter Upper Limit to 100 kHz.
M_AF:UL:STAT 1         // Enables Upper Limit.
M_AF:LL:LEV 25         // Sets AF Meter Lower Limit to 25 kHz.
M_AF:LL:STAT 1         // Enables Lower Limit.
M_AF:INP:XAUDIO        // Selects EXT MOD IN Connector for AF Meter
                       // Input.
M_AF:FILT:LPAS:FREQ 10 // Sets Low-Pass Filter cutoff frequency to
                       // 10 kHz.
M_AF:FILT:LPAS:STAT 1  // Enables Low-Pass Filter.
M_AF:ALARM 1           // Enables Alarm.
M_AF:PH 1              // Enables Peak Hold feature.
M_AF:PEAK?             // Queries AF Meter Peak reading.
M_AF?                  // Queries AF Meter reading.
```

4-11-2 FREQUENCY ERROR METER COMMANDS

M_RF:

ALARM *b*
Enables Alarm if *b* is 1, disables if *b* is 0. Enabled Alarm sounds when Upper or Lower Limit is surpassed.

LL:

LEVel *f*
Sets Lower Limit to *f* kHz. Range of *f* is 0.0000 to 0.1000 for an upper range of 0.1. Range of *f* is 0.000 to 100.000 for upper ranges of 1, 10 and 100.

STATe *b*
Enables Lower Limit if *b* is 1, disables if *b* is 0. When enabled, exceeding the Lower Limit sets bit 5 of the Instrument Status Register to one (activating bit 13 of the Questionable Status Register and bit 3 of the Status Byte) (see 3-16). When the Lower Limit and Alarm (**M_RF:ALARM 1**) are enabled, exceeding the Lower Limit activates an audio alarm.

PEAK?
Returns Frequency Error Meter Peak reading in Hz (0.0 to 100000.0).

PH *b*
Enables Peak Hold Feature if *b* is 1, disables Peak Hold Feature if *b* is 0.

RANGe:

AUTO
Sets Frequency Error Meter range to Autorange.

UPPer *f*
Sets Frequency Error Meter range to *f* kHz. Select 0.1, 1, 10 or 100.

Example:
```
M_RF:RANG:UPP 10        // Sets Frequency Error Meter Range to
                        // 10 kHz.
M_RF:PH 1               // Enables Frequency Error Meter Peak Hold.
M_RF:LL:LEV 20.5        // Sets a Lower Limit of 20.5 kHz.
M_RF:LL:STAT 1          // Enables Lower Limit.
M_RF:ALARM 1            // Enables Alarm.
```

RCL *n*
Recalls Frequency Error Meter environment (routings and settings) stored in memory location *n*. Range of *n* is 1 to 9.

RESolution *f*
Sets Frequency Error Resolution (Gate Time) to *f* Hz. Select 1 (Gate Time of 1 sec) or 10 (Gate Time of 0.1 sec).

STORe *n*
Stores current Frequency Error Meter environment (routings and settings) in memory location *n*. Range of *n* is 1 to 9.

M_RF:

UL:

LEVel *f*
Sets Upper Limit to *f* kHz. Range of *f* is 0.0000 to 0.1000 for an upper range of 0.1. Range of *f* is 0.000 to 100.000 for upper ranges of 1, 10 and 100.

STATe *b*
Enables Upper Limit if *b* is 1, disables if *b* is 0. When enabled, exceeding the Upper Limit sets bit 4 of the Instrument Status Register to one (activating bit 13 of the Questionable Status Register and bit 3 of the Status Byte) (see 3-16). When the Upper Limit and Alarm (**M_RF:ALARM 1**) are enabled, exceeding the Upper Limit activates an audio alarm.

M_RF?
Returns Frequency Error Meter reading in Hz (-100000.0 to +100000.0). Actual signal frequency can be calculated by converting the Receiver Frequency setting (**RECeive:FREQuency?** query) from kHz to Hz and adding the Frequency Error Meter reading. For an actual measured frequency reading, see **MEASure:FREQuency?** query (4-15).

NOTE: Reading is invalid if squelch is not broken for some internal routings.

Example:
```
SCREEN:FREQ          // Displays Frequency Error Meter Operation
                     // Screen.
M_RF:RANG:AUTO       // Sets Frequency Error Meter Range to Autorange.
M_RF:UL:LEV 100      // Sets Frequency Error Meter Upper Limit to
                     // 100 kHz.
M_RF:UL:STAT 1       // Enables Upper Limit.
M_RF:LL:LEV 25       // Sets Frequency Error Meter Lower Limit to
                     // 25 kHz.
M_RF:LL:STAT 1       // Enables Lower Limit.
M_RF:ALARM 1         // Enables Alarm.
M_RF:PH 1            // Enables Peak Hold feature.
M_RF:PEAK?           // Queries Frequency Error Meter Peak reading.
M_RF?                // Queries Frequency Error Meter reading.
```

4-11-3 POWER METER COMMANDS

M_PWR:

ALARM *b*
Enables Power Meter Alarm if *b* is 1, disables if *b* is 0. Enabled Power Meter Alarm sounds when Upper or Lower Limit is surpassed.

EXT:

STATe *b*
Enables External Loss/Gain Offset if *b* is 1, disables if *b* is 0. External Loss/Gain Offset compensates Power Meter readings for external gains or losses.

STATe?
Returns External Loss/Gain Offset state setting.

OFFSet *n*
Sets External Loss/Gain Offset value to *n* dBm. Range of *n* is -99.9 to 99.9. Positive values lower Power Meter readings and compensate for external gains. Negative values raise Power Meter readings and compensate for external losses.

OFFSet?
Returns External Loss/Gain Offset value in dBm.

LL:

LEVel *n*
Sets Power Meter Lower Limit to *n* W. Range of *n* is 0.0000 to 0.5000 for ranges: 0.02, 0.05, 0.1, 0.2 and 0.5. Range of *n* is 0.00 to 200.00 otherwise.

STATe *b*
Enables Power Meter Lower Limit if *b* is 1, disables if *b* is 0. When enabled, exceeding the Lower Limit sets bit 7 of the Instrument Status Register to one (activating bit 13 of the Questionable Status Register and bit 3 of the Status Byte) (see 3-16). When the Lower Limit and Alarm (**M_PWR:ALARM 1**) are enabled, exceeding the Lower Limit activates an audio alarm.

PEAK?
Returns Power Meter Peak reading in mW (0.0 to 200000.0).

PH *b*
Enables Power Meter Peak Hold Feature if *b* is 1, disables if *b* is 0.

RANGe:

AUTO
Sets Power Meter range to Autorange.

UPPer *n*
Sets Power Meter range to *n* W. Select from:

0.02	0.05	0.1
0.2	0.5	1
2	5	10
20	50	100
200		

M_PWR:

RCL *n*
Recalls Power Meter environment (routings and settings) stored in memory location *n*. Range of *n* is 1 to 9.

```
Example: M_PWR:RANG:UPP 2        // Sets Power Meter Range to 2 W.
         M_PWR:PH 1              // Enables Power Meter Peak Hold.
         M_PWR:LL:LEV 0.75       // Sets a Lower Limit of 0.75 W.
         M_PWR:LL:STAT 1         // Enables Lower Limit.
         M_PWR:ALARM 1           // Enables Alarm.
```

STORe *n*
Stores current Power Meter environment (routings and settings) in memory location *n*. Range of *n* is 1 to 9.

TYPE:

CW
Selects Average Power Measurement Type for Power Meter.

PEAK
Selects Peak Power Measurement Type for Power Meter.

RMS
Selects RMS Power Measurement Type for Power Meter.

UL:

LEVel *n*
Sets Power Meter Upper Limit to *n* W. Range of *n* is 0.0000 to 0.5000 for upper ranges: 0.02, 0.05, 0.1, 0.2 and 0.5. Range of *n* is 0.00 to 200.00 otherwise.

STATe *b*
Enables Power Meter Upper Limit if *b* is 1, disables if *b* is 0. When enabled, exceeding the Upper Limit sets bit 6 of the Instrument Status Register to one (activating bit 13 of the Questionable Status Register and bit 3 of the Status Byte) (see 3-16). When the Upper Limit and Alarm (**M_PWR:ALARM 1**) are enabled, exceeding the Upper Limit activates an audio alarm.

M_PWR?
Returns a Power Meter reading in mW (0.0 to 200000.0). See also **MEASure:POWer?** query (4-15).

```
Example: SCREEN:POW              // Displays Power Meter Operation Screen.
         M_PWR:RANG:AUTO         // Sets Power Meter Range to Autorange.
         M_PWR:UL:LEV 3          // Sets Power Meter Upper Limit to 3 W.
         M_PWR:UL:STAT 1         // Enables Upper Limit.
         M_PWR:LL:LEV .5         // Sets Power Meter Lower Limit to 0.5 W.
         M_PWR:LL:STAT 1         // Enables Lower Limit.
         M_PWR:ALARM 1           // Enables Alarm.
         M_PWR:PH 1              // Enables Peak Hold feature.
         M_PWR:PEAK?             // Queries Power Meter Peak reading.
         M_PWR?                  // Queries Power Meter reading.
```

4-11-4 DEVIATION METER (Peak) COMMANDS

M_DEV:

ALARM *b*
Enables Deviation Meter Alarm if *b* is 1, disables if *b* is 0. Enabled Alarm sounds when Upper or Lower Limit is surpassed.

AVErage *b*
Enables Deviation Meter Averaging if *b* is 1, disables if *b* is 0.

LL:

LEVel *f*
Sets the Deviation Meter Lower Limit to *f* kHz. *f* can vary from 0.00 to 20.00 for range values of 2, 5, 10 or 20 kHz. f varies from 0 to 100 otherwise.

STATe *b*
Enables Deviation Meter Lower Limit if *b* is 1, disables if *b* is 0. When enabled, exceeding the Lower Limit sets bit 9 of the Instrument Status Register to one (activating bit 13 of the Questionable Status Register and bit 3 of the Status Byte) (see 3-16). When the Lower Limit and Alarm (**M_DEV:ALARM 1**) are enabled, exceeding the Lower Limit activates an audio alarm.

MODE:

BOTH
Selects Both Mode for Deviation Meter, reading positive and negative deviation. Use **M_DEV:POS?** and **M_DEV:NEG?** to query meter readings.

NEGative
Selects Negative Mode for Deviation Meter, reading negative deviation. Use **M_DEV:NEG?** to query meter readings.

NORMalize
Selects Normalized Mode for Deviation Meter, reading (positive + negative)/2 deviation. Use **M_DEV:POS?** to query meter readings.

POSitive
Selects Positive Mode for Deviation Meter, reading positive deviation. Use **M_DEV:POS?** to query meter readings.

NOTE: Deviation Meter Screen must be updated for a Deviation Meter Mode change to take effect.

Example:
```
M_DEV:LL:LEV 10        // Sets a Lower Limit of 10 kHz.
M_DEV:LL:STAT 1        // Enables Lower Limit.
M_DEV:ALARM 1          // Enables Alarm.
M_DEV:MODE:POS         // Selects the Positive Mode of the
                       // Deviation (Peak) Meter.
SCREEN:DEV             // Updates the Deviation (Peak) Meter
                       // Operation Screen.
```

NEG?
Returns a negative Deviation Meter reading as an absolute value in kHz (0.00 to 100.00).

M_DEV:

PEAK:

NEG?
Returns a negative Deviation Meter Peak reading as an absolute value in kHz (0.00 to 100.00).

POS?
Returns a positive Deviation Meter Peak reading as an absolute value in kHz (0.00 to 100.00).

PH *b*
Enables the Deviation Meter Peak Hold Feature if *b* is 1, disables if *b* is 0.

POS?
Returns a positive Deviation Meter reading as an absolute value in kHz (0.00 to 100.00).

RANGe:

AUTO
Sets Deviation Meter range to Autorange.

UPPer *f*
Sets Deviation Meter range to *f* kHz. Settings for *f* are: 2, 5, 10, 20, 50 or 100.

Example:
```
M_DEV:MODE:BOTH        // Selects Both Mode.
SCREEN:DEV             // Updates the Deviation (Peak) Meter
                       // Operation Screen.
M_DEV:RANG:UPP 10      // Sets the Range to 10 kHz.
M_DEV:PH 1             // Enables Peak Hold feature.
```

RCL *n*
Recalls Deviation Meter environment (routings and settings) stored in memory location *n*. Range of *n* is 1 to 9.

STORe *n*
Stores current Deviation Meter environment (routings and settings) in memory location *n*. Range of *n* is 1 to 9.

M_DEV:

UL:

LEVel *f*
Sets Deviation Meter Upper Limit to *f* kHz. *f* can vary from 0.00 to 20.00 for range values of 2, 5, 10 or 20 kHz. *f* varies from 0 to 100 otherwise.

STATe *b*
Enables Deviation Meter Upper Limit if *b* is 1, disables if *b* is 0. When enabled, exceeding the Upper Limit sets bit 8 of the Instrument Status Register to one (activating bit 13 of the Questionable Status Register and bit 3 of the Status Byte) (see 3-16). When the Upper Limit and Alarm (**M_DEV:ALARM 1**) are enabled, exceeding the Upper Limit activates an audio alarm.

Example:
```
SCREEN:DEV              // Displays Deviation (Peak) Meter
                        // Operation Screen.
M_DEV:RANG:AUTO         // Sets Range to Autorange.
M_DEV:UL:LEV 30         // Sets Upper Limit to 30 kHz.
M_DEV:UL:STAT 1         // Enables Upper Limit.
M_DEV:LL:LEV 5          // Sets Lower Limit to 5 kHz.
M_DEV:LL:STAT 1         // Enables Lower Limit.
M_DEV:ALARM 1           // Enables Alarm.
M_DEV:PH 1              // Enables Peak Hold feature.
M_DEV:AVE 1             // Enables Deviation (Peak) Meter averaging.
M_DEV:MODE:NEG          // Selects the Negative Mode.
SCREEN:DEV              // Updates the Deviation (Peak) Meter
                        // Operation Screen.
M_DEV:NEG?              // Queries Negative Deviation (Peak) Meter
                        // reading.
```

4-11-5 MODULATION METER COMMANDS

The Modulation Meter measures AM.

M_MOD:

ALARM *b*
Enables Modulation Meter Alarm if *b* is 1, disables if *b* is 0. Enabled Alarm sounds when Upper or Lower Limit is surpassed.

LL:

LEVel *n*
Sets Modulation Meter Lower Limit to *n*%. Range of *n* is 0.0 to 100.0.

STATe *b*
Enables Modulation Meter Lower Limit if *b* is 1, disables if *b* is 0. When enabled, exceeding the Lower Limit sets bit 11 of the Instrument Status Register to one (activating bit 13 of the Questionable Status Register and bit 3 of the Status Byte) (see 3-16). When the Lower Limit and Alarm (**M_MOD:ALARM 1**) are enabled, exceeding the Lower Limit activates an audio alarm.

PEAK?
Returns Modulation Meter Peak reading as a percentage (0.0 to 100.0).

PH *b*
Enables Modulation Meter Peak Hold Feature if *b* is 1, disables if *b* is 0.

RANGe:

AUTO
Sets Modulation Meter range to Autorange.

UPPer *n*
Sets Modulation Meter range to *n*%. Select 40 or 100.

Example:
```
M_MOD:RANG:UPP 100     // Sets AM Modulation Meter Range to 100%.
M_MOD:PH 1             // Enables AM Modulation Meter Peak Hold.
M_MOD:LL:LEV 20.5      // Sets a Lower Limit of 20.5%.
M_MOD:LL:STAT 1        // Enables Lower Limit.
M_MOD:ALARM 1          // Enables Alarm.
```

RCL *n*
Recalls Modulation Meter environment (routings and settings) stored in memory location *n*. Range of *n* is 1 to 9.

STORe *n*
Stores current Modulation Meter environment (routings and settings) in memory location *n*. Range of *n* is 1 to 9.

M_MOD:

UL:

LEVel *n*
Sets Modulation Meter Upper Limit to *n*%. Range of *n* is 0.0 to 100.0.

STATe *b*
Enables Modulation Meter Upper Limit if *b* is 1, disables if *b* is 0. When enabled, exceeding the Upper Limit sets bit 10 of the Instrument Status Register to one (activating bit 13 of the Questionable Status Register and bit 3 of the Status Byte) (see 3-16). When the Upper Limit and Alarm (**M_MOD:ALARM 1**) are enabled, exceeding the Upper Limit activates an audio alarm.

M_MOD?
Returns a Modulation Meter reading as a percentage (0.0 to 100.0).

Example:
```
SCREEN:MOD              // Displays Modulation Meter Operation Screen.
M_MOD:RANG:AUTO         // Sets Modulation Meter Range to Autorange.
M_MOD:UL:LEV 30         // Sets Modulation Meter Upper Limit to 30%.
M_MOD:UL:STAT 1         // Enables Upper Limit.
M_MOD:LL:LEV 5          // Sets AM Modulation Meter Lower Limit to 5%.
M_MOD:LL:STAT 1         // Enables Lower Limit.
M_MOD:ALARM 1           // Enables Alarm.
M_MOD:PH 1              // Enables Peak Hold feature.
M_MOD:PEAK?             // Queries Modulation Meter Peak reading.
M_MOD?                  // Queries Modulation Meter reading.
```

4-11-6 DISTORTION METER COMMANDS

M_DIST:

ALARM *b*
Enables Distortion Meter Alarm if *b* is 1, disables if *b* is 0. Enabled Alarm sounds when Upper or Lower Limit is surpassed.

AVErage *b*
Enables Average Feature if *b* is 1, disables if *b* is 0.

FILTer *f*
Sets Notch Filter frequency to *f* Hz. Range of *f* is 600 to 1400.

INPut:*type*
Sets Distortion Meter Input to *type*. Select DEMOD (Demod Audio), SINAD (SINAD/BER), XAUDio (Ext Mod) or FGEN (Func Gen).

LL:

LEVel *n*
Sets Lower Limit to *n*%. Range of *n* is 0.0 to 20.0.

STATe *b*
Enables Distortion Meter Lower Limit if *b* is 1, disables if *b* is 0. When enabled, exceeding the Lower Limit sets bit 13 of the Instrument Status Register to one (activating bit 13 of the Questionable Status Register and bit 3 of the Status Byte) (see 3-16). When the Lower Limit and Alarm (**M_DIST:ALARM 1**) are enabled, exceeding the Lower Limit activates an audio alarm.

Example:
```
M_DIST:INP:XAUD        // Selects EXT MOD IN Connector for
                       // Distortion Meter Input.
M_DIST:FILT 770        // Sets Notch Filter frequency to 770 Hz.
M_DIST:PH 1            // Enables Distortion Meter Peak Hold.
M_DIST:LL:LEV 5.3      // Sets a Lower Limit of 5.3 kHz.
M_DIST:LL:STAT 1       // Enables Lower Limit.
M_DIST:ALARM 1         // Enables Alarm.
```

PEAK?
Returns a Distortion Meter Peak reading as a percentage (0.0 to 20.0).

PH *b*
Enables Distortion Meter Peak Hold Feature if *b* is 1, disables if *b* is 0. Peak Hold takes effect only after **SCREEN:DISTortion** command.

RCL *n*
Recalls Distortion Meter environment (routings and settings) stored at memory location *n*. Range of *n* is 1 to 9.

M_DIST:

SELECT:

CWEight
Selects the C-Weight Filter.

LPASs *f*
Selects a Low-Pass Filter with cutoff frequency of *f* Hz. Range of *f* is 100 to 30000.

Example:
```
M_DIST:SELECT:LPAS 15000  // Selects a Low-Pass Filter with a
                          // 15 kHz cutoff frequency.
```

STORe *n*
Stores current Distortion Meter environment (routings and settings) in memory location *n*. Range of *n* is 1 to 9.

UL:

LEVel *n*
Sets Upper Limit to *n*%. Range of *n* is 0.0 to 20.0.

STATe *b*
Enables Distortion Meter Upper Limit if *b* is 1, disables if *b* is 0. When enabled, exceeding the Upper Limit sets bit 12 of the Instrument Status Register to one (activating bit 13 of the Questionable Status Register and bit 3 of the Status Byte) (see 3-16). When the Upper Limit and Alarm (**M_DIST:ALARM 1**) are enabled, exceeding the Upper Limit activates an audio alarm.

M_DIST?
Returns a Distortion Meter reading as a percentage (0.0 to 20.0).

Example:
```
SCREEN:DIST              // Displays Distortion Meter Operation Screen.
M_DIST:INP:DEMOD         // Selects Demodulated Audio as the Distortion
                         // Meter Input.
M_DIST:FILT 1000         // Sets Notch Filter frequency to 1 kHz.
M_DIST:UL:LEV 15         // Sets Distortion Meter Upper Limit to 15%.
M_DIST:UL:STAT 1         // Enables Upper Limit.
M_DIST:LL:LEV 4          // Sets Distortion Meter Lower Limit to 4%.
M_DIST:LL:STAT 1         // Enables Lower Limit.
M_DIST:ALARM 1           // Enables Alarm.
M_DIST:SELECT:LPAS 20000    // Selects a Low-Pass Filter with a 20 kHz
                            // cutoff frequency.
M_DIST:AVE 1             // Enables Averaging.
M_DIST:PH 1              // Enables Peak Hold feature.
M_DIST:PEAK?             // Queries Distortion Meter Peak reading.
M_DIST?                  // Queries Distortion Meter reading.
```

4-11-7 SINAD METER COMMANDS

M_SINAD:

ALARM *b*
Enables SINAD Meter Alarm if *b* is 1, disables if *b* is 0. Enabled Alarm sounds when Upper or Lower Limit is surpassed.

AVErage *b*
Enables Average Feature if *b* is 1, disables if *b* is 0.

FILTer *f*
Sets Notch frequency to *f* Hz. Range of *f* is 600 to 1400.

INPut:*type*
Selects *type* as the SINAD Meter Input. Select DEMOD (Demod Audio), SINAD (SINAD/BER), XAUDio (Ext Mod) or FGEN (Func Gen).

LL:

LEVel *n*
Sets SINAD Meter Lower Limit to *n* dB. Range of *n* is 3.0 to 40.0.

STATe *b*
Enables SINAD Meter Lower Limit if *b* is 1, disables if *b* is 0. When enabled, exceeding Lower Limit sets bit 1 of the Instrument Summary Status Register to one (activating bit 1 of the Instrument Status Register, bit 13 of the Questionable Status Register and bit 3 of the Status Byte) (see 3-16). When the Lower Limit and Alarm (**M_SINAD:ALARM 1**) are enabled, exceeding the Lower Limit activates an audio alarm.

Example:
```
M_SINAD:INP:XAUD        // Selects EXT MOD IN Connector for
                        // Distortion Meter Input.
M-SINAD:FILT 850        // Sets Notch Filter frequency to 850 Hz.
M_SINAD:PH 1            // Enables Distortion Meter Peak Hold.
M_SINAD:LL:LEV 4        // Sets a Lower Limit of 4 dB.
M_SINAD:LL:STAT 1       // Enables Lower Limit.
```

PEAK?
Returns a SINAD Meter Peak reading in dB (3.0 to 40.0).

PH *b*
Enables SINAD Meter Peak Hold Feature if *b* is 1, disables if *b* is 0.

RCL *n*
Recalls SINAD Meter environment (routings and settings) stored in memory location *n*. Range of *n* is 1 to 9.

RESolution *n*
Sets SINAD Meter readout resolution to *n* dB. *n* is 0.1 or 0.5.

RESolution?
Returns current resolution setting in dB (0.1 or 0.5).

M_SINAD:

SELECT:

CWEight
Selects C-Weight Filter.

LPASs *f*
Selects Low-Pass Filter with cutoff frequency of *f* Hz. Range of *f* is 100 to 30000.

STORe *n*
Stores current SINAD Meter environment (routings and settings) in memory location *n*. Range of *n* is 1 to 9.

UL:

LEVel *n*
Sets SINAD Meter Upper Limit to *n* dB. Range of *n* is 3.0 to 40.0.

STATe *b*
Enables SINAD Meter Upper Limit if *b* is 1, disables if *b* is 0. When enabled, exceeding Upper Limit sets bit 0 of the Instrument Summary Status Register to one (activating bit 1 of the Instrument Status Register, bit 13 of the Questionable Status Register and bit 3 of the Status Byte) (see 3-16). When the Upper Limit and Alarm (**M_SINAD:ALARM 1**) are enabled, exceeding the Upper Limit activates an audio alarm.

M_SINAD?
Returns a SINAD Meter reading in dB (3.0 to 40.0).

Example:
```
SCREEN:SINAD            // Displays SINAD Meter Operation Screen.
M_SINAD:INP:DEMOD       // Selects Demodulated Audio as the
                        // SINAD Meter.
M_SINAD:FILT 1100       // Sets Notch Filter frequency to 1100 Hz.
M_SINAD:UL:LEV 15       // Sets SINAD Meter Upper Limit to 15 dB.
M_SINAD:UL:STAT 1       // Enables Upper Limit.
M_SINAD:LL:LEV 4        // Sets SINAD Meter Lower Limit to 4 dB.
M_SINAD:LL:STAT 1       // Enables Lower Limit.
M_SINAD:SELECT:LPAS 15000   // Selects a Low-Pass Filter with a 15 KHz
                            // cutoff frequency.
M_SINAD:AVE 1           // Enables Averaging.
M_SINAD:PH 1            // Enables Peak Hold feature.
M_SINAD:PEAK?           // Queries SINAD Meter Peak reading.
M_SINAD?                // Queries SINAD Meter reading.
```

4-11-8 SIGNAL STRENGTH METER COMMANDS

M_SIG:

PEAK?
Returns a Signal Strength Meter Peak reading (0 to 100).

PH *b*
Enables Peak Hold Feature if *b* is 1, disables if *b* is 0.

RCL *n*
Recalls Signal Strength Meter environment (routings and settings) stored in memory location *n*. Range of *n* is 1 to 9.

STORe *n*
Stores current Signal Strength Meter environment (routings and settings) in memory location *n*. Range of *n* is 1 to 9.

M_SIG?
Returns a Signal Strength Meter reading (0 to 100).

```
Example:  M-SIG:PH 1            // Enables Peak Hold feature.
          M_SIG:PEAK?           // Queries Signal Strength Meter Peak reading.
          M_SIG?                // Queries Signal Strength Meter reading.
```

4-11-9 BIT ERROR RATE (BER) METER COMMANDS

M_BER:

PATtern:

FIXED
Selects Fixed pattern for the BER Meter test data.

RANDom
Selects Random pattern for BER Meter test data.

USER *nn*
Selects a USER pattern (could be a variable or expression) for the BER Meter test data using the specified 8 bit pattern *nn*. Patterns entered that are not base 10 are preceded with # character and letter signifying the number base: H (Hexadecimal), B (Binary) or Q (Octal). Pattern is displayed on the CRT in Hexadecimal.

```
Example: M_BER:PAT:USER #Q216  // Selects a User Pattern of 8E hexadecimal
                               // (216 Octal) for the BER Meter test data.
         X=#HC0                // Assigns variable to a usable pattern.
         M_BER:PAT:USER X+3    // Selects a User Pattern of C3 hexadecimal.
```

POLarity:

NEGative
Selects Negative Polarity for the BER Meter.

POSitive
Selects Positive Polarity for the BER Meter.

M_BER:

RATE *n*
Sets BER Meter rate to *n*. Select from:

75	150	300
600	1200	2400
4800	16000	

RATE?
Returns the BER Meter Rate setting.

RCL *n*
Recalls the BER Meter environment (routings and settings) stored in memory location *n*. Range of *n* is 1 to 9.

SIZE *n*
Sets BER Meter block size in bits to *n*. Range of *n* is 100 to 100000.

```
Example: M_BER:POL:NEG          // Selects Negative Polarity for the BER Meter.
         M_BER:RATE 1200        // Sets the data rate to 1200 bps.
         M_BER:SIZE 1000        // Sets the data pattern size to 1000 bits.
```

SIZE?
Returns the BER Meter block size setting.

STORe *n*
Stores current BER Meter environment (routings and settings) in memory location *n*. Range of *n* is 1 to 9.

TYPE:*xxx*
Selects *xxx* for the Bit Error Rate Type. Select GENerator, RECeiver, DUPlex or BASEband.

M_BER?
Returns the number of errors for the last pass.

```
Example: M_BER:TYPE:DUP         // Selects Duplex for BER Meter Type.
         M_BER:RATE 2400        // Sets BER Meter data rate to 2400 bps.
         M_BER:SIZE 10000       // Sets BER Meter Pattern Size to 10,000 bits.
         M_BER:POL:POS          // Sets BER Meter Polarity to Positive.
         M_BER:PAT:RAND         // Selects the Random Pattern Type for the BER
                                // Meter test.
         M_BER?                 // Queries a bit error rate reading.
```

M_DMM:

ALARM *b*
Enables Multimeter Alarm if *b* is 1, disables if *b* is 0. Enabled Alarm sounds when Upper or Lower Limit is surpassed.

FUNCtion:

CURRent:

AC
Selects the AC Ammeter for the Multimeter Function.

DC
Selects the DC Ammeter for the Multimeter Function.

RESistance
Selects the Ohmmeter for the Multimeter Function.

VOLTage:

AC
Selects the AC Voltmeter for the Multimeter Function.

DC
Selects the DC Voltmeter for the Multimeter Function.

FUNCtion?
Returns active Multimeter Function (CURR:AC, CURR:DC, RES, VOLT:AC or VOLT:DC).

INPut:IMPedance *n*
Sets Input Impedance to *n* ohms. Select 150, 600 or 1e6. Command ignored if not in AC Voltmeter Function.

Example:
```
M_DMM:FUNC:CURR:DC     // Selects DC Ammeter for the Multimeter
                       // Function.
M_DMM:FUNC?            // Queries Multimeter Function.  DCC is
                       // returned.
M_DMM:FUNC:VOLT:AC     // Selects AC Voltmeter for the Multimeter
                       // Function.
M_DMM:INP:IMP 1E6      // Sets Input Impedance at 1 MΩ.
```

LL:

LEVel *n*
Sets the Multimeters Lower Limit to *n*. Table 4-1 lists units and ranges for *n*.

STATe *b*
Enables the Multimeters Lower Limit if *b* is 1, disables if *b* is 0. When enabled, exceeding Lower Limit sets bit 3 of the Instrument Summary Status Register to one (activating bit 1 of the Instrument Status Register, bit 13 of the Questionable Status Register and bit 3 of the Status Byte) (see 3-16). When the Lower Limit and Alarm (**M_DMM:ALARM 1**) are enabled, exceeding the Lower Limit activates an audio alarm.

M_DMM:

Function	ACC, DCC		ACV, DCV		Ohmmeter	
Range	20 and 200 mA	2 A and 20 A	200 mV	2, 20, 200, 2000 V	200 Ω	2 kΩ to 20 MΩ
n Units	A		V		kΩ	
Range of *n*	0.00000 to 0.19990	0.000 to 19.990	0.0000 to 0.1999	0.00 to 1000.00	0.0000 to 0.1999	0.000 to 19990

Table 4-1 DMM Upper and Lower Limit Ranges and Units

PH *b*
Enables Multimeter Peak Hold feature if *b* is 1, disables if *b* is 0.

RANGe:

AUTO
Sets the Multimeter range to Autorange.

UPPer *n*
For AC or DC Voltmeter, sets range to *n* volts. Settings for *n* are: 0.2, 2, 20, 200 or 2000.

For AC or DC Ammeter, sets range to *n* amps. Settings for *n* are: 0.02, 0.2, 2 or 20.

For Ohmmeter, sets range to *n* kΩ. Settings for *n* are: 0.2, 2, 20, 200, 2000 or 20000.

NOTE: M_DMM:RANGe:AUTO and **M_DMM:RANGe:UPPer** commands must be followed by a **SCREEN:DMM** command.

Example:
```
M_DMM:FUNC:CURR:AC     // Selects AC Ammeter for the Multimeter
                       // Function.
M_DMM:UPP 2            // Sets Ammeter range to 2 A.
M_DMM:FUNC:VOLT:AC     // Selects AC Voltmeter for the Multimeter
                       // Function.
M_DMM:RANG:UPP .2      // Sets Ohmmeter range to 200 Ω.
```

RCL *n*
Recalls Digital Multimeter environment (routings and settings) stored in memory location *n*. Range of *n* is 1 to 9.

STORe *n*
Stores current Digital Multimeter environment (routings and settings) in memory location *n*. Range of *n* is 1 to 9.

UL:

LEVel *n*
Sets Multimeters Upper Limit to *n*. Table 4-1 lists units and ranges for *n*.

STATe *b*
Enables Multimeters Upper Limit if *b* is 1, disables if *b* is 0. When enabled, exceeding Upper Limit sets bit 2 of the Instrument Summary Status Register to one (activating bit 1 of the Instrument Status Register, bit 13 of the Questionable Status Register and bit 3 of the Status Byte) (see 3-16). When the Upper Limit and Alarm (**M_DMM:ALARM 1**) are enabled, exceeding the Upper Limit activates an audio alarm.

M_DMM?
Returns a Multimeter reading in the current Function units according to Table 4-1. See also **MEASure:** commands (4-15).

```
Example:  M_DMM:FUNC:CURR:DC    // Selects DC Ammeter for the Multimeter
                                // Function.
          M_DMM:RANG:LEV 0.2    // Sets Ammeter range to 200 mA.
          M_DMM:PH 1            // Enables Peak Hold feature.
          M_DMM:ALARM 1         // Enables Alarm.
          M_DMM:UL:LEV 150      // Sets Upper Limit to 150 mA.
          M_DMM:UL:STAT 1       // Enables Upper Limit.
          M_DMM:LL:LEV 30       // Sets Lower Limit to 30 mA.
          M_DMM:LL:STAT 1       // Enables Lower Limit.
```

4-11-11 PHASE METER COMMANDS

M_PM:

ALARM *b*
Enables Phase Meter Alarm if *b* is 1, disables if *b* is 0. Enabled Alarm sounds when a limit is surpassed.

LL:

LEVel *n*
Sets the Phase Meter Lower Limit to *n* radians. Range of *n* is 0.00 to 10.00.

STATe *b*
Enables the Phase Meter Lower Limit if *b* is 1, disables if *b* is 0. When enabled, exceeding Lower Limit sets bit 5 of the Instrument Summary Status Register to one (activating bit 1 of the Instrument Status Register, bit 13 of the Questionable Status Register and bit 3 of the Status Byte) (see 3-16). When the Lower Limit and Alarm (**M_PM:ALARM 1**) are enabled, exceeding the Lower Limit activates an audio alarm.

PH *b*
Enables the Phase Meter Peak Hold Feature if *b* is 1, disables if *b* is 0.

RANGe:

AUTO
Sets Phase Meter range to Autorange.

UPPer *n*
Sets Phase Meter range to *n* radians. Select 1, 5 or 10.

```
Example: M_PM:RANG:UPP 5        // Sets Phase Meter Range to 5 radians.
         M_PM:PH 1              // Enables Phase Meter Peak Hold.
         M_PM:LL:LEV 1.65       // Sets a Lower Limit of 1.65 radians.
         M_PM:LL:STAT 1         // Enables Lower Limit.
         M_PM:ALARM 1           // Enables Alarm.
```

RCL *n*
Recalls Phase Meter environment (routings and settings) stored in memory location *n*. Range of *n* is 1 to 9.

M_PM:

STORe *n*
Stores current Phase Meter environment (routings and settings) in memory location *n*. Range of *n* is 1 to 9.

UL:

LEVel *n*
Sets Phase Meter Upper Limit to *n* radians. Range of *n* is 0.00 to 10.00.

STATe *b*
Enables Phase Meter Upper Limit if *b* is 1, disables if *b* is 0. When enabled, exceeding Upper Limit sets bit 4 of the Instrument Summary Status Register to one (activating bit 1 of the Instrument Status Register, bit 13 of the Questionable Status Register and bit 3 of the Status Byte) (see 3-16). When the Upper Limit and Alarm (**M_PM:ALARM 1**) are enabled, exceeding the Upper Limit activates an audio alarm.

M_PM?
Returns a Phase Meter reading in radians (0.00 to 10.00).

Example:
```
SCREEN:PM              // Displays Phase Meter Operation Screen.
M_PM:RANG:AUTO         // Sets Phase Meter Range to Autorange.
M_PM:UL:LEV 7          // Sets Phase Meter Upper Limit to 7 radians.
M_PM:UL:STAT 1         // Enables Upper Limit.
M_PM:LL:LEV .5         // Sets Phase Meter Lower Limit to 0.5 radians.
M_PM:LL:STAT 1         // Enables Lower Limit.
M_PM:ALARM 1           // Enables Alarm.
M_PM:PH 1              // Enables Peak Hold feature.
M_PM:PEAK?             // Queries Phase Meter Peak reading.
M_PM?                  // Queries Phase Meter reading.
```

4-11-12 DEVIATION METER (RMS) COMMANDS

M_DRMS:

ALARM *b*
Enables Deviation Meter (RMS) Alarm if *b* is 1, disables if *b* is 0. Enabled Alarm sounds when a limit is surpassed.

AVErage *b*
Enables Deviation Meter (RMS) Averaging if *b* is 1, disables if *b* is 0.

LL:

LEVel *f*
Sets Deviation (RMS) Lower Limit to *f* kHz. Range of *f* is 0.00 to 10.00.

STATe *b*
Enables Deviation (RMS) Meter Lower Limit if *b* is 1, disables if *b* is 0. When the Lower Limit and Alarm (**M_DRMS:ALARM 1**) are enabled, exceeding the Lower Limit activates an audio alarm.

PH *b*
Enables Deviation Meter (RMS) Peak Hold feature if *b* is 1, disables if *b* is 0.

RANGe:

AUTO
Sets Deviation Meter (RMS) range to Autorange.

UPPer *f*
Sets Deviation Meter (RMS) range to *f* kHz. Select 2, 5 or 10.

Example:
```
M_DRMS:RANG:UPP 10      // Sets Deviation Meter (RMS) Range to
                        // 10 kHz.
M_DRMS:PH 1             // Enables Deviation Meter (RMS) Peak Hold.
M_DRMS:LL:LEV 1.5       // Sets a Lower Limit of 1.5 kHz.
M_DRMS:LL:STAT 1        // Enables Lower Limit.
M_DRMS:ALARM 1          // Enables Alarm.
```

RCL *n*
Recalls Deviation Meter (RMS) environment (routings and settings) stored in memory location *n*. Range of *n* is 1 to 9.

STORe *n*
Stores current Deviation Meter (RMS) environment (routings and settings) in memory location *n*. Range of *n* is 1 to 9.

M_DRMS:

UL:

LEVel *f*
Sets Deviation (RMS) Upper Limit to *f* kHz. Range of *f* is 0.00 to 10.00.

STATe *b*
Enables Deviation Meter (RMS) Upper Limit if *b* is 1, disables if *b* is 0. When the Upper Limit and Alarm (**M_DRMS:ALARM 1**) are enabled, exceeding the Upper Limit activates an audio alarm.

M_DRMS?
Returns a Deviation Meter (RMS) reading in kHz (0.00 to 10.00).

Example:
```
SCREEN:DRMS              // Displays Deviation Meter (RMS) Operation
                         // Screen.
M_DRMS:RANG:AUTO         // Sets Deviation Meter (RMS) Range to Autorange.
M_DRMS:UL:LEV 8.4        // Sets Deviation Meter (RMS) Upper Limit
                         // to 8.4 kHz.
M_DRMS:UL:STAT 1         // Enables Upper Limit.
M_DRMS:LL:LEV 2          // Sets Deviation Meter (RMS) Lower Limit
                         // to 2 kHz.
M_DRMS:LL:STAT 1         // Enables Lower Limit.
M_DRMS:ALARM 1           // Enables Alarm.
M_DRMS:PH 1              // Enables Peak Hold feature.
M_DRMS:AVE 1             // Enables Averaging.
M_DRMS?                  // Queries Deviation Meter (RMS) reading.
```

4-11-13 PHASE METER (RMS) COMMANDS

M_PMRMS:

ALARM *b*
Enables Phase Meter (RMS) Alarm if *b* is 1, disables Alarm if *b* is 0. Enabled Alarm sounds when a limit is surpassed.

LL:

LEVel *n*
Sets Phase Meter (RMS) Lower Limit to *n* radians. Range of *n* is 0.00 to 10.00.

STATe *b*
Enables Phase Meter (RMS) Lower Limit if *b* is 1, disables Lower Limit if *b* is 0. When the Lower Limit and Alarm (**M_PMRMS:ALARM 1**) are enabled, exceeding the Lower Limit activates an audio alarm.

PH *b*
Enables Phase Meter (RMS) Peak Hold feature if *b* is 1, disables Peak Hold feature if *b* is 0.

RANGe:

AUTO
Sets Phase Meter (RMS) range to Autorange.

UPPer *n*
Sets Phase Meter (RMS) range to *n* radians. Select 1, 5 or 10.

RCL *n*
Recalls Phase Meter (RMS) environment (routings and settings) stored in memory location *n*. Range of *n* is 1 to 9.

STORe *n*
Stores current Phase Meter (RMS) environment (routings and settings) in memory location *n*. Range of *n* is 1 to 9.

Example:
```
M_PMRMS:RANG:AUTO      // Sets Phase Meter (RMS) range to Autorange.
M_PMRMS:PH 1           // Enables Phase Meter (RMS) Peak Hold.
M_PMRMS:STOR 3         // Stores state of Phase Meter (RMS) to
                       // Phase Meter (RMS) memory location 3.
```

UL:

LEVel *n*
Sets Phase Meter (RMS) Upper Limit to *n* radians. Range of *n* is 0.00 to 10.00.

STATe *b*
Enables Phase Meter (RMS) Upper Limit if *b* is 1, disables Upper Limit if *b* is 0. When the Upper Limit and Alarm (**M_PMRMS:ALARM 1**) are enabled, exceeding the Upper Limit activates an audio alarm.

M_PMRMS?
Returns a Phase Meter (RMS) reading in radians (0.00 to 10.00).

Example:
```
SCREEN:PMRMS          // Displays Phase Meter (RMS) Operation Screen.
M_PMRMS:UPP 5         // Sets Phase Meter (RMS) Range to 5 radians.
M_PMRMS:UL:LEV 4.6    // Sets Phase Meter (RMS) Upper Limit to
                      // 4.6 radians.
M_PMRMS:UL:STAT 1     // Enables Phase Meter (RMS) Upper Limit.
M_PMRMS?              // Queries Phase Meter (RMS) reading.
```

4-11-14 AF LEVEL METER

M_VRMS?
Returns a Voltage RMS reading of the received AF Level (0.00 to 10.00).

4-12 CELLULAR AMPS

4-12-1 CELLULAR AMPS COMMANDS

The AMPS Cell Site Monitor and Mobile Simulator is option 10 of the FM/AM-1600S and FM/AM-1600CSA. The following commands remotely control this option. Except for **CELL:CHAN** and **CELL:CHAN?**, these commands have no effect if the AMPS Cell Site Monitor is not installed.

CELL: commands remotely operate the Cell Site Monitor and require a Cell Site Monitor screen to be displayed on the CRT. **CELL:GEN** commands remotely operate the Mobile Simulator and require a Mobile Simulator screen to be displayed on the CRT.

NOTE: A Cellular query command (command with ?) returns the value once, returning -1 for further queries until additional data occurs.

CELL:

ACTion?
Returns the Global Action value or -1 if not available.

BIS?
Returns latest Busy-Idle Status bit value or -1 if not available.

B_I?
Returns the Busy-Idle bit value, 0 for busy, 1 for idle. -1 is returned if Busy-Idle bit is not available.

BOTH
Selects stream A and B words for decoding.

C12?
Returns latest C12 bit value or -1 if not available.

C13?
Returns latest C13 bit value or -1 if not available.

CAPTure:

MIN "*xxx/xxx-xxxx*"
Selects MIN to capture. Range of *x* is 0 to 9, A to D or a to d. An x, X, # or * can be used as a wildcard character. MIN string must be entered as shown. Does not select Capture Mode.

MIN?
Returns current MIN used for capture (string).

MODE:*xxx*
Selects *xxx* as Capture Mode. Select MIN, ORDer, BOTH or OFF for *xxx*.

MODE?
Returns the current Capture Mode.

CELL:

CAPTure:

ORDer:*xxx*
Selects *xxx* as Order to capture. Does not select Capture Mode. Select one of the following for *xxx*:

PAGE	ALERT	RELease
REORDer	SALERT	AUDIT
SNDAddr	INTERCEPT	MAINTenance
POWer	DRETRY	AUTREG
AINTERCEPT	AREORDer	AALERT
VCDES		

ORDer?
Returns current Order used for capture (string).

CHANnel *n*
Selects Cellular Channel *n* for testing. Range of *n* is 0 to 1023.

CHANnel?
Returns the current Cellular Channel.

CHANPOS1?
Returns Channel Position 1 (seven bit value) or -1 if not applicable.

CHANPOS2?
Returns Channel Position 2 (seven bit value) or -1 if not applicable.

CHANPOS3?
Returns Channel Position 3 (seven bit value) or -1 if not applicable.

CHANPOS4?
Returns Channel Position 4 (seven bit value) or -1 if not applicable.

CHANPOS5?
Returns Channel Position 5 (seven bit value) or -1 if not applicable.

CHANPOS6?
Returns Channel Position 6 (seven bit value) or -1 if not applicable.

CMAC?
Returns Control Mobile Attenuation Code (three bit value) or -1 if not available.

CMAX_1?
Returns Number of Access Channels minus one (seven bit value) or -1 if not available.

CPA?
Returns the Combined Paging/Access bit or -1 if not available.

DCC?
Returns Digital Color Code (two bit value) or -1 if not available.

DIGITs?
Returns the Call Address (16 character string) or -1 if not available.

CELL:

DSCC?
Returns the Digital SAT Color Code (2 bit value) or -1 if not available.

DTX?
Returns the Discontinuous Transmission bit value or -1 if not available.

E?
Returns the Extended Address bit value or -1 if not available.

EF?
Returns the Expanded Protocol Forward Control Channel Indicator (one bit value) or -1 if not available.

END?
Returns the End Indication bit value or -1 if not available.

EP?
Returns the Extended Protocol Capable value (one bit value) or -1 if not available.

ESN?
Returns the Electronic Serial Number (32 bit value) or -1 if not available.

FORMat:

AMPS
Sets Channel Format to AMPS.

NT400
Sets Channel Format to NT400.

FORMat?
Returns the Channel Format setting.

LOCALCTRL1?
Returns the first position Local Control field (16 bit value) or -1 if not available.

LOCALCTRL2?
Returns the second position Local Control field (16 bit value) or -1 if not available.

MAXBusy:

OTHer?
Returns maximum number of busy occurrences for other than page responses (four bit value) or -1 if not available.

PGR?
Returns maximum number of busy occurrences for page responses (four bit value) or -1 if not available.

CELL:

MAXSztr:

OTHer?
Returns maximum number of seizure tries for other than page responses (four bit value) or -1 if not available.

PGR?
Returns maximum number of seizure tries for page responses (four bit value) or -1 if not available.

MIN?
Returns the current MIN value in AAA/DDD-DDDD or BBB-BBBB format as a string or -1 if not available.

MSL?
Returns the Message Length (eight bit value) or -1 if not available.

MST?
Returns the Message Type (five bit value) or -1 if not available.

N_1?
Returns the Number of Paging Channels, minus one, to be scanned by the Mobile Equipment (five bit value) or -1 if not available.

NAWC?
Returns the Number of Additional Words Coming (three bit value) or -1 if not available.

NEWACC?
Returns the New Access Channel Starting point (11 bit value) or -1 if not available.

OLC?
Returns the Overload Control Class (15 bit value) or -1 if not available.

ORDer?
Returns the current Order as a string (PAGE, ALERT, RELEASE, REORDER, SALERT, AUDIT, SNDADDR, INTERCEPT, MAINTENANCE, POWER, DRETRY, AUTREG, AINTERCEPT, AREORDER, AALERT, VCDES) or -1 if not available.

PDSCC?
Returns the Present Digital SAT Color Code (two bit value) or -1 if not available.

PSCC?
Returns the Present SAT Color Code (two bit value) or -1 if not available.

RCF?
Returns the Read Control-Filler bit value or -1 if not available.

REGH?
Returns the Registration for Home Mobile Stations value (one bit value) or -1 if not available.

REGID?
Returns the Registration Identification value (20 bit value) or -1 if not available.

CELL:

REGINCR?
Returns the Registration Increment (12 bit value) or -1 if not available.

REGR?
Returns the Registration for Roaming Mobile Stations (one bit value) or -1 if not available.

S?
Returns the Serial Number bit value or -1 if not available.

SCC?
Returns the Supervisory Audio Tone Color Code (two bit value) or -1 if not available.

SID?
Returns the System Identification Number (14 bit value) or -1 if not available.

VCHAN?
Returns the Voice Channel (1 to 1023) from Channel field or -1 if not available.

VMAC?
Returns the Voice Mobile Attenuation Code (three bit value) or -1 if not available.

WFOM?
Returns the Wait for Overhead Message bit or -1 if not available.

WORD?
Returns the current decoding word selection (A or B).

WORDA
Selects the stream A words for decoding.

WORDB
Selects the stream B words for decoding.

CELL:GEN commands operate the AMPS Mobile Simulator (part of option 10). **CELL:GEN:RECC** commands operate the Reverse Control Channel Screen and **CELL:GEN:RVC** commands operate the Reverse Voice Channel Screen.

CELL:

GEN:

CHANnel *n*
Selects an AMPS Reverse Channel to send messages to Cell Site. Range of *n* is 1 to 1023.

CHANnel?
Returns current selected channel.

DCC *x*
Selects Digital Color Code sent to Cell Site. Range of *x* is from 0 to 3.

DCC?
Returns current DCC setting.

CELL:

GEN:

DIGITs "*x*"
Selects simulated Called Address sent to Cell Site. Range of *x* is 0 to 9999999999999999.

DIGITs?
Returns current Called Address setting.

ESN:

DECimal *x*
Selects simulated Electronic Serial Number sent to Cell Site using decimal format. Range of *x* is 0 to 23116113919. ESN is displayed on screen in current format.

DECimal?
Returns current ESN setting in decimal format.

FORMat:

DECimal
Selects ESN decimal format.

HEX
Selects ESN hexadecimal format.

OCTal
Selects ESN octal format.

FORMat?
Returns current ESN format (DEC, HEX or OCT).

HEXadecimal *x*
Selects simulated ESN sent to Cell Site using hexadecimal format. Range of *x* is 0 to FFFFFFFF. ESN is displayed on screen in current format.

HEXadecimal?
Returns current ESN setting in hexadecimal format.

OCTal *x*
Selects simulated ESN sent to Cell Site using octal format. Range of *x* is 0 to 37777777777. ESN is displayed on screen in current format.

OCTal?
Returns current ESN setting in octal format.

MIN "*xxx/xxx-xxxx***"**
Selects simulated MIN sent to Cell Site. Range of *x* is 0 to 9.

MIN?
Returns current MIN setting (string).

CELL:

GEN:

MODL *f*
Sets Cellular Deviation Level to *f* kHz. Range of *f* is 0.0 to 25.0.

MODL?
Returns current Cellular Deviation Level in kHz.

RECC:

CONTinuous
Sends the last order selected continuously to the Cell Site.

ONCE
Sends the last order selected to the Cell Site once.

ORDer:

AUDIT
Selects Audit for the Receive Control Channel Order field.

AUTREG
Selects Autonomous Registration for the RECC Order field.

ORIGIN
Selects Origination for the RECC Order field.

PAGE
Selects Page Response for the RECC Order field.

ORDer?
Returns current RECC Order field setting (AUDIT, AUTREG, ORIGIN or PAGE).

STOP
Stops transmission of RECC order sent continuously.

RVC:

CONTinuous
Sends the last order selected continuously to the Cell Site.

ONCE
Sends the last order selected to the Cell Site once.

CELL:

GEN:

RVC:

ORDer:

AUDIT
Selects Audit for the Reverse Voice Channel Order field.

POWer *x*
Selects Power Level order of power level *x* for the RVC Order field. Range of *x* is 0 to 7.

SENDCall
Selects Send Called Address for the RVC Order field.

SNREQuest
Selects Autonomous Registration for the RVC Order field.

ORDer?
Returns current RVC Order field setting (AUDIT, POWER, SENDCALL or SNREQUEST).

STOP
Stops transmission of RVC order sent continuously.

SCM:

BANDwidth *x*
Selects Mobile Phone Bandwidth sent to Cell Site. Select 20 or 25 (MHz).

BANDwidth?
Returns current Mobile Phone Bandwidth setting.

POWer *x*
Selects Power Level of simulated Mobile Phone. Range of *x* is 1 to 4 (4 reserved).

POWer?
Returns current Power Level setting.

TRANsmit:

CONTinuous
Selects Continuous Transmission for simulated Mobile Phone.

DISCONTinuous
Selects Discontinuous Transmission for simulated Mobile Phone.

TRANsmit?
Returns current Continuous/Discontinuous Transmission setting.

4-12-2 REMOTE AMPS CELL SITE MONITOR EXAMPLE

The following macro performs the Receiver Setup for AMPS Cell Site Operation:

```
*DMC "CELL_SETUP",BEGIN        // Define a macro named CELL_SETUP.
SCREEN:REC                     // Displays the Receiver Operation Screen.
*WAI                           // Waits for command to finish executing.
REC:MOD:USER:MOD DATA          // Selects User Defined FM Data for Receiver
                               // Modulation.
REC:MOD:USER:FILT 30           // Selects a Low-Pass IF Filter with a 30 kHz
                               // cutoff frequency.
REC:MOD:USER:POST:LPAS 15      // Selects a Low-Pass Post Detection Filter
                               // with a 15 kHz cutoff frequency.
REC:AGC:USER:HIGH              // Selects User Defined High Speed for AGC.
M_DEV:RANG:UPP 10              // Sets Deviation Meter Range to 10 kHz.
M_AF:RES 1                     // Sets AF Meter Gate Time to 0.1 sec (1 Hz).
END                            // End macro CELL_SETUP.
```

The following macro queries cellular readings once a min number is received:

```
*DMC "CELL_TEST",BEGIN         // Define macro named Cell_Read.
STRING MIN,ORDER               // Declare strings needed.
VAR SCC,DCC,SID,VMAC           // Declare variables needed.
VAR CHN,CMX,N1,CMAC            // Declare variables needed.
SCREEN:CELL                    // Display AMPS Cell Site Monitor Screen.
*WAI                           // Wait for command execution to finish.
CELL:CHAN 327                  // Select Control Channel 327
DO                             // Loop while no Min is received.
  $=STR(CELL:MIN?)             // Query for a MIN.
  TPAUSE                       // Pauses macro to allow Test Set processing.
UNTIL $ != "-1"                // End of loop.
MIN=$                          // Set MIN string equal to received MIN.
ORDER=STR(CELL:ORDER?)         // Query Order.
SCC=CELL:SCC?                  // Query SCC value.
DCC=CELL:DCC?                  // Query DCC value.
SID=CELL:SID?                  // Query SID value.
VMAC=CELL:VMAC?                // Query VMAC value.
CHN=CELL:VCHAN?                // Query Voice Channel.
CMX=CELL:CMAX_1?               // Query CMAX value.
N1=CELL:N_1?                   // Query N-1 value.
CMAC=CELL:CMAC?                // Query CMAC value.
PPRINT STR(MIN)+","+STR(ORDER)   // Send MIN and Order to RS-232 Connector.
PPRINT SCC,DCC,SID,VMAC        // Send readings to RS-232 Connector.
PPRINT CHN,CMX,N1,CMAC         // Send readings to RS-232 Connector.
END                            // End of macro CELL_TEST.
```

4-13 PROGRAM COMMANDS

PROGram:STARTup:NAME "*name*"
Selects the macro *name* to execute at power up. The startup macro executes after the POWER Switch is pressed and the automatic self test is performed. The startup macro can be avoided by continually pressing STOP TEST CONTROL Key during the 2 beeps of the 1-2-4 beep startup procedure.

PROGram:STARTup:NAME?
Returns the name of the current power up macro designated by a **PROGram:STARTup:NAME** command.

PROGram:STARTup:DELETE
Deletes the power up designation of the power up macro. After this command is executed, there is no power up macro until a **PROGram:STARTup:NAME:** command is executed. This command does not delete the macro itself.

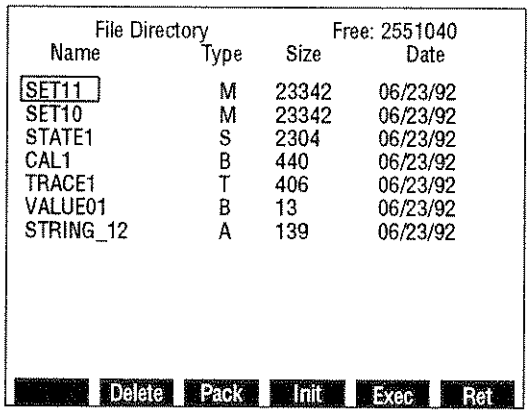
Example:
```
PROG:START:NAME "Main_Menu" // Selects macro Main_Menu as the power
                            // up macro.  This macro will now
                            // execute when the Test Set is turned on.
PROG:START:NAME?            // Queries the name of the current power
                            // up macro.  Main_Menu is returned.
PROG:START:DELETE           // Deletes the power up designation.
```

4-14 FLASH MEMORY FILE DIRECTORY OPERATION

The Flash Memory File Directory allows storage of various files including Calibration Data Sets and allows user to select a macro to be executed without remote instruction. Operate Flash Memory File Directory using following procedure:

STEP PROCEDURE

1. Press MTRS MODE Key. Press "AUX" Soft Function Key F6 to display Auxiliary Functions Menu. Press 7 DATA ENTRY Key to display File Directory Screen.



8617099

2. To load a macro from Flash memory into Test Set memory and execute it, move cursor to macro. Press "Exec" Soft Function Key F5.

3. To load a Calibration Data Set or a stored Test Set State (Test Set settings at time of store), move cursor to file. Press "Load" Soft Function Key F5.

4. To delete a file, move cursor to file. Press "Delete" Soft Function Key F2. Flash Memory space is not released until Pack operation is done.

5. To perform Pack operation, press "Pack" Soft Function Key F3. Pack releases memory space taken by deleted files.

   NOTE: Do not power off Test Set during Pack operation as files may be lost.

6. To Initialize Flash Memory, press "Init" Soft Function Key F4. Initializing clears Flash Memory and all files are lost.

7. To return to Auxiliary Functions Menu, press "Ret" Soft Function Key F6.

Files are stored in Flash Memory using remote commands only. Spectrum Analyzer and Oscilloscope Traces and variables are loaded into Test Set using remote commands. Table 4-2 contains remote commands used to operate Flash Memory File Directory.

COMMAND	RANGE/VALUE	DESCRIPTION
MMEMory:CATalog?		Returns Flash Memory status. First number returned is memory space used in bytes. Second number returned is memory space available in bytes. Remainder data is returned in sets of 3 consisting of file name, file type and file size for each file stored in Flash Memory.
MMEMory:CATalog:ENTRY? n	n is line number (index) in Flash Memory File Directory. Set n from 0 to 512.	Returns file entry (file name, file type, file size) for given index. Returns $$$ if past end of directory or --- for deleted file.
MMEMory:CATalog:USED?		Returns file space used, in bytes.
MMEMory:CATalog:FREE?		Returns available file space, in bytes.
MMEMory:DELete "f"	f is Flash Memory file name.	Deletes file but does not release memory space until Pack operation is done.
MMEMory:INITialize		Erases all files stored in Flash Memory.
MMEMory:INITialize?		Returns 1 if file system has been initialized, 0 otherwise.
MMEMory:LOAD:MACRo "m","f"	m is name of designated macro. f is Flash Memory file name.	Loads macros and variables stored as the file name from Flash Memory into Test Set memory. If m is *, designated macro is executed. If m is macro name, that macro is executed. If m is omitted (""), no macro is executed.
MMEMory:LOAD:STATe n,"f"	n is number of stored state of Test Set. Set n from 0 to 9. f is Flash Memory file name.	Loads Test Set State stored as f from Flash Memory into Auxiliary Functions "Store Parameters Menu" as entry n. (n=0 loads current state.)
MMEMory:LOAD:TRACe:SCOPe n,"f"	n is number of stored trace. Set n from 0 to 9. f is Flash Memory file name.	Loads Oscilloscope trace stored as f into Oscilloscope "Store Parameters Menu" as entry n. (n=0 loads live trace.)
MMEMory:LOAD:TRACe:ANLZ n,"f"	n is number of stored trace. Set n from 0 to 9. f is Flash Memory file name.	Loads Spectrum Analyzer trace stored as f into Spectrum Analyzer "Store Parameters Menu" as entry n. (n=0 loads live trace.)
MMEMory:LOAD:DATA v,"f"	v is name of variable. f is Flash Memory file name.	Loads variable stored as f into Test Set memory with name v.
MMEMory:LOAD:CALibration "f"	f is Flash Memory file name.	Loads Calibration Data from Flash Memory into Test Set memory.
MMEMory:PACK		Packs Flash Memory and frees memory space from deleted files. Do not power off Test Set during Pack function (may lose files). Do not Pack with a WINDOW open.
MMEMory:STORe:MACRo "m","f"	m is name of designated macro. f is Flash Memory file name.	Stores all Test Set macros and variables (except free variables) in Flash Memory as f with macro specified as designated macro.

Table 4-2 Flash Memory Remote Commands

COMMAND	RANGE/VALUE	DESCRIPTION
MMEMory:STORe:STATe n,"f"	n is number of stored state of Test Set. Set n from 0 to 9.	Stores entry n of Auxiliary Functions "Store Parameters Menu" as f in Flash Memory. (n=0 stores current state.)
MMEMory:STORe:TRACe:SCOPe n,"f"	n is number of stored trace. Set n from 0 to 9. f is Flash Memory file name.	Stores entry n (stored trace) of Oscilloscope "Store Parameters Menu" as f in Flash Memory. (n=0 stores live trace.)
MMEMory:STORe:TRACe:ANLZ n,"f"	n is number of stored trace. Set n from 0 to 9. f is Flash Memory file name.	Stores entry n (stored trace) of Spectrum Analyzer "Store Parameters Menu" as f in Flash Memory. (n=0 stores live trace.)
MMEMory:STORe:DATA v,"f"	v is name of variable. f is Flash Memory file name.	Stores variable v into Flash Memory as f.
MMEMory:STORe:CALibration "f"	f is Flash Memory file name.	Stores Test Set Calibration Data into Flash Memory.
MMEMory:TYPE? "f"	f is Flash Memory file name.	Returns file type. Returns null string if file does not exist.

Table 4-2 Flash Memory Remote Commands (continued)

Error messages are returned to Host when an error occurs. Refer to Table 4-3 for description of error messages.

ERROR NUMBER	ERROR DEFINITION	DESCRIPTION
220	Parameter Error	Incorrect number of parameters were entered with command.
224	Illegal Parameter Value	A parameter entered was not appropriate for command.
225	Out of Memory	Insufficient memory space to perform command.
250	Flash Storage Error	Indicates Flash Memory could not be erased or data could not be stored in Flash Memory.
253	Corrupt Media	Indicates Flash Memory not properly initialized. Initialize Flash Memory.
254	Media Full	Indicates insufficient Flash Memory space to perform command.
255	Directory Full	Indicates command not performed because 512 file names have been used.
256	File Name Not Found	Specified file not stored in Flash Memory.
257	File Name Error	Indicates command attempted to create file name already stored or file name syntax incorrect.

Table 4-3 Flash Memory Error Messages

4-15 GENERIC MEASURE COMMANDS

Generic commands use optional values, *e* to signify expected value and *r* to signify resolution. Expected values help determine Meter range. If *r* is omitted, the last resolution value is used.

MEASure:

AUDio? *e*, *r*
Returns an AF Meter frequency counter reading in Hz (0.0 to 200000.0).

CURRent:

AC? *e*
Returns a DMM ac current reading in amps (0.00000 to 19.990).

DC? *e*
Returns a DMM dc current reading in amps (0.00000 to 19.990).

FREQuency? *e*, *r*
Returns signal frequency reading in kHz (250.0 to 999999.9 kHz).

MIC?
Returns a 0 if receiving MIC/ACC Input or 1 otherwise.

PHASe? *e*
Returns a Phase Meter reading in radians (0.00 to 10.00).

POWer? *e*
Returns a Power Meter reading in mW (0.0 to 100000.0).

RESistance? *e*
Returns a DMM resistance reading in kΩ (0.0000 to 19990).

SINAD? *r*
Returns a SINAD Meter reading in dB (3.0 to 40.0). Select .1 or .5 as optional *r* in dB.

SQUelch?
Returns a 1 if squelch broken, 0 if squelch is unbroken.

TEMPerature:

AMBient?
Returns the ambient temperature in °C (0.00000 to 100.00000).

POWer?
Returns the Power Termination temperature in °C (0.00000 to 100.00000).

VOLTage:

AC? *e*
Returns a DMM ac voltage reading in volts (0.0000 to 1000.00).

DC? *e*
Returns a DMM dc voltage reading in volts (0.0000 to 1000.00).

MEASure:

VOLTage:SUPply? *n*
Returns a voltage measurement of *n* power supply in volts (0.00 to 20.00). Select -15, 5 or 15 for *n*.

4-16 INITIATE AND FETCH COMMANDS

INITiate and **FETCh** commands break the measure process into two commands. **INITiate** commands prepare the meter and **FETCh** commands return the reading. The event register of the Operation Instrument Status Register indicates when an **INITiate** command is completed (see example). Table 3-6 displays the Operation Instrument Status bits.

INITiate:

AF
Prepares the Audio Frequency Meter for a **FETCh:** command to take a reading.

DMM
Prepares the Audio Frequency Meter for a **FETCh:** command to take a reading.

RF
Prepares the Frequency Error Meter for a **FETCh:** command to take a reading.

FETCh:

AF?
Returns Audio Frequency Meter reading in kHz. Must be used with **INITiate** command.

DMM?
Returns Digital Multimeter reading. For ACV and DCV Multimeter Function, reading is returned in V. For ACC and DCC Multimeter Function, reading is returned in A. For Ohm Multimeter Function, reading is returned in kΩ. Must be used with **INITiate** command.

RF?
Returns Frequency Error Meter reading in kHz. Must be used with **INITiate** command.

The following example reads the Operation Instrument Register until the RF Meter is ready to be read.

Example:
```
*DMC "RF_Read",BEGIN          // Define macro named RF_Read.
N=0                           // Set variable equal to zero.
SCREEN:FREQ                   // Display Frequency Error Meter Operation
                              // Screen.
SCREEN:USER                   // Display User Screen.
*CLS                          // Clear all condition and event registers.
INIT:RF                       // Initialize for a RF Meter reading.
WHILE (N&8)=0                 // Loop until Frequency Error Meter bit of
  N=STAT:OPER:INSTR:EVENT?    // Operation Instrument event register is
                              // set to 1.
WEND
PRINT FETCH:RF?               // Print Frequency Error Meter reading.
END                           // End of macro RF_Read.
```

SECTION 5 - 1600S QUICK REFERENCE LIST

The 1600S Quick Reference List is a brief listing of the Specific and most General TMAC commands used with the FM/AM-1600S. The Quick Reference List is an aid to the experienced TMAC user. If more detailed information is needed, refer to the specified page.

COMMAND	RANGE	PAGE	DESCRIPTION
AF GENERATOR COMMANDS			
FGEN:			
DATA:			
MODL *n*	0.0 to 100.0, (%/AM) 0.0 to 25.0 (kHz/FM), 0.0 to 10.0 (radians/PM)	4-38	Sets Data Generator Modulation level.
MODL?		4-38	Returns Data Generator Modulation level setting.
MODulation:*type*	AM, FM, PM or OFF	4-38	Selects Data Generator Modulation type.
MODulation?		4-38	Returns selected Data Generator Modulation type.
STATe *b*	1 or 0	4-38	Enables/Disables Data Generator.
EXT:			
LEVel *n*	0 to 100 (%)	4-38	Sets External Modulation Proportional output level.
LEVel?		4-38	Returns External Modulation Proportional output level setting.
MODL *n*	0.0 to 100.0, (%/AM) 0.0 to 25.0 (kHz/FM), 0.0 to 10.0 (radians/PM)	4-38	Sets External Modulation level.
MODL?		4-38	Returns External Modulation level setting.
MODulation:*type*	AM, FM, PM or OFF	4-38	Selects External Modulation type.
MODulation?		4-39	Returns selected External Modulation type.
STATe *b*	1 or 0	4-39	Enables/Disables External Modulation.
GEN1:			
FREQuency *f*	0.0 to 40000.0	4-39	Sets AF Generator 1 frequency in Hz.
FREQuency?		4-39	Returns AF Generator 1 frequency setting in Hz.
LEVel *n*	0 to 100 (%)	4-39	Sets AF Generator 1 Proportional output level.
LEVel?		4-39	Returns set AF Generator 1 Proportional output level.
MODL *n*	0.0 to 100.0, (%/AM) 0.0 to 25.0 (kHz/FM), 0.0 to 10.0 (radians/PM)	4-39	Sets AF Generator 1 Modulation level.
MODL?		4-39	Returns AF Generator 1 Modulation level.
MODulation:*type*	AM, FM, PM or OFF	4-39	Selects AF Generator 1 Modulation type.
MODulation?		4-40	Returns selected AF Generator 1 Modulation type.
SHAPE:			
DC *n*	1, 0 or -1	4-40	Selects dc level for AF Generator 1 wave shape.
PULse:DCYCLe 50		4-40	Selects Pulse (50% duty cycle) for AF Generator 1.
RAMP		4-40	Selects ramp wave shape for AF Generator 1.
SIN		4-40	Selects sine wave shape for AF Generator 1.
SQU		4-40	Selects square wave shape for AF Generator 1.
TRI		4-40	Selects triangle wave shape for AF Generator 1.
STATe *b*	1 or 0	4-40	Enables/Disables AF Generator 1.
GEN2:			
FREQuency *f*	0.0 to 40000.0	4-39	Sets AF Generator 2 frequency in Hz.
FREQuency?		4-39	Returns AF Generator 2 frequency setting in Hz.
LEVel *n*	0 to 100 (%)	4-39	Sets AF Generator 2 Proportional output level.
LEVel?		4-39	Returns set AF Generator 2 Proportional output level.
MODL *n*	0.0 to 100.0, (%/AM) 0.0 to 25.0 (kHz/FM), 0.0 to 10.0 (radians/PM)	4-39	Sets AF Generator 2 Modulation level.
MODL?		4-39	Returns AF Generator 2 Modulation level.
MODulation:*type*	AM, FM, PM or OFF	4-39	Selects AF Generator 2 Modulation type.
MODulation?		4-40	Returns selected AF Generator 2 Modulation type.
SHAPE:			
DC *n*	1, 0 or -1	4-40	Selects dc level for AF Generator 2 wave shape.
PULse:DCYCLe 50		4-40	Selects Pulse (50% duty cycle) for AF Generator 2.
RAMP		4-40	Selects ramp wave shape for AF Generator 2.
SIN		4-40	Selects sine wave shape for AF Generator 2.
SQU		4-40	Selects square wave shape for AF Generator 2.
TRI		4-40	Selects triangle wave shape for AF Generator 2.
STATe *b*	1 or 0	4-40	Enables/Disables AF Generator 2.

COMMAND	RANGE	PAGE	DESCRIPTION
FGEN:			
GEN3:			
DIGital *type*	DCSINV, POCSAG, DCS, DSAT or DST	4-41	Selects digital encoding type.
ENCODE *type*	DTMF, TONE, DIGital or RCC	4-41	Selects signalling format to encode.
MODL *n*	0.0 to 100.0, (%/AM) 0.0 to 100.0 (kHz/FM/ Tone or RCC), 0.0 to 10.0 (kHz/FM/DTMF), 0.0 to 25.0 (kHz/FM/Digital), 0.0 to 10.0 (radians/PM)	4-41	Sets AF Generator 3 Modulation level.
MODL?		4-41	Returns AF Generator 3 Modulation level setting.
MODulation:*type*	AM, FM, PM or OFF	4-41	Selects AF Generator 3 Modulation type.
MODulation?		4-41	Returns selected AF Generator 3 Modulation type.
MIC:			
LEVel *n*	0 to 100 (%)	4-41	Sets MIC/ACC Connector Proportional output level.
LEVel?		4-41	Returns MIC/ACC Connector Proportional output level setting.
MODL *n*	0.0 to 100.0, (%/AM) 0.0 to 25.0 (kHz/FM), 0.0 to 10.0 (radians/PM)	4-41	Sets MIC/ACC Connector input Modulation level.
MODL?		4-41	Returns set MIC/ACC Connector input Modulation level.
MODulation:	AM, FM, PM or OFF	4-41	Selects MIC/ACC Connector Modulation type.
MODulation?		4-42	Returns selected MIC/ACC Connector Modulation type.
STATe *b*	1 or 0	4-42	Enables/Disables MIC/ACC Connector Modulation.
OUTput:			
AUDio *b*	1 or 0	4-42	Enables/Disables AF Generator output to AUDIO OUT Connector.
AUDio?		4-42	Returns AUDIO OUT Connector routing status.
DEMod *b*	1 or 0	4-42	Enables/Disables AF Generator output to DEMOD OUT Connector.
DEMod?		4-42	Returns DEMOD OUT Connector routing status.
LEVel *v*	0 to 3.1000	4-42	Sets AF Generator output level in volts.
LEVel?		4-43	Returns AF Generator output level setting in volts.
SPEAKer *b*	1 or 0	4-43	Enables/Disables AF Generator output to Test Set Speaker.
SPEAKer?		4-43	Returns Test Set Speaker routing status.
PROPortional *b*	1 or 0	4-43	Enables/Disables Proportional mode for AF Generator.
PROPortional?		4-43	Returns Proportional mode status.
RCL *n*	1 to 9	4-43	Recalls stored AF Generator environment.
STORe *n*	1 to 9	4-43	Stores current AF Generator environment.
PTT:STATe b	1 or 0	4-42	Sets push to talk pin on MIC/ACC Connector.

AF LEVEL METER COMMANDS

M_VRMS?		4-89	Returns voltage RMS reading of received AF level.

AF METER COMMANDS

M_AF:			
ALARM *b*	1 or 0	4-64	Enables/Disables Alarm for Upper or Lower Limit.
FILTer:			
HPASs:			
FREQuency *f*	0.5 to 20.0	4-64	Sets High-Pass Filter cutoff frequency in kHz.
FREQuency?		4-64	Returns High-Pass Filter cutoff frequency setting (kHz).
STATe *b*	1 or 0	4-64	Enables/Disables High-Pass Filter.
STATe?		4-64	Returns High-Pass Filter State.
LPASs:			
FREQuency *f*	0.1 to 30.0	4-64	Selects Low-Pass Filter cutoff frequency in kHz.
FREQuency?		4-64	Returns Low-Pass Filter cutoff frequency setting (kHz).
STATe *b*	1 or 0	4-64	Enables/Disables Low-Pass Filter.
STATe?		4-64	Returns Low-Pass Filter State.
INPut:*type*	XAUDIO, SINAD, DEMOD, POWer or FGEN	4-64	Selects AF Meter Input.

COMMAND	RANGE	PAGE	DESCRIPTION
M_AF:			
LL:			
LEVel *f*	0.0000 to 200.0	4-65	Sets Lower Limit level in kHz.
STATe *b*	1 or 0	4-65	Enables/Disables Lower Limit.
PEAK?		4-65	Returns AF Meter Peak reading in Hz.
PH *b*	1 or 0	4-65	Enables/Disables AF Meter Peak Hold Feature.
RANGe:			
AUTO		4-65	Selects Autorange for AF Meter Range.
UPPer *f*	0.2, 2, 20 or 200	4-65	Sets AF Meter Range in kHz.
RCL *n*	1 to 9	4-65	Recalls stored AF Meter environment.
RESolution *n*	0.1 or 1	4-65	Selects AF Meter resolution in Hz.
STORe *n*	1 to 9	4-65	Stores current AF Meter environment.
UL:			
LEVel *f*	0.0000 to 200.0	4-66	Sets Upper Limit Level in kHz.
STATe *b*	1 or 0	4-66	Enables/Disables Upper Limit.
M_AF?		4-66	Returns AF Meter reading in Hz.

ANALYZER COMMANDS (SEE SPECTRUM ANALYZER COMMANDS.)

AUXILIARY TEST SET COMMANDS

AHIT?		4-3	Returns 1 if an input is waiting from Auxiliary Test Set.
AUX "string"		4-3	Sends commands, as strings, to Auxiliary Test Set.
AUX? "string?"		4-3	Sends queries, as strings, to Auxiliary Test Set.

BIT ERROR RATE METER (BER) COMMANDS

M_BER:			
PATtern:			
FIXED		4-80	Selects Fixed pattern for BER Meter test data.
RANDom		4-80	Selects Random pattern for BER Meter test data.
USER *nn*		4-80	Selects User Defined pattern (8 bit pattern) for BER Meter test data.
POLarity:			
NEGative		4-80	Selects Negative polarity.
POSitive		4-80	Selects Positive polarity.
RATE *n*	75, 150, 300, 600, 1200, 2400, 4800 or 16000	4-81	Sets BER Meter rate in bps.
RATE?		4-81	Returns BER Meter rate setting in bps.
RCL *n*	1 to 9	4-81	Recalls stored BER Meter environment.
SIZE *n*	100 to 100000	4-81	Sets BER Meter block size in bits.
SIZE?		4-81	Returns BER Meter block size setting in bits.
STORe *n*	1 to 9	4-81	Stores current BER Meter environment.
TYPE:xxx	BASEband, DUPlex, GENerator or RECeiver	4-81	Set Bit Error Rate Type.
M_BER?		4-81	Returns number of errors for last pass.

CELLULAR AMPS COMMANDS

NOTE: Queries for received data, return -1 if data is not available or has already been read.

CELL:			
ACTion?		4-90	Returns Global Action value.
BIS?		4-90	Returns Busy-Idle Status bit value.
B_I?		4-90	Returns Busy-Idle bit value.
BOTH		4-90	Selects words from streams A and B, for decoding.
C12?		4-90	Returns C12 bit value.
C13?		4-90	Returns C13 bit value.
CAPTure:			
MIN *"xxx/xxx-xxxx"*	*x*=0 to 9, A to D, a to d, x, X, # or *	4-90	Selects MIN to Capture.
MIN?		4-90	Returns MIN currently used for Capture.
MODE:*xxx*	MIN, ORDer, BOTH or OFF	4-90	Selects Capture Mode.
MODE?		4-90	Returns current Capture Mode.

COMMAND	RANGE	PAGE	DESCRIPTION
CELL:			
CAPTure:			
ORDer:*xxx*	PAGE, ALERT, RELease REORDer, SALERT, SNDAddr, INTERCEPT, MAINTenance, AALERT, DRETRY, AINTERCEPT, AUTREG, AREORDer, POWer, AUDIT, VCDES	4-91	Selects Order to capture.
ORDer?		4-91	Returns current Order used for capture.
CHANnel *n*	1 to 1023	4-91	Selects Cellular Frequency by channel number.
CHANnel?		4-91	Returns currently selected Cellular channel.
CHANPOS1?		4-91	Returns Channel Position 1.
CHANPOS2?		4-91	Returns Channel Position 2.
CHANPOS3?		4-91	Returns Channel Position 3.
CHANPOS4?		4-91	Returns Channel Position 4.
CHANPOS5?		4-91	Returns Channel Position 5.
CHANPOS6?		4-91	Returns Channel Position 6.
CMAC?		4-91	Returns Control Mobile Attenuation Code.
CMAX_1?		4-91	Returns Number of Access Channels minus one.
CPA?		4-91	Returns Combined Paging/Access.
DCC?		4-91	Returns Digital Color Code.
DIGITs?		4-91	Returns Call Address.
DSCC?		4-92	Returns Digital SAT Color Code.
DTX?		4-92	Returns Discontinuous Transmission.
E?		4-92	Returns Extended Address.
EF?		4-92	Returns Expanded Protocol Forward Channel Indicator.
END?		4-92	Returns End Indication.
EP?		4-92	Returns Extended Protocol Capable value.
ESN?		4-92	Returns Electronic Serial Number.
FORMat:			
AMPS		4-92	Sets Channel Format to AMPS.
NT400		4-92	Sets Channel Format to NT400.
FORMat?		4-92	Returns Channel Format setting.

NOTE: CELL:GEN commands are part of option 10.

CELL:			
GEN:			
CHANnel *n*	1 to 1023	4-94	Selects AMPS Reverse Channel.
CHANnel?		4-94	Returns AMPS Reverse Channel setting.
DCC *x*	0 to 3	4-94	Sets Digital Color Code.
DCC?		4-94	Returns current Digital Color Code setting.
DIGITs *"x"*	0 to 9999999999999999	4-95	Sets simulated Called Address.
DIGITs?		4-95	Returns current Called Address setting.
ESN:			
DECimal *x*	0 to 23116113919	4-95	Sets Electronic Serial Number (ESN) in decimal.
DECimal?		4-95	Returns ESN setting in decimal.
FORMat:			
DECimal		4-95	Sets ESN screen format to decimal.
HEX		4-95	Sets ESN screen format to hexadecimal.
OCTal		4-95	Sets ESN screen format to octal.
FORMat?		4-95	Returns current ESN format.
HEXadecimal *x*	0 to FFFFFFFF	4-95	Sets ESN in hexadecimal format.
HEXadecimal?		4-95	Returns current ESN setting in hexadecimal format.
OCTal *x*	0 to 37777777777	4-95	Sets ESN in octal format.
OCTal?		4-95	Returns current ESN setting in octal format.
MIN *"xxx/xxx-xxxx"*	*x*=0 to 9	4-95	Sets simulated MIN.
MIN?		4-95	Returns current MIN setting.
MODL *f*	0.0 to 25.0	4-96	Sets Cellular Deviation Level in kHz.
MODL?		4-96	Returns current Cellular Deviation Level setting in kHz.
RECC:			
CONTinuous		4-96	Sends last order selected continuously to Cell Site.
ONCE		4-96	Sends last order selected to Cell Site one time.
ORDer:			
AUDIT		4-96	Selects Audit as Order to send.
AUTREG		4-96	Selects Autonomous Registration as Order to send.

COMMAND	RANGE	PAGE	DESCRIPTION
CELL:			
GEN:			
RECC:			
ORDer:			
ORIGIN		4-96	Selects Origination as Order to send.
PAGE		4-96	Selects Page Response as Order to send.
ORDer?		4-96	Returns current Order selected to send to Cell Site.
STOP		4-96	Stops transmission of Order sent continuously.
RVC:			
CONTinuous		4-96	Sends last order selected continuously to Cell Site.
ONCE		4-96	Sends last order selected to Cell Site one time.
ORDer:			
AUDIT		4-97	Selects Audit as Order to send.
POWer *x*	0 TO 7	4-97	Selects Power Level *x* as Order to send.
SENDCall		4-97	Selects Send Called Address as Order to send.
SNREQuest		4-97	Selects Autonomous Registration as Order to send.
ORDer?		4-97	Returns current Order selected to send to Cell Site.
STOP		4-97	Stops transmission of Order sent continuously.
SCM:			
BANDwidth *x*	20 or 25	4-97	Selects Mobile Phone Bandwidth in MHz.
BANDwidth?		4-97	Returns current Mobile Phone Bandwidth setting.
POWer *x*	1 to 4	4-97	Selects Power Level of simulated Mobile Phone.
POWer?		4-97	Returns current Power Level setting.
TRANsmit:			
CONTinuous		4-97	Selects Continuous Transmission.
DISCONTinuous		4-97	Selects Discontinuous Transmission.
TRANsmit?		4-97	Returns current transmission setting.
LOCALCTRL1?		4-92	Returns first position Local Control field.
LOCALCTRL2?		4-92	Returns second position Local Control field.
MAXBusy:			
OTHer?		4-92	Returns Max # of busy incidents (Non-Page responses).
PGR?		4-92	Returns Max # of busy incidents (Page responses).
MAXSztr:			
OTHer?		4-93	Returns Max # of seizure tries (Non-Page responses).
PGR?		4-93	Returns Max # of seizure tries (Page responses).
MIN?		4-93	Returns Mobile Identification Number as a string.
MSL		4-93	Returns Message Length.
MST?		4-93	Returns Message Type.
N_1?		4-93	Returns Number of Paging Channels minus one.
NAWC?		4-93	Returns Number of Additional Words Coming.
NEWACC?		4-93	Returns New Access Channel starting point.
OLC?		4-93	Returns Overload Control Class.
ORDer?		4-93	Returns current Order as string.
PDSCC?		4-93	Returns Present Digital SAT Color Code.
PSCC?		4-93	Returns Present SAT Color Code.
RCF?		4-93	Returns Read Control-Filler.
REGH?		4-93	Returns Registration for Home Mobile Stations.
REGID?		4-93	Returns Registration Identification.
REGINCR?		4-94	Returns Registration Increment.
REGR?		4-94	Returns Registration for Roaming Mobile Stations.
S?		4-94	Returns Serial Number bit value.
SCC?		4-94	Returns Supervisory Audio Tone Color Code.
SID?		4-94	Returns System Identification Number.
VCHAN?		4-94	Returns Voice Channel.
VMAC?		4-94	Returns Voice Mobile Attenuation Code.
WFOM?		4-94	Returns Wait For Overhead Message bit value.
WORD?		4-94	Returns word selection (WORDA, WORDB or BOTH).
WORDA		4-94	Selects Stream A words for decoding.
WORDB		4-94	Selects Stream B words for decoding.

COMMAND	RANGE	PAGE	DESCRIPTION
DEFINE COMMANDS			
*DMC "*name, command*; ...		3-48	Defines a macro.
CONST *name,expression*		3-54	Declares and defines a constant.
DATA *name={expression,...}*		3-55	Declares and defines a data array.
FORMat *X*	BINary, OCTal, HEXadecimal or ASCii	3-64	Sets data format.
STRING *name,...,name*		3-84	Declares string variables.
STRING *name[n],...,name[n]*		3-84	Declares string arrays with index [n] ([n] mandatory, not optional).
VAR *name,...,name*		3-86	Declares variables.
VAR *name[n],...,name[n]*		3-86	Declares arrays with index [n] ([n] mandatory, not optional).
DEVIATION METER (Peak) COMMANDS			
M_DEV:			
ALARM *b*	1 or 0	4-71	Enables/Disables Alarm for Upper or Lower Limit.
AVErage *b*	1 or 0	4-71	Enables/Disables Averaging.
LL:			
LEVel *f*	0.00 to 100.00	4-71	Sets Lower Limit Level (in 0.05 kHz steps).
STATe *b*	1 or 0	4-71	Enables/Disable Lower Limit.
MODE:			
BOTH		4-71	Selects Both Mode (positive and negative).
NEGative		4-71	Selects Negative Mode.
NORMalize		4-71	Selects Normalized Mode.
POSitive		4-71	Selects Positive Mode.
NEG?		4-71	Returns Negative Deviation in kHz.
PEAK:			
NEG?		4-72	Returns Negative Peak Deviation in kHz.
POS?		4-72	Returns Positive Peak Deviation kHz.
PH *b*	1 or 0	4-72	Enables/Disables Peak Hold Feature.
POS?		4-72	Returns Positive Deviation in kHz.
RANGe:			
AUTO		4-72	Selects Autorange for Deviation Meter Range.
UPPer *f*	2, 5, 10, 20, 50 or 100	4-72	Selects Deviation Meter Range in kHz.
RCL *n*	1 to 9	4-72	Recalls stored Deviation Meter (Peak) environment.
STORe *n*	1 to 9	4-72	Stores current Deviation Meter (Peak) environment.
UL:			
LEVel *f*	0.00 to 100.00	4-73	Sets Upper Limit Level (in 0.05 kHz steps).
STATe *b*	1 or 0	4-73	Enables/Disables Upper Limit.
DEVIATION METER (RMS) COMMANDS			
M_DRMS:			
ALARM *b*	1 or 0	4-86	Enables/Disables Alarm for Upper or Lower Limit.
AVErage *b*	1 or 0	4-86	Enables/Disables Averaging.
LL:			
LEVel *f*	0.00 to 10.00	4-86	Sets Lower Limit Level in kHz.
STATe *b*	1 or 0	87	Enables/Disables Lower Limit.
PH *b*	1 or 0	87	Enables/Disables Peak Hold Feature.
RANGe:			
AUTO		87	Selects Autorange for Meter Range.
UPPer *n*	2, 5 or 10	87	Sets Meter Range in kHz.
RCL *n*	1 to 9	4-86	Recalls stored Deviation Meter (RMS) environment.
STORe *n*	1 to 9	4-86	Stores current Deviation Meter (RMS) environment.
UL:			
LEVel *f*	0.00 to 10.00	4-87	Sets Upper Limit Level in kHz.
STATe *b*	1 or 0	4-87	Enables/Disables Upper Limit.
M_DRMS?		4-87	Returns Deviation Meter (RMS) in kHz.
DIGITAL MULTIMETER (DMM) COMMANDS			
M_DMM:			
ALARM *b*	1 or 0	4-82	Enables/Disables Alarm for Upper or Lower Limit.

COMMAND	RANGE	PAGE	DESCRIPTION
M_DMM:			
FUNCtion:			
CURRent:			
AC		4-82	Selects ac Current Meter as DMM function.
DC		4-82	Selects dc Current Meter as DMM function.
RESistance		4-82	Selects Ohmmeter as DMM function.
VOLTage:			
AC		4-82	Selects ac Voltmeter as DMM function.
DC		4-82	Selects dc Voltmeter as DMM function.
FUNCtion?		4-82	Returns current DMM function.
INPut:IMPedance *n*	150, 600 or 1e6	4-82	Sets ac Voltmeter Load in ohms.
LL:			
LEVel *n*		4-82	Sets Lower Limit for current function.
STATe *b*	1 or 0	4-82	Enables/Disables Lower Limit for current function.
PH *b*	1 or 0	4-83	Enables/Disables Peak Hold Feature.
RANGe:			
AUTO		4-83	Selects Autorange for Meter Range of current function.
UPPer *n*	0.2, 2, 20, 200 or 2000 (VAC or Vdc); 0.2, 2, 20, 200, 2000 or 20000 (Ω); 0.02, 0.2, 2 or 20 (A)	4-83	Selects Meter Range for current function.
RCL *n*	1 to 9	4-83	Recalls stored Digital Multimeter environment.
STORe *n*	1 to 9	4-83	Stores current Digital Multimeter environment.
UL:			
LEVel *n*		4-83	Sets Upper Limit for current function.
STATe *b*	1 or 0	4-83	Enables/Disables Upper Limit of current function.
M_DMM?		4-84	Returns DMM reading in amps, volts or ohms.

DISPLAY COMMANDS

BCOLOR *b*	0 to 15 or name	3-25	Sets background color.
BOX *f, x1, y1, x2, y2, c*	1 or 0, 0 to 639, 0 to 349, 0 to 639, 0 to 349, 0 to 15	3-51	Creates box on CRT display.
CLS		3-54	Clears display screen or selected window of everything except active readings.
COLOR *f,c*	0 to 15 or name, 0 to 15 or name	3-25	Sets foreground and character contrast colors.
COLOR?		3-25	Returns current foreground color.
DRAW *x1, y1, x2, y2, c*	0 to 639, 0 to 349, 0 to 639, 0 to 349, 0 to 15	3-58	Creates line on CRT display.
EDIT:			
COLOR:			
LETTER *c*	0 to 15 or name	3-25	Sets menu letter color.
LETTER?		3-25	Sets menu letter color setting.
MENU *c*	0 to 15 or name	3-25	Sets menu color.
MENU?		3-25	Sets menu color setting.
SOFT:			
BOX *c*	0 to 15 or name	3-26	Sets Soft Function Key background color.
BOX?		3-26	Returns Soft Function Key background color setting.
LETTER *c*	0 to 15 or name	3-26	Sets Soft Function Key letter color.
LETTER?		3-26	Returns Soft Function Key letter color setting.
SELECT *c*	0 to 15 or name	3-26	Sets Soft Function Key selected color.
SELECT?		3-26	Returns Soft Function Key selected color setting.
WIDTH *n*	0 to 639	3-26	Sets width of blinking cursor for **INPUT** command.
ELLIPSE *b,x,y,r,a,c*	0 or 1, 0 to 639, 0 to 349, (*r&a* vary), 0 to 15	3-30	Creates an ellipse on CRT display.
HEIGHT *n*	1 to 4	3-64	Sets height of text from 0.175 to 0.7 inches.
ICON *b,r,name*	0 to 639, 0 to 349	3-30	Creates bit pattern graphic on CRT display.
PIXEL		3-77	Displays point at current xy position on CRT.
PPRINT *%0nl, expression, ...*		3-78	Prints expression out 1600S RS-232 Connector.
PRINT *%0nl, expression,\ expression, ... ,expression,*		3-79	Prints expression at current xy position on CRT.
PSCREEN		3-80	Prints CRT screen out 1600S RS-232 Connector.
ROTATE *n*	0,90,180 or 270	3-82	Rotates text character by character in degrees.
SCREEN:USER		3-26	Clears display screen or active window of everything.
XY *n, m*	0 to 639, 0 to 349	3-90	Sets xy position for other display commands.

COMMAND	RANGE	PAGE	DESCRIPTION
DISTORTION METER COMMANDS			
M_DIST:			
ALARM *b*	1 or 0	4-76	Enables/Disables Alarm for Upper or Lower Limit.
AVErage *b*	1 or 0	4-76	Enables/Disables Averaging.
FILTer *f*	600 to 1400	4-76	Sets Notch Filter Frequency in Hz.
INPut:*type*	DEMOD, SINAD, XAUDIO or FGEN	4-76	Selects Distortion Meter Input.
LL:			
LEVel *n*	0.0 to 20.0	4-76	Sets Lower Limit Level in percent.
STATe *b*	1 or 0	4-76	Enables/Disables Lower Limit.
PEAK?		4-76	Returns Distortion Meter Peak reading in percent.
PH *b*	1 or 0	4-76	Enables/Disables Peak Hold Feature.
RCL *n*	1 to 9	4-76	Recalls stored Distortion Meter environment.
SELect:			
CWEight		4-77	Selects C-Weight Filter.
LPASs *f*	100 to 30000	4-77	Selects Low-Pass Filter with cutoff frequency in Hz.
STORe *n*	1 to 9	4-77	Stores current Distortion Meter environment.
UL:			
LEVel *n*	0.0 to 20.0	4-77	Sets Upper Limit Level in percent.
STATe *b*	1 or 0	4-77	Enables/Disables Upper Limit.
M_DIST?		4-77	Returns Distortion Meter reading.
DUPLEX COMMANDS			
DUPlex:			
INPut:			
AGC:			
AUTO		4-29	Sets AGC to automatic setting.
MANual *n*	0 to 255	4-29	Sets AGC to manual setting.
USER:*xxx*	MEASure, SPeech, DATA, HIGH, TYPE1 TYPE2 or TYPE3.	4-29	Sets AGC to User setting.
ANTenna		4-30	Selects ANTENNA IN Connector for Duplex Tx input.
ATTenuation *n*	0, 20 or 40	4-30	Sets Duplex Input Attenuation in dB.
CHANnel *n*	1 to 1023	4-30	Selects Duplex Transmitter channel.
CHANnel:			
FORMat:			
AMPS:			
FORward		4-30	Selects AMPS Forward channels for Channel Mode.
REVerse		4-30	Selects AMPS Reverse channels for Channel Mode.
ETACS:			
FORward		4-30	Selects ETACS Forward channels for Channel Mode.
REVerse		4-30	Selects ETACS Reverse channels for Channel Mode.
NAMPS:			
FORward		4-30	Selects NAMPS Forward channels for Channel Mode.
REVerse		4-30	Selects NAMPS Reverse channels for Channel Mode.
NT400:			
FORward		4-30	Selects NT400 Forward channels for Channel Mode.
REVerse		4-30	Selects NT400 Reverse channels for Channel Mode.
FORMat?		4-31	Returns Duplex Transmitter channel format setting.
FIND:			
FREQuency?		4-31	Returns first frequency with amplitude > reference.
REFerence *n*		4-31	Sets Find frequency reference level in dB.
REFerence?		4-31	Returns Find frequency reference level setting in dB.
FREQuency *f* [*units*]	250.0 to 999999.9 (kHz); HZ, KHZ or MHZ	4-31	Sets Duplex Transmitter Frequency in kHz or selected optional units.
FREQuency?		4-31	Returns Duplex Transmitter frequency setting in kHz.
METER:			
DEVRms		4-31	Displays Deviation Meter (RMS)/Duplex Xmtr Opn Screen.
DISTortion		4-31	Displays Distortion Meter/Duplex Xmtr Opn Screen.
MODMeter		4-31	Displays Modulation Meter/Duplex Xmtr Opn Screen.
PMRms		4-31	Displays Phase Meter (RMS)/Duplex Xmtr Opn Screen.
SINAD		4-31	Displays SINAD Meter/Duplex Xmtr Opn Screen.
MODE *xxx*	DIRect or CHANnel	4-32	Selects Duplex Transmitter Mode.

COMMAND	RANGE	PAGE	DESCRIPTION
DUPlex:			
INPut:			
MODulation:			
AM*n*	1 or 2	4-32	Selects an Amplitude Modulation.
BFO		4-32	Selects Beat Frequency Oscillator Modulation.
FM*n*	1, 2, 3 or 4	4-32	Selects a Frequency Modulation.
LSB		4-32	Selects Lower Side Band Modulation.
PM		4-32	Selects Phase Modulation.
USB		4-32	Selects Upper Side Band Modulation.
USER:			
FILTer *f*	3, 30 or 300	4-32	Selects User Defined IF Filter in kHz.
MODulation:*type*	FM, AM, USB, LSB, BFO, PM or DATA	4-32	Selects User Defined Modulation.
POST:			
APASs		4-32	Selects All Pass Post Detection Filter.
BPASs *fl,fh*	0.5 to 20; 0.1 to 30	4-32	Selects Bandpass Post Detection Filter with low and high cutoff frequencies in kHz.
CWEight		4-32	Selects C-Weighted Post Detection Filter.
HPASs *f*	0.5 to 20	4-33	Selects High-Pass Post Detection Filter with cutoff frequency in kHz.
LPASs *f*	0.1 to 30	4-33	Selects Low-Pass Post Detection Filter with cutoff frequency in kHz.
MODulation?		4-33	Returns current Duplex Transmitter Modulation.
TO:			
AUDIO *b*	1 or 0	4-33	Sets demodulated routing to AUDIO OUT Connector.
DEMOD *b*	1 or 0	4-33	Sets demodulated routing to DEMOD OUT Connector.
SPEAKer *b*	1 or 0	4-33	Sets demodulated routing to Test Set Speaker.
TR		4-33	Selects T/R Connector as Duplex Transmitter Input.
VOLume:			
AUTO *b*	1 or 0	4-33	Enables/Disables Automatic Volume Control.
AUTO?		4-33	Returns Automatic Volume Control status.
METER:			
DISTortion		4-29	Displays Distortion Meter/Duplex Operation Screen.
MODMeter		4-29	Displays Modulation Meter/Duplex Operation Screen.
OFF		4-29	Disables Modulation, Distortion and SINAD Meters.
SINAD		4-29	Displays SINAD Meter/Duplex Operation Screen.
OUTput:			
AUDio *b*	1 or 0	4-34	Sets AF Generator Output to AUDIO OUT Connector.
CHANnel *n*	1 to 1023	4-34	Selects Duplex Receiver channel.
CHANnel:			
FORMat:			
AMPS:			
FORward		4-34	Selects AMPS Forward channels for Channel Mode.
REVerse		4-34	Selects AMPS Reverse channels for Channel Mode.
ETACS:			
FORward		4-34	Selects ETACS Forward channels for Channel Mode.
REVerse		4-34	Selects ETACS Reverse channels for Channel Mode.
NAMPS:			
FORward		4-34	Selects NAMPS Forward channels for Channel Mode.
REVerse		4-34	Selects NAMPS Reverse channels for Channel Mode.
NT400:			
FORward		4-34	Selects NT400 Forward channels for Channel Mode.
REVerse		4-34	Selects NT400 Reverse channels for Channel Mode.
FORMat?		4-35	Returns Duplex Receiver channel format setting.
DEMOD *b*	1 or 0	4-35	Sets AF Generator Output to DEMOD OUT Connector.
DUPlex		4-35	Selects DUPLEX OUT Connector as Duplex Receiver Output.
FREQuency *f* [*units*]	250.0 to 999999.9 (kHz); HZ, KHZ or MHZ	4-35	Sets Duplex Receiver Frequency in kHz or selected optional units.
FREQuency?		4-35	Returns Duplex Receiver frequency setting in kHz.
LEVel:			
DBm *n*	-137.0 to 0.0	4-35	Sets Duplex RF Level in dBm.
DBm?		4-35	Returns Duplex RF Level setting in dBm.

COMMAND	RANGE	PAGE	DESCRIPTION
DUPlex:			
OUTput:			
METER:			
AF		4-36	Displays AF Meter/Duplex Receiver Operation Screen.
DISTortion		4-36	Displays Distortion Meter/Duplex Rcvr Opn Screen.
DMM		4-36	Displays Digital Multimeter/Duplex Rcvr Opn Screen.
SINAD		4-36	Displays SINAD Meter/Duplex Rcvr Opn Screen.
MODE xxx	DIRect or CHANnel	4-36	Selects Duplex Receiver Mode.
OFFSet *f*	-999749.9 to 999749.9	4-36	Sets Offset Frequency in kHz.
OFFSet?		4-36	Returns Offset Frequency setting in kHz.
TR		4-36	Selects T/R Connector as Duplex Receiver Output.
RCL *n*	1 to 9	4-29	Recalls stored Duplex environment.
SPEAKer:			
SOURce *type*	OFF, FGEN, SINAD or EXT MOD	4-29	Routes selected connector or source to Speaker.
STORe *n*	1 to 9	4-29	Stores current Duplex environment.

FETCH COMMANDS (SEE INITIATE AND FETCH COMMANDS.)

FREQUENCY ERROR METER COMMANDS

M_RF:			
ALARM *b*	1 or 0	4-67	Enables/Disables Alarm for Upper or Lower Limit.
LL:			
LEVel *f*	0.0000 to 100.000	4-67	Sets Lower Limit Level in kHz.
STATe *b*	1 or 0	4-67	Enables/Disables Lower Limit.
PEAK?		4-67	Returns Frequency Error Meter Peak reading in Hz.
PH *b*	1 or 0	4-67	Enables/Disables Peak Hold Feature.
RANGe:			
AUTO		4-67	Selects Autorange for Meter Range.
UPPer *f*	0.1, 1, 10 or 100	4-67	Sets Meter Range in kHz.
RCL *n*	1 to 9	4-67	Recalls stored Frequency Error Meter environment.
RESolution *f*	1 or 10	4-67	Sets Meter Resolution in Hz.
STORe *n*	1 to 9	4-67	Stores current Frequency Error Meter environment.
UL:			
LEVel *f*	0.0000 to 100.000	4-68	Sets Upper Limit Level in kHz.
STATe *b*	1 or 0	4-68	Enables/Disables Upper Limit.
M_RF?		4-68	Returns RF Frequency Error Meter reading in Hz.

FUNCTION GENERATOR COMMANDS (SEE AF GENERATOR COMMANDS.)

GENERATOR COMMANDS (SEE RF GENERATOR COMMANDS.)

GENERIC MEASURE COMMANDS

MEASure:			
AUDio? [*e*,*r*]		4-103	Returns AF Meter reading in Hz.
CURRent:			
AC? [*e*]		4-103	Returns DMM ac Current reading in amps.
DC? [*e*]		4-103	Returns DMM dc Current reading in amps.
FREQuency? [*e*,*r*]		4-103	Returns signal frequency in kHz.
MIC?		4-103	Returns state of MIC/ACC Connector Input.
PHASe? [*e*]		4-103	Returns Phase Meter reading in radians.
POWer? [*e*]		4-103	Returns Power Meter readings in mW.
RESistance? [*e*]		4-103	Returns DMM Resistance readings in kW.
SINAD? *r*		4-103	Returns SINAD Meter reading in dB.
SQUelch?		4-103	Returns Squelch state (1 if broken).
TEMPerature:			
AMBient?		4-103	Returns Ambient temperature in degrees Celsius.
POWer?		4-103	Returns Power Term temperature in degrees Celsius.
VOLTage:			
AC? [*e*]		4-103	Returns DMM ac voltage reading in volts.
DC? [*e*]		4-103	Returns DMM dc voltage reading in volts.
SUPply? [*n*]	-15, 5 or 15	4-104	Returns selected Power Supply voltage in volts.

COMMAND	RANGE	PAGE	DESCRIPTION
IEEE 488.2 COMPLIANCE COMMANDS			
*CLS		3-46	Clears all Status Registers to zero.
*DDT		3-46	Defines Device Trigger.
*DMC "*name, command*; ...		3-48	Defines a macro.
*EMC *b*	1 or 0	3-46	Enables/disables macros.
*EMC?		3-46	Returns enable/disable macros status.
*ESE *n*	0 to 255	3-46	Sets Standard Event Status enable register.
*ESE?		3-46	Returns Standard Event Status enable register value.
*ESR?		3-46	Returns Standard Event Status Register value.
*IDN?		3-49	Returns FM/AM-1600S identification.
*LMC?		3-46	Returns list of currently defined macros.
*OPC		3-46	Activates bit 0 of Standard Event Status Register when operation finishes.
*OPC?		3-46	Activates bit 4 of Status Byte when operation finishes.
*PMC		3-49	Deletes all macros (except predefined) and declared variables from FM/AM-1600S.
*RCL *n*	1 to 9	3-47	Recalls stored FM/AM-1600S environment.
*RST		3-47	Returns FM/AM-1600S to default settings.
*SAV *n*	1 to 9	3-47	Stores current FM/AM-1600S environment.
*SRE *n*	0 to 255	3-47	Sets Service Request enable register.
*SRE?		3-47	Returns Service Request enable register value.
*STB?		3-47	Returns Status Byte value.
*TRG		3-47	Executes trigger command as defined by ***DDT**.
*TST?		3-47	Performs Self Test/Returns 0 (passed) or 1 (failed).
*WAI		3-49	Pauses command execution to complete operations.
INITIATE and FETCH COMMANDS			
FETCh:			
AF?		4-104	Returns Audio Frequency Meter reading in kHz.
DMM?		4-104	Returns Digital Multimeter reading in A, V or kΩ.
RF?		4-104	Returns Frequency Error Meter reading in kHz.
INITiate:			
AF		4-104	Prepares AF Meter for **FETch** reading.
DMM		4-104	Prepares DMM for **FETch** reading.
RF		4-104	Prepares Frequency Error Meter for **FETch** reading.
KEY COMMANDS			
KEY		3-32	Returns keycode of next Front Panel Key pressed.
KEY?		3-32	Returns 1 if any Front Panel Key is pressed or 0.
KEYPAD COMMANDS			
KEYPAD:			
CLAIM		3-32	Sends Front Panel Keyboard input to TMAC Interpreter.
ERASE *n*	1 to 6	3-73	Erases current Soft Function Key F*n* definition.
LABel *n*, "*Label*"	1 to 6	3-74	Creates *Label* for Soft Function Key F*n*.
SOFT		3-75	Creates Soft Function Key frame in current screen.
UNCLAIM		3-32	Releases Front Panel Keyboard for normal use.
MACRO FUNCTION COMMANDS			
BEGIN		3-50	Allows macro to have multiple lines.
CASE variable		3-52	Provides predetermined decision point.
OF *value1: command*			
OF *value2: command*			
OTHERWISE: *command*			
DELAY *t*		3-56	Provides time delay in ms.
DO *sequence* UNTIL *condition*		3-56	Performs set of commands.
END		3-59	Ends macro.
ENDCASE		3-52	Ends Case command.
EXEC *address* or EXEC &*name*		3-59	Executes a macro.
FLUSH		3-60	Returns pending responses.

COMMAND	RANGE	PAGE	DESCRIPTION
FOR *variable=initial* TO *ending* STEP *step sequence*		3-61	Performs set of commands repeatedly.
NEXT *variable*			
FORGET "*name*"		3-63	Deletes macro called "*name*" from memory.
IF *condition sequence*; ENDIF		3-68	Provides decision point.
IF *condition sequence*; ELIF *condition sequence*		3-69	Provides decision points.
ELIF *condition sequence*;ENDIF			
IF *condition sequence*		3-70	Provides decision point.
ELSE *condition sequence*;ENDIF			
INPUT *variable*		3-72	Allows data entry through RS-232 Connector during macro execution.
INTERP "*string;string;...;string*"		3-72	Executes commands, unknown when macro is defined.
RETURN		3-81	Returns value from function or returns macro execution to previous macro.
ROOM		3-82	Used with **PPRINT** to return memory bytes available.
WHILE *condition sequence*;WEND		3-87	Performs set of commands repeatedly until conditional expression is false.

MATHEMATICAL FUNCTION COMMANDS

ABS(n)	3-13	Produces absolute value of n.
CALCULATE? *expression*	3-52	Calculates expression functions and returns result.
COS(n)	3-14	Produces cosine of n in radians.
EXP(n)	3-14	Produces e to the power of n.
FALSE,OFF	3-12	Produces 0.
FLOOR(n)	3-13	Produces truncated value of n.
LN(n)	3-13	Produces natural logarithm (base e) of n.
LOG(n)	3-13	Produces base 10 logarithm of n.
RAND x	3-12	Sets seed (starting point) to x for random generator.
RND(n)	3-12	Produces random integer between 0 and n.
SIGN(n)	3-14	Produces -1 if n<0, 0 if n=0 or 1 if n>0.
SIN(n)	3-14	Produces sine of n in radians.
SQR(n)	3-14	Produces positive square root of n.
TRUE,ON	3-12	Produces 1.

MASS MEMORY COMMANDS

MMEMory:			
CATalog:			
ENTRY? n	0 to 512	4-101	Returns line n from Flash Memory File Directory.
FREE?		4-101	Returns available file space in bytes.
USED?		4-101	Returns used file space in bytes.
CATalog?		4-101	Returns Flash Memory status.
DELete "f"		4-101	Deletes file with file name f.
INITialize		4-101	Erases all files stored in Flash Memory.
INITialize?		4-101	Returns 1 for file system initialized or 0 otherwise.
LOAD:			
CALibration "f"		4-101	Loads Calibration Data from Flash to 1600S Memory.
DATA v,"f"		4-101	Loads variable stored as file name f, from Flash to 1600S Memory as v.
MACRo "m","f"		4-101	Loads macros and variables stored as file name f from Flash to 1600S memory, executing macro m.
STATe n,"f"	0 to 9	4-101	Loads 1600S State stored as f, into Auxiliary Functions "Store Parameters Menu" as entry n.
TRACe:			
ANLZ n,"f"	0 to 9	4-101	Loads Spectrum Analyzer trace stored as f, into Spectrum Analyzer "Store Parameters Menu" as entry n.
SCOPe n,"f"	0 to 9	4-101	Loads Oscilloscope trace stored as f, into Oscilloscope "Store Parameters Menu" as entry n.
PACK		4-101	Frees memory space from deleted files.
STORe:			
CALibration "f"		4-102	Stores current Calibration Data as f in Flash Memory.
DATA v,"f"		4-102	Stores variable v in Flash Memory as file name f.
MACRo "m","f"		4-101	Stores currently loaded macros and variables as file name f in Flash Memory with designated macro m to activate from Flash Memory File Directory.

COMMAND	RANGE	PAGE	DESCRIPTION
MMEMory:			
STORe:			
STATe n,"f"	0 to 9	4-102	Stores entry n of Auxiliary Functions "Store Parameters Menu" as f in Flash Memory.
TRACe:			
ANLZ n,"f"	0 to 9	4-102	Stores entry n of Spectrum Analyzer "Store Parameters Menu" as f in Flash Memory.
SCOPe n,"f"	0 to 9	4-102	Stores entry n of Oscilloscope "Store Parameters Menu" as f in Flash Memory.
TYPE? "f"		4-102	Returns file type of file name f.

MODULATION METER (AM) COMMANDS

M_MOD:			
ALARM *b*	1 or 0	4-74	Enables/Disables Alarm for Upper or Lower Limit.
LL:			
LEVel *n*	0.0 to 100.0	4-74	Sets Lower Limit Level in percent.
STATe *b*	1 or 0	4-74	Enables/Disables Lower Limit.
PEAK?		4-74	Returns Modulation Meter Peak reading in percent.
PH *b*	1 or 0	4-74	Enables/Disables Peak Hold Feature.
RANGe:			
AUTO		4-74	Selects Autorange for Meter Range.
UPPer *n*	40 or 100	4-74	Selects Meter Range in percent.
RCL *n*	1 to 9	4-74	Recalls stored Modulation Meter environment.
STORe *n*	1 to 9	4-74	Stores current Modulation Meter environment.
UL:			
LEVel *n*	0.0 to 100.0	4-75	Sets Upper Limit Level in percent.
STATe *b*	1 or 0	4-75	Enables/Disables Upper Limit.
M_MOD?		4-75	Returns Modulation Meter reading in percent.

MULTITASKING MACROS COMMANDS

ACTIVATE "*name*"	3-50	Puts specified task in schedule queue.
SLEEP "*name*"	3-83	Removes specified task from schedule queue.
STOP	3-84	Stops all macros.
TASK "*name*"	3-85	Declares specified loaded macro to be a task.
TPAUSE	3-85	Passes command execution to next task in queue.
TSTOP	3-85	Stops and removes current task from queue.
WAKE "*name*"	3-86	Puts specified task in queue/executes from last stop.

NON-VOLATILE MEMORY COMMANDS

NVRCL *name,location*	3-76	Recalls variable or array store in NVRAM.
NVSAV *name,location*	3-76	Stores variable or array in NVRAM.

OSCILLOSCOPE COMMANDS

SCOPe:			
ARM		4-45	Arms Oscilloscope if in One shot Mode.
AVErage *n*	1 to 100	4-45	Selects Average Mode and number of samples.
COMPare *n*	1 to 9	4-45	Selects Compare Mode with stored trace *n*.
COUPling *type*	AC, DC or GROund	4-45	Selects external coupling.
FULL		4-45	Selects full size Oscilloscope display for other Operation Screens.
HORIZontal *n*	-12 to 12	4-45	Sets Horizontal Time Offset in major divisions.
INPut:			
FILTer:			
CWEight:			
STATe *b*	1 or 0	4-45	Selects C-Weight Filter as Input Filter.
HPASs:			
FREQuency *f*	0.2 to 100	4-45	Sets High-Pass cutoff frequency in kHz.
STATe *b*	1 or 0	4-45	Selects High-Pass Filter as Input Filter.
LPASs:			
FREQuency *f*	0.2 to 50	4-45	Sets Low-Pass cutoff frequency in kHz.
STATe *b*	1 or 0	4-45	Selects Low-Pass Filter as Input Filter.

COMMAND	RANGE	PAGE	DESCRIPTION
SCOPe:			
INPut:			
FILTer:			
NOTch:			
FREQuency	0.5 to 1.5	4-46	Sets Notch Filter width in kHz.
STATE	1 or 0	4-46	Selects Notch Filter as Input Filter.
INTernal *type*	IF, DEMOD, POWer, SINAD, FUNCtion or XAUDIO	4-46	Selects Internal Oscilloscope Input.
LEVel *n*	0 to 255	4-46	Sets Trigger Level.
LIVe [*b*]	1 or 0	4-46	Selects Live Trace Mode performing **SCREEN:SCOPE** if *b* is 1. Default *b* is 1.
MARKer:			
AOFF		4-46	Disable both markers.
DELTA:			
AMPLitude?		4-46	Returns amplitude difference between Trace Marker crossings in volts.
POINt?		4-46	Returns Marker position difference in graticules.
TIME?		4-46	Returns difference in Marker time positions in ms.
TRACK *b*	1 or 0	4-46	Enables/Disables marker tracking.
MARKer1:			
AMPLitude?		4-47	Returns Vertical value at Marker 1 in volts.
POINt *n*	1 to 100	4-47	Sets Marker 1 position in graticules.
POINt?		4-47	Returns Marker 1 position in graticules.
STATe *b*	1 or 0	4-47	Enables/Disables Marker 1.
STATe?		4-47	Returns Marker 1 status.
TIME?		4-47	Returns Marker 1 position in ms.
MARKer2:			
AMPLitude?		4-47	Returns Vertical value at Marker 2 in volts.
POINt *n*	1 to 100	4-47	Sets Marker 2 position in graticules.
POINt?		4-47	Returns Marker 2 position in graticules.
STATe *b*	1 or 0	4-47	Enables/Disables Marker 2.
STATe?		4-47	Returns Marker 2 status.
TIME?		4-47	Returns Marker 2 position in ms.
QTR		4-48	Selects 1/4 size Oscilloscope display for other Operation Screens.
RCL *n*	1 to 9	4-48	Recalls stored Oscilloscope Trace and environment.
SCALe *n*	1 to 50000 (1-2-5 sequence);	4-48	Sets Scale in mV/div for AC, DC or GND Input.
	2, 4, 10 or 20;		Sets Scale in kHz/div for FM Demod Audio Input.
	500, 1000 or 2500		Sets Scale in mV/div for Func Gen or Ext Mod Input.
SCALe?		4-48	Returns Scope Scale.
SOURce *type*	EXTernal or INTernal	4-48	Selects Input type.
STATe *b*	1 or 0	4-48	Enables/Disables Scope display in other Operation Screens.
STORe *n*	1 to 9	4-48	Stores current Oscilloscope Trace and environment.
SWEep *t*	1 to 100000 (1-2-5 sequence)	4-49	Sets Oscilloscope Sweep Rate in µs.
SWEep?		4-49	Returns Oscilloscope Sweep Rate in µs.
TRACE:			
DATA *n,offset,data*	1 to 9;0 to 399; 0 to 255	4-49	Sets points into trace beginning at offset.
DATA? *n,offset,points*	0 to 9;0 to 399;1 to 400	4-49	Sends trace data points directly to host.
GET *name,n*	array name;0 to 9	4-49	Assigns values of trace *n* to named array.
GET? *n,offset*	0 to 9;0 to 399	4-49	Returns value of trace 0 to 9 at offset.
MAX? *n,offset,points*	0 to 9;0 to 399;1 to 400	4-50	Returns maximum vertical value. Returns x,y position.
MIN? *n,offset,points*	0 to 9;0 to 399;1 to 400	4-50	Returns minimum vertical value. Returns x,y position.
PUT *name,n*	array name; 1 to 9	4-50	Assigns values of an array to a stored trace.
TRIGger:			
AUTO		4-51	Selects Auto Sweep for Trigger Mode.
IMMediate		4-51	Triggers Oscilloscope after command interpretation.
NORM		4-51	Selects Normal Sweep for Trigger Mode.
ONE		4-51	Selects One Shot for Trigger Mode.
SOURce *type*	EXTernal, INTernal or BUS	4-51	Selects source to trigger on. External or Internal (AC or DC Input only). BUS is ***TRG** or GET command.
VERTical *n*	0 to 255	4-51	Sets Vertical Offset in graticules.

COMMAND	RANGE	PAGE	DESCRIPTION
PHASE METER COMMANDS			
M_PM:			
ALARM *b*	1 or 0	4-84	Enables/Disables Alarm for Upper or Lower Limit.
LL:			
LEVel *n*	0.00 to 10.00	4-84	Sets Lower Limit Level in radians.
STATe *b*	1 or 0	4-84	Enables/Disables Lower Limit.
PH *b*	1 or 0	4-84	Enables/Disables Peak Hold Feature.
RANGe:			
AUTO		4-84	Selects Autorange for Meter Range.
UPPer *n*	1, 5 or 10	4-84	Sets Meter Range in radians.
RCL *n*	1 to 9	4-84	Recalls stored Phase Meter environment.
STORe *n*	1 to 9	4-85	Stores current Phase Meter environment.
UL:			
LEVel *n*	0.00 to 10.00	4-85	Sets Upper Limit Level in radians.
STATe *b*	1 or 0	4-85	Enables/Disables Upper Limit.
M_PM?		4-85	Returns Phase Meter reading in radians.
PHASE METER (RMS) COMMANDS			
M_PMRMS:			
ALARM *b*	1 or 0	4-88	Enables/Disables Alarm for Upper or Lower Limit.
LL:			
LEVel *n*	0.00 to 10.00	4-88	Sets Lower Limit Level in radians.
STATe *b*	1 or 0	4-88	Enables/Disables Lower Limit.
PH *b*	1 or 0	4-88	Enables/Disables Peak Hold Feature.
RANGe:			
AUTO		4-88	Selects Autorange for Meter Range.
UPPer *n*	1, 5 or 10	4-88	Sets Meter Range in radians.
RCL *n*	1 to 9	4-88	Recalls stored Phase Meter (RMS) environment.
STORe *n*	1 to 9	4-88	Stores current Phase Meter (RMS) environment.
UL:			
LEVel *n*	0.00 to 10.00	4-88	Sets Upper Limit Level in radians.
STATe *b*	1 or 0	4-88	Enables/Disables Upper Limit.
M_PMRMS?		4-88	Returns Phase Meter (RMS) reading in radians.
POWER METER COMMANDS			
M_PWR:			
ALARM *b*	1 or 0	4-69	Enables/Disables Alarm for Upper or Lower Limit.
EXT:			
STATe *b*	1 or 0	4-69	Enables/Disables External Loss/Gain Offset.
STATe?		4-69	Returns External Loss/Gain Offset state.
OFFSet *n*	-99.9 to 99.9	4-69	Sets External Loss/Gain Offset in dBm.
OFFSet?		4-69	Returns External Loss/Gain Offset value in dBm.
LL:			
LEVel *n*	0.0000 to 200.0000	4-69	Sets Lower Limit Level in Watts.
STATe *b*	1 or 0	4-69	Enables/Disables Lower Limit.
PEAK?		4-69	Returns Power Meter Peak reading in mW.
PH *b*	1 or 0	4-69	Enables/Disables Peak Hold Feature.
RANGe:			
AUTO		4-69	Selects Autorange for Meter Range.
UPPer *n*	.02, .05, .1, .2, .5, 1, 2, 5, 10, 20, 50, 100 or 200	4-69	Selects Meter Range value in Watts.
RCL *n*	1 to 9	4-70	Recalls stored Power Meter environment.
STORe *n*	1 to 9	4-70	Stores current Power Meter environment.
TYPE:			
CW		4-70	Selects Average Power Measurement.
PEAK		4-70	Selects Peak Power Measurement.
RMS		4-70	Selects RMS Power Measurement.
UL:			
LEVel *n*	0.0000 to 200.00	4-70	Sets Upper Limit Level in Watts.
STATe *b*	1 or 0	4-70	Enables/Disables Upper Limit.
M_PWR?		4-70	Returns Power Meter reading in mW.

COMMAND	RANGE	PAGE	DESCRIPTION
PROGRAM COMMANDS			
PROGram:			
STARTup:			
NAME "*name*"		4-99	Selects macro *name* to execute at power up.
NAME?		4-99	Returns name of macro set to execute at power up.
DELETE		4-99	Deletes power up macro designation (not macro).
RECEIVER COMMANDS			
NOTE: Queries return -1 if data is not available or is invalid.			
CALibration:FMZ *n*	1 to 4	4-21	Zeros FM Deviation Meter, displays Rcvr Opn Screen, and selects FM*n* Modulation.
RECeive:			
AGC:			
AUTO		4-14	Selects Auto Mode for AGC.
MANual *n*	0 to 255	4-14	Selects Manual Mode and level for AGC.
USER:*xxx*	DATA, HIGH, MEASure, SPeech, TYPE1, TYPE2 or TYPE3	4-14	Selects User Defined Mode and type for AGC.
CHANnel *n*	1 to 1023	4-14	Sets Receiver to selected cellular channel.
CHANnel:			
FORMat:			
AMPS:			
FORward		4-14	Selects AMPS Forward channels for Channel Mode.
REVerse		4-14	Selects AMPS Reverse channels for Channel Mode.
ETACS:			
FORward		4-14	Selects ETACS Forward channels for Channel Mode.
REVerse		4-14	Selects ETACS Reverse channels for Channel Mode.
NAMPS:			
FORward		4-14	Selects NAMPS Forward channels for Channel Mode.
REVerse		4-14	Selects NAMPS Reverse channels for Channel Mode.
NT400:			
FORward		4-15	Selects NT400 Forward channels for Channel Mode.
REVerse		4-15	Selects NT400 Reverse channels for Channel Mode.
FORMat?		4-15	Returns channel format setting.
DCS:			
INVert?		4-15	Returns 3 octal digits from inverted DCS.
NORMal?		4-15	Returns 3 octal digits from normal DCS.
STATe *b*	1 or 0	4-15	Enables/Disables DCS decoding.
DECode *type*	DTMF, TONE or DIGital	4-15	Sets Receiver decoding type.
DEVRms		4-15	Displays Deviation Meter (RMS) on Rcvr Opn Screen.
DIGital *type*	DCS, DCSINV, DST POCSAG or DSAT	4-15	Sets digital type.
DISTortion		4-15	Displays Distortion Meter on Rcvr Opn Screen.
DMM		4-16	Displays Digital Multimeter on Rcvr Opn Screen.
DSAT:			
STATe *b*	1 or 0	4-16	Enables/Disables DSAT decode.
DSAT?		4-16	Returns DSAT reading (NAMPS option installed).
DST:			
STATe *b*	1 or 0	4-16	Enables/Disables DST decode.
DST?		4-16	Returns DST reading (NAMPS option installed).
DTMF:			
STATe *b*	1 or 0	4-16	Enables/Disables DTMF decode.
DTMF?		4-16	Returns string of decoded digits.
FIND:			
FREQuency?		4-16	Searches for first frequency > Reference Level.
REFerence *n*	-110 to -5	4-16	Sets Find Reference Level in dBm.
REFerence?		4-16	Returns Find Reference Level setting in dBm.
FREQuency *f*[*units*]	250.0 to 999999.9 kHz; HZ, KHZ or MHZ	4-17	Sets Receiver Frequency.
FREQuency?		4-17	Returns Receiver Frequency setting in kHz.
INPut:			
ANTenna		4-17	Selects ANTENNA IN Connector for Receiver Input.
ATTenuation *n*	0, 20 or 40	4-17	Sets Receiver Input Attenuation Level in dB.
TR		4-17	Selects T/R Connector for Receiver Input.

COMMAND	RANGE	PAGE	DESCRIPTION
RECeive:			
MODE *type*	DIR, CHAN or SCAN	4-17	Selects Receiver Operation Mode.
MODMeter		4-17	Displays Modulation Meter on Rcvr Opn Screen.
MODulation:			
AM*n*	1 or 2	4-17	Selects an Amplitude Modulation Type.
BFO		4-17	Selects Beat Frequency Oscillation.
FM*n*	1, 2, 3 or 4	4-17	Selects a Frequency Modulation Type.
LSB		4-18	Selects Lower Side Band Modulation.
PM		4-18	Selects Phase Modulation.
USB		4-18	Selects Upper Side Band Modulation.
USER:			
FILTer *f*	3, 30 or 300	4-18	Sets User Defined IF Filter cutoff frequency in kHz.
MODulation:	AM, FM, PM, DATA, BFO, LSB or USB	4-18	Selects User Defined Modulation Type.
POST:			
APASs		4-18	Selects All Pass Post Detection Filter.
BPASs *fl,fh*	.5 to 20; 0.1 to 30	4-18	Selects Bandpass Post Detection Filter and sets low-high cutoff frequencies in kHz.
CWEight		4-18	Selects C-Weighted Post Detection Filter.
HPASs *f*	.5 to 20	4-18	Selects High-Pass Post Detection Filter and sets cutoff frequency in kHz.
LPASs *f*	.1 to 30	4-18	Selects Low-Pass Post Detection Filter and sets cutoff frequency in kHz.
MODulation?		4-18	Returns Modulation Type.
OUTput:			
AUDio *b*	1 or 0	4-19	Enables/Disables AUDIO OUT Connector Output.
DEMOD *b*	1 or 0	4-19	Enables/Disables DEMOD OUT Connector Output.
SPEAKer *b*	1 or 0	4-19	Enables/Disables Test Set Speaker Output.
PMRms		4-19	Displays Phase Meter (RMS) on Rcvr Opn Screen.
POCSAG:			
CAPcode?		4-19	Returns received capcode.
MESSage?		4-19	Returns message string.
RATe *b*	1 or 0	4-19	Selects High (1) or Low (0) POCSAG rate to decode.
RATe?		4-19	Returns selected POCSAG rate.
STATe *b*	1 or 0	4-19	Enables/Disables POCSAG decoding.
TYPE?		4-19	Returns POCSAG Function type.
RCL *n*	1 to 9	4-20	Recalls stored Receiver environment.
SCAN:			
ABORt		4-20	Stops Receiver Scan Function.
CONTinue		4-20	Starts or, if paused, continues Scan Function.
FREQuency?		4-20	Returns frequency being scanned.
INCrement *f*	0.0 to 99999.9	4-20	Sets Receiver Scan increment in kHz.
PAUSe *t*	0.0 to 99.9	4-20	Sets Receiver Scan Pause time in seconds.
PAUSe?		4-20	Returns 1 if paused, 0 otherwise.
RATe *t*	0.00 to 99.99	4-20	Sets Receiver Scan Rate in seconds.
STARt *f*	250.0 to 999999.9	4-20	Sets Receiver Scan start frequency in kHz.
STOP *f*	250.0 to 999999.9	4-20	Sets Receiver Scan stop frequency in kHz.
SINAD		4-20	Displays SINAD Meter on Receiver Operation Screen.
SQUelch *n*	0.0 to 1.0	4-20	Sets squelch level.
SQUelch?		4-21	Returns squelch level setting.
STORe *n*	1 to 9	4-21	Stores current Receiver environment.
TONE:			
STATe *b*	1 or 0	4-21	Enables/Disables Audio Tone decoding.
TYPE *xxx*	CCIR, EEA, EIA, ZVEI, DDZVEI, DZVEI, NATEL, EURO, TONE56, CCIRH, CCIRH4 or USER	4-21	Selects Audio Tone Type to decode.
TONE?		4-21	Returns received Audio Tone sequence string.
VOLume *n*	0.0 to 1.0	4-21	Sets volume level.
VOLume?		4-21	Returns volume level setting.
VOLume:			
AUTO b		4-21	Enables/Disables Automatic Volume Control.
AUTO?		4-21	Returns Automatic Volume Control state.

COMMAND	RANGE	PAGE	DESCRIPTION
RF GENERATOR COMMANDS			
GENerator:			
AF		4-4	Displays AF Level Meter on RF Gen Opn Screen.
CHANnel *n*	1 to 1023	4-4	Selects RF Generator cellular channel.
CHANnel:			
FORMat:			
AMPS:			
FORward		4-4	Selects AMPS Forward channels for Channel Mode.
REVerse		4-4	Selects AMPS Reverse channels for Channel Mode.
ETACS:			
FORward		4-4	Selects ETACS Forward channels for Channel Mode.
REVerse		4-4	Selects ETACS Reverse channels for Channel Mode.
NAMPS:			
FORward		4-4	Selects NAMPS Forward channels for Channel Mode.
REVerse		4-4	Selects NAMPS Reverse channels for Channel Mode.
NT400:			
FORward		4-4	Selects NT400 Forward channels for Channel Mode.
REVerse		4-4	Selects NT400 Reverse channels for Channel Mode.
FORMat?		4-4	Returns channel format setting.
DCS:			
INVert *nnn*	000 to 777	4-5	Generates 3 digit octal DCS Code in inverted mode.
NORMal *nnn*	000 to 777	4-5	Generates 3 digit octal DCS Code in normal mode.
STOP		4-5	Stops generating continuous DCS Code.
DIAL "*sequence*"	0 to 9 (each digit)	4-5	Encodes sequence (≤16 digits) as 2805 Pulse Code.
DIAL:			
FREQuency *f*	0.0 to 40000.0	4-5	Sets 2805 Pulse Tone Frequency.
FREQuency?		4-5	Returns 2805 Pulse Tone Frequency.
DISTortion		4-5	Displays Distortion Meter on RF Gen Opn Screen.
DMM		4-5	Displays DMM on RF Generator Operation Screen.
DSAT *n*	0 to 6	4-5	Generates DSAT code (NAMPS option installed).
DSAT:STOP		4-5	Stops generating continuous DSAT code.
DST *n*	0 to 6	4-5	Generates DST code.
DST:STOP		4-5	Stops generating continuous DST code.
DTMF "*seq*",*mark*,*space*	0 to 9 (each digit), 25 to 9999 (ms), 25 to 99 (ms)	4-6	Generates DTMF coded sequence (≤16 digits).
FREQuency *f*[*units*]	250.0 to 999999.99 kHz; HZ,KHZ or MHZ	4-6	Sets RF Generator Frequency.
FREQuency?		4-6	Returns RF Generator Frequency setting in kHz.
IMTS "*sequence*"	0 to 9 (each digit)	4-6	Generates sequence (≤16 digits) as IMTS Code.
LEVel *n*[*units*]	-137.0 to 0.0 (dBm); 0.031 µV to 0.224 V; DBm V, MV (mV) or UV (µV)	4-6	Sets RF Generator Output Level.
LEVel:UNIT?		4-6	Returns current units for RF Generator Output Level.
LEVel?		4-6	Returns RF Generator Output Level in current units.
MODE *type*	DIRect or CHANnel	4-6	Selects RF Generator Operation Mode.
MTS "*sequence*"	0 to 9	4-6	Generates sequence (≤16 digits) as MTS Code.
OUTput:			
AUDio *b*	1 or 0	4-6	Routes AF Gen Output to AUDIO OUT Connector.
DEMOD *b*	1 or 0	4-7	Routes AF Gen Output to DEMOD OUT Connector.
POCSAG:			
ALPHA:			
LOWer *capcode*	0 to 2097151	4-7	Generates lower case alpha message for capcode.
NUMeric *capcode*	0 to 2097151	4-7	Generates alphanumeric message for capcode.
SPECial *capcode*	0 to 2097151	4-7	Generates special character message for capcode.
UPPer *capcode*	0 to 2097151	4-7	Generates upper case alpha message for capcode.
BEEP *n*,*capcode*	1 to 4, 0 to 2097151	4-7	Generates *n* Tone beep POCSAG message for capcode.
NUMeric *capcode*	0 to 2097151	4-7	Generates numeric message for capcode.
RATe *b*	1 or 0	4-7	Selects High (1) or Low (1) POCSAG rate.
RATe?		4-7	Returns current POCSAG rate setting.
RCL *n*	1 to 9	4-8	Recalls stored RF Generator environment.
SINAD		4-8	Displays SINAD Meter on RF Gen Opn Screen.
SPEAKer:SOURce *type*	OFF, FGEN, SINAD or EXTMOD	4-8	Routes selected connector or source to Test Set Speaker.
STORe *n*	1 to 9	4-8	Stores current RF Generator environment.
TONE "*sequence*"		4-8	Generates sequence as an Audio Tone.

COMMAND	RANGE	PAGE	DESCRIPTION
GENerator:			
TONE:			
TYPE *code*	CCIR, EEA, EIA, ZVEI, DDZVEI, DZVEI, NATEL, EURO, TONE56, CCIRH, CCIRH4 or USER	4-8	Selects an Audio Tone Type.
USER:			
DEFine "*id*",*f*,*d*	0 to 9 or A to T, 0.0 to 9999.9, 20.0 to 9999.9	4-8	Assigns frequency *f* in Hz and duration *d* in ms to *id* character for User Defined Audio Tone Type.
TREMote *f*	1050 to 2050 (by 100)	4-9	Generates Tone Remote sequence for *f* tone in Hz.
TREMote:STOP		4-9	Stops Tone remote Guard Tone.

SCOPE COMMANDS (SEE OSCILLOSCOPE COMMANDS.)

SCREEN COMMANDS

SCREEN:		
AF	4-2	Displays AF Meter Operation Screen.
ANLZ	4-2	Displays Spectrum Analyzer Operation Screen.
BER	4-2	Displays BER Meter Operation Screen.
CELL	4-2	Displays Forward Control Channel Screen of AMPS Cell Site Monitor (AMPS option installed).
DEViation	4-2	Displays Deviation Meter (Peak) Operation Screen.
DISTortion	4-2	Displays Distortion Meter Operation Screen.
DMM	4-2	Displays DMM Meter Operation Screen.
DRMS	4-2	Displays Deviation Meter (RMS) Operation Screen.
DUPlex	4-2	Displays Duplex Operation Screen.
DUPRX	4-2	Displays Duplex Transmitter Operation Screen.
DUPTX	4-2	Displays Duplex Receiver Operation Screen.
FREQuency	4-2	Displays Frequency Error Meter Operation Screen.
FUNC	4-2	Displays AF Generator Operation Screen.
GENCELLular	4-2	Displays Main Menu of AMPS Mobile Simulator (AMPS option installed).
GENerator	4-2	Displays RF Generator Operation Screen.
GENRECC	4-2	Displays Reverse Control simulation Screen of AMPS Mobile Simulator (AMPS option installed).
GENRVC	4-3	Displays Reverse Voice Channel Simulation Screen of AMPS Mobile Simulator (AMPS option installed).
MODulation	4-3	Displays Modulation Meter Operation Screen.
PM	4-3	Displays Phase Meter Operation Screen.
PMRMS	4-3	Displays Phase Meter (RMS) Operation Screen.
POWer	4-3	Displays Power Meter Operation Screen.
RECeiver	4-3	Displays Receiver Operation Screen.
SCOPe	4-3	Displays Oscilloscope Operation Screen.
SIGnal	4-3	Displays Signal Strength Meter Operation Screen.
SINAD	4-3	Displays SINAD Meter Operation Screen.
USER	4-3	Displays blank User Screen.

SETUP COMMANDS

SETUP:		
AF	4-1	Selects AF Meter routing.
ANLZ	4-1	Selects Spectrum Analyzer routing.
DISTortion	4-1	Selects Distortion Meter routing.
DUPlex	4-1	Selects Duplex routing.
DUPRX	4-1	Selects Duplex Receiver routing.
DUPTX	4-1	Selects Duplex Transmitter routing.
FUNC	4-1	Selects AF Generator routing.
GENerator	4-1	Selects RF Generator routing.
MONitor	4-1	Selects Generator/Monitor routing.
RECeiver	4-1	Selects Receiver routing.
SCOPe	4-1	Selects Oscilloscope routing.
SINAD	4-1	Selects SINAD Meter routing.

COMMAND	RANGE	PAGE	DESCRIPTION
SIGNAL STRENGTH METER COMMANDS			
M_SIG:			
PEAK?		4-80	Returns Signal Strength Meter Peak reading.
PH *b*	1 or 0	4-80	Enables/Disables Peak Hold Feature.
RCL *n*	1 to 9	4-80	Recalls stored Signal Strength Meter environment.
STORe *n*	1 to 9	4-80	Stores current Signal Strength Meter environment.
M_SIG?	79	4-80	Returns Signal Strength Meter reading.
SINAD METER COMMANDS			
M_SINAD:			
ALARM *b*	1 or 0	4-78	Enables/Disables Alarm for Upper or Lower Limit.
AVErage *b*	1 or 0	4-78	Enables/Disables Averaging.
FILTer *f*	600 to 1400	4-78	Sets Notch Filter Frequency in Hz.
INPut:*type*	DEMOD, SINAD, XAUDIO or FGEN	4-78	Selects SINAD Meter Input.
LL:			
LEVel *n*	3.0 to 40.0	4-78	Sets Lower Limit Level in dB.
STATe *b*	1 or 0	4-78	Enables/Disables Lower Limit.
PEAK?		4-78	Returns SINAD Meter Peak reading in dB.
PH *b*	1 or 0	4-78	Enables/Disables Peak Hold Feature.
RCL *n*	1 to 9	4-78	Recalls stored SINAD Meter environment.
RESolution *n*	0.1 or 0.5	4-78	Sets SINAD Meter readout resolution in dB.
RESolution?		4-78	Returns SINAD Meter readout resolution setting in dB.
SELect:			
CWEight		4-79	Selects C-Weight Filter.
LPASs *f*	100 to 30000	4-79	Selects Low-Pass Filter with cutoff frequency in Hz.
STORe *n*	1 to 9	4-79	Stores current SINAD Meter environment.
UL:			
LEVel *n*	3.0 to 40.0	4-79	Sets Upper Limit Level in dB.
STATe *b*	1 or 0	4-79	Enables/Disables Upper Limit.
M_SINAD?		4-79	Returns SINAD Meter reading in dB.
SOUND COMMAND			
SOUND *f,t*		3-83	Sends *f* Hz tone for *t* ms to Test Set Speaker.
SPECTRUM ANALYZER COMMANDS			
ANLZ:			
AVErage *n*	1 to 100	4-55	Selects Average Mode for Analyzer.
CHANnel *n*	1 to 1023	4-55	Sets frequency to selected cellular channel.
CHANnel:			
FORMat:			
AMPS:			
FORward		4-55	Selects AMPS Forward for Analyzer Channel Mode.
REVerse		4-55	Selects AMPS Reverse for Analyzer Channel Mode.
ETACS:			
FORward		4-55	Selects ETACS Forward for Analyzer Channel Mode.
REVerse		4-55	Selects ETACS Reverse for Analyzer Channel Mode.
NAMPS:			
FORward		4-55	Selects NAMPS Forward for Analyzer Channel Mode.
REVerse		4-55	Selects NAMPS Reverse for Analyzer Channel Mode.
NT400:			
FORward		4-55	Selects NT400 Forward for Analyzer Channel Mode.
REVerse		4-55	Selects NT400 Reverse for Analyzer Channel Mode.
FORMat?		4-55	Returns channel format setting.
COMPare *n*	1 to 9	4-56	Selects Compare Mode with stored trace *n*.
FIND:			
FREQuency?		4-56	Searches for first frequency > Reference Level.
REFerence *n*	(varies)	4-56	Sets Find Reference Level in dB.
REFerence?		4-56	Returns Find Reference Level.
FREQuency *f*	250.0 to 999999.9	4-56	Sets Analyzer Frequency in kHz.
FREQuency?		4-56	Returns Analyzer Frequency setting in kHz.
FULL		4-56	Selects full size Analyzer for other Operation Screens.

COMMAND	RANGE	PAGE	DESCRIPTION
ANLZ:			
INPut:			
ANTenna		4-56	Selects ANTENNA IN Connector for Analyzer Input.
Attenuation *n*	0, 20 or 40	4-56	Selects Input Attenuation in dB.
Attenuation?		4-56	Returns Input Attenuation setting in dB.
TR		4-56	Selects T/R Connector for Analyzer Input.
INPut?		4-57	Returns Analyzer Input source.
LIVe		4-57	Selects Live Mode for Analyzer.
MARKer:			
AOFF		4-57	Disables both Markers.
DELTa:			
AMPLitude?		4-57	Returns amplitude difference between mkrs in dB.
FREQuency?		4-57	Returns frequency difference between mkrs in MHz.
POINt?		4-57	Returns position difference between mkrs in graticules.
TRACK *b*	1 or 0	4-57	Enables/Disables Marker Tracking feature.
MARKer1:			
AMPLitude?		4-57	Returns amplitude of Trace at Marker 1 crossing.
FREQuency?		4-57	Returns Marker 1 frequency position in kHz.
POINt *n*	1 to 100	4-57	Sets Marker 1 position in graticules.
POINt?		4-57	Returns Marker 1 position setting in graticules.
STATe *b*	1 or 0	4-58	Enables/Disables Marker 1.
MARKer2:			
AMPLitude?		4-57	Returns amplitude of Trace at Marker 2 crossing.
FREQuency?		4-57	Returns Marker 2 frequency position in kHz.
POINt *n*	1 to 100	4-57	Sets Marker 2 position in graticules.
POINt?		4-57	Returns Marker 2 position in graticules.
STATe *b*	1 or 0	4-58	Enables/Disables Marker 2.
MODE *type*	DIRect or CHANnel	4-58	Selects Analyzer RF Mode.
NORMalize		4-58	Normalizes Analyzer Trace to match RF Gen Output.
PEAK		4-58	Selects Peak Hold Mode for Analyzer.
QTR		4-58	Selects 1/4 size Analyzer for other Operation Screens.
RCL *n*	1 to 9	4-58	Recalls stored Spectrum Analyzer Trace/environment.
RLEVel?		4-58	Returns offset value used in 2 dB/div scale.
SCALe *n*	2 or 10	4-59	Selects Analyzer Units/Division Factor in dB.
UNIT:*type*	DBM, DBMV, DBUV, DBUW, DBV or DBW	4-59	Sets Analyzer Scale Units.
UNIT?		4-59	Returns Analyzer Scale Units setting.
SCALe?		4-59	Returns Analyzer Units/Division setting in dB.
SCAN *n*	0 or 1 to 100000 (1-2-5)	4-59	Sets Analyzer Scan Width in kHz.
SCAN?		4-59	Returns Analyzer Scan Width setting kHz.
STATe *b*	1 or 0	4-59	Enables/Disables Analyzer in other Operation Screens.
STORe *n*	1 to 9	4-59	Stores current Spectrum Analyzer Trace/environment.
TOP?		4-59	Returns top of screen scale value in current units.
TRACE:			
DATA *n,offset,v,...,v*	1 to 9,0 to 399,0 to 255	4-60	Replaces points in stored Trace starting at *offset.*
DATA? *n,offset,points*	0 to 9,0 to 399,1 to 400	4-60	Sends Trace data points directly to Host.
GET *name,n*	array name,0 to 9	4-60	Assigns values of Trace *n* to a declared array.
GET? *n,offset*	0 to 9,0 to 399	4-60	Returns vertical point value of Trace.
MAX? *n,offset,points*	0 to 9,0 to 399,1 to 400	4-60	Returns x,y point position of maximum vertical value.
MIN? *n,offset,points*	0 to 9,0 to 399,1 to 400	4-61	Returns x,y point position of minimum vertical value.
PUT *name,n*	array name,1 to 9	4-61	Assigns values of an array to Trace *n.*
TRACK:			
BWIDth *f*	.3, 3, 30, 300 or 3000	4-61	Selects Tracking Generator bandwidth in kHz.
BWIDth?		4-61	Returns Tracking Generator bandwidth setting in kHz.
LEVel *n*	-127.0 to 0.0	4-61	Sets Tracking Generator Level in dB.
LEVel?		4-61	Returns Tracking Generator Level setting.
OUTput:			
DUPlex		4-62	Selects DUPLEX OUT Connector as Track Gen Output.
TR		4-62	Selects T/R Connector as Tracking Generator Output.
OUTput?		4-62	Returns current Tracking Generator Output Connector.
RESolution:			
HIGH		4-62	Selects High Tracking Generator Resolution.
LOW		4-62	Selects Low Tracking Generator Resolution.
MED		4-62	Selects Medium Tracking Generator Resolution.
RESolution?		4-62	Returns Tracking Generator Resolution setting.
STATe *b*	1 or 0	4-62	Enables/Disables Tracking Generator.
STATe?		4-62	Returns Tracking Generator State.

COMMAND	RANGE	PAGE	DESCRIPTION
STATUS COMMANDS			
STATus:			
OPERation:			
CONDition?		3-43	Returns Operation Status condition register value.
ENABle *n*	0 to 65535	3-43	Sets Operation Status enable register contents.
ENABle?		3-43	Returns Operation Status enable register setting.
EVENt?		3-43	Returns Operation Status event register value.
INSTRument:			
CONDition?		3-44	Returns Operation Instrument condition register value.
ENABle *n*	0 to 65535	3-44	Sets Operation Instrument enable register contents.
ENABle?		3-44	Returns Operation Instrument enable register setting.
EVENt?		3-44	Returns Operation Instrument event register value.
QUEStionable:			
CONDition?		3-44	Returns Questionable Status condition register value.
ENABle *n*	0 to 65535	3-44	Sets Questionable Status enable register contents.
ENABle?		3-44	Returns Questionable Status enable register setting.
EVENt?		3-44	Returns Questionable Status event register value.
INSTRument:			
CONDition?		3-44	Returns Instrument Status condition register value.
ENABle *n*	0 to 65535	3-44	Sets Instrument Status enable register contents.
ENABle?		3-44	Returns Instrument Status enable register setting.
EVENt?		3-44	Returns Instrument Status event register value.
ISUMmary:			
CONDition?		3-45	Returns Instrument Summary Status condition register value.
ENABle *n*	0 to 65535	3-45	Sets Instrument Summary Status enable register contents.
ENABle?		3-45	Returns Instrument Summary Status enable register setting.
EVENt?		3-45	Returns Instrument Summary Status event register value.
STRING FUNCTION COMMANDS			
ASC($)		3-17	Returns ASCII value of first character in string $.
CHR(n)		3-16	Returns character equivalent of ASCII number n.
LEN($)		3-17	Returns length of string $ in number of characters.
PIXLEN($)		3-17	Returns length of string $ in pixels.
STR(n)		3-17	Returns string equivalent of number n.
STRPOS($1,$2)		3-18	Returns position of string $2 inside string $1.
TAB(n)		3-16	Returns blank string containing n number of spaces.
VAL($)		3-17	Returns numeric value represented by string $.

COMMAND	RANGE	PAGE	DESCRIPTION
SYSTEM COMMANDS			
SYSTem:			
COMMunicate:			
GPIB:			
ADDRess *a*	0 to 31	3-36	Sets 1600S GPIB address.
CONTroller		3-37	Sets 1600S as GPIB controller.
CMD "*string*"		3-38	Issues commands to subordinate test sets on GPIB.
DCL		3-38	Issues Device Clear message.
GET		3-38	Issues Group Execute Trigger.
LONly		3-39	Sets 1600S to Listen Only mode.
SLAVe *a*	0 to 31	3-37	Sets address destination of pass through commands.
SPOLL? *a*	0 to 31	3-38	Performs Serial Poll on device with selected address.
SRQ?		3-38	Checks peripheral devices for Service Request.
TONly		3-39	Sets 1600S to Talk Only mode.
SERial:			
BAUD *n*	300 to 38400	3-34	Sets baud rate.
BITS *n*	7 or 8	3-34	Sets number of data bits per word.
ECHO *b*	1 or 0	3-34	Enables/Disables RCI echo.
PACE *type*	XON or NONE	3-34	Sets software handshake mode.
PARity *type*	NONE, ODD or EVEN MARK or SPACE	3-34	Sets parity check.
SBITs *n*	1 or 2	3-34	Sets number of stop bits per word.
DATE *year,month,day*	1992 to 9999, 1 to 12, 1 to 31	3-33	Sets date.
DATE?		3-33	Returns date.
DEFaults		3-33	Restores Test Set to factory default state.
ERRor?		3-33	Returns error number and message.
FREQuency:			
LOCK *b*	1 or 0	3-35	Enables/Disables RF lock feature.
LOCK?		3-35	Returns RF lock feature state.
KEY *n*		3-32	Simulates pressing Front Panel Key with keycode *n*.
KEY?		3-32	Returns keycode of last key or simulated key pressed.
KEY:			
DEFine *n,sequence*		3-32	Assigns command *sequence* to key with keycode *n*.
DELete *n*		3-32	Deletes command assignment from key with keycode *n*.
KLOCk *b*	1 or 0	3-35	Locks/Unlocks Front Panel Keyboard.
PLOT:			
GPIB		3-33	Selects 1600S GPIB Connector for plotter output.
SERial		3-33	Selects 1600S RS-232 Connector for plotter output.
PTHRough:			
GPIB [*a*],"*string*"	0 to 31	3-38	Sends command string out 1600S GPIB Connector.
GPIB? [*a*]	0 to 31	3-38	Waits to receive string at 1600S GPIB Connector.
SERial "*string*"		3-34	Sends command string out 1600S RS-232 Connector.
SERial?		3-34	Waits to receive string at 1600S RS-232 Connector.
SERial:			
KEY?		3-34	Waits to receive character at 1600S RS-232 Connector.
QUEue?		3-34	Returns non-zero value if data is in RS-232 queue.
SECurity:			
STATe *n*	0 to 255	3-35	Activates (1 to 255) or deactivates (0) system security.
STATe?		3-35	Returns system security setting.
TIME *hour,minute,second*	0 to 23, 0 to 59, 0 to 59	3-33	Sets time.
TIME?		3-33	Returns time.
WINDOW COMMANDS			
CLS		3-54	Clears everything except active readings from selected window.
SCREEN:USER		3-26	Clears everything from active window.
WCLOSE *n*	1 to 16	3-87	Erases selected generated window(s) from CRT.
WINDOW?		3-88	Returns number of currently selected window.
WMOVE *x,y*	0 to 639,0 to 349	3-89	Moves currently selected window to selected new top left xy location.
WOPEN *c,x1,y1,x2,y2*	0 to 15,0 to 639, 0 to 349,0 to 639, 0 to 349	3-89	Creates windows.
WSEL *n,h*	0 to 15,0 or 1	3-90	Selects and hides or shows window.

SECTION 6 - CREATING AND LOADING CSA SPECIFIC PROGRAMS

6-1 SETUP

6-1-1 TERMINAL EMULATION PROGRAM SETUP

Any RS-232 Terminal or PC with Terminal Emulation software can be used to create and load CSA macros. For the purpose of clarity and actual operation, the following procedures use Procomm Plus™ as the Terminal Emulation program.

STEP PROCEDURE

1. Enter Procomm Plus™ from Host System terminal.

2. Set Line Settings as follows:

PARAMETER	SETTING
Baud Rate	300 to 38400
Parity	NONE
Data Bits	8
Stop Bits	1
Port	(Comm Connector on PC used for RS-232 to CSA)

3. Set Terminal Options as follows:

PARAMETER	SETTING
Terminal emulation	ANSI
Duplex	FULL
Soft flow ctrl (Xon/Xoff)	ON
Hard Flow ctrl (RTS/CTS)	OFF
Line wrap	ON
Screen scroll	ON
CR translation	CR
BS translation	NON-DESTRUCTIVE
Break length (millisecs)	N/A
Enquiry (ENQ)	N/A
EGA/VGA true underline	N/A
Terminal width	80
ANSI 7 or 8 bit commands	8 Bit

4. Set ASCII Protocol Options as follows:

PARAMETER	SETTING
Echo locally	NO
Expand blank lines	NO
Expand tabs	NO
Character pacing (millisec)	0
Line pacing (1/10 sec)	0
Pace character	0
Strip 8th bit	NO
ASCII download timeout	N/A
CR translation (upload)	NONE
LF translation (upload)	STRIP
CR translation (download)	N/A
LF translation (download)	N/A

6-1-2 CONNECTION AND FM/AM-1600CSA SETUP

STEP PROCEDURE

1. Connect Host System Comm Connector, selected in Line Settings, to RS-232 Connector on FM/AM-1600CSA rear panel, using standard 9-pin to 9-pin RS-232 cable.

2. Press DPLX Mode Key to access Duplex Operation Screen.

3. Press "Sp Tst" Soft Function Key F5 to access Dual Mode Cellular Main Menu.

4. Press "AUX2" Soft Function Key F5 to access Auxiliary Functions Menu (#2).

5. Press 5 DATA ENTRY Key to access External I/O Parameters Screen.

6. Use FIELD SELECT Keys to select parameter. Use DATA SCROLL Spinner or DATA SCROLL ↑ and ↓ Keys to edit parameter. Set parameters as follows:

PARAMETER	SETTING
RS-232 BAUDRATE	(same as Host System Line Settings Baud rate)
RS-232 PARITY	NONE
RS-232 BIT LENGTH	8
RS-232 STOP BITS	1
RS-232 PACE	XON

7. Verify link: press Enter key on Host System terminal keyboard and verify question mark appears on Host System monitor.

6-2 WRITING PROGRAMS

STEP PROCEDURE

1. Write TMAC programs using text editor. Refer to examples in 6-2 and 6-5.

2. Save as text file (program file).

3. Upload program file according to 6-3-1.

MACRO NAME: minit1

SYNTAX: minit1

PURPOSE: Perform a Mobile Station originated call.

MACRO: (The following example uses the predefined constants F1 and F6 [see Appendix A].)

```
*dmc "minit1",begin
var done=0
var key_code
do
  css:setup
  css:chan 334
  css:rflvl -60
  css:start
  user
  css:call:type 0
  host ":keypad:soft"
  color 1,15
  keypad:label 1,"START"
  keypad:label 6,"Ret"
  color 1,11
  center "Mobile Init Call",0,5,640
  do
    key_code=val(host? "syst:key?")
  until (key_code=F6) or (key_code=F1)          // Waits for valid key.
  if key_code=6
    done=1
  else
    keypad:label 1,""                           // Erases F1 label.
    keypad:label 1,"STOP"
    keypad:label 6,""                           // Erases F6 label.
    css:call:proc:mobinit
    center "Place Call",0,35,640
    do
      key_code=val(host? "syst:key?")
    until (key_code=F1) or css:recc:status?
    erase:text 0,35,640
    center "Assign Channel",0,35,640
    if(key_code!=F1)
      css:call:proc:assign
      do
        key_code=val(host? "syst:key?)
      until (key_code=F1) or (meas:sat? > 5960) // Waits for somewhat valid SAT.
      if(key_code!=F1)
        erase:text 0,35,640
        center "Call Successfull",0,35,640
      endif
    endif
    while key_code!=F1
      key_code=val(host? "syst:key?")
    wend
  endif
until done
end
```

6-3 LOADING

6-3-1 UPLOAD

STEP	PROCEDURE

1. Enter Procomm Plus™ from Host System terminal.
2. Press PgUp key on Host System terminal and select ASCII Upload Protocol.
3. Type in program file name (from 6-2) and press Enter key.
4. If failure occurs:
   - Utilize Procomm Plus™ Log function to create separate text file (log file) of Upload function (program file with error messages).
   - Repeat Steps 1 through 3.
   - View log file and correct errors in original program file.
   - Repeat Steps 1 through 3.

6-3-2 STORING IN FLASH MEMORY

Use **MMEMory:STORe:MACRo** "*m*","*f*" command to store uploaded program file from 6-3-1 in Flash Memory. (The **MMEMory:STORe:MACRo** command also stores all currently loaded macros into Flash Memory.) Set *m* to name of designated macro to run from Front Panel, contained in uploaded program file. Set *f* to desired JBox Flash File. (*f* is the name to appear in the JBox Flash File Directory.) Refer to 7-13 for more information on Flash Memory commands.

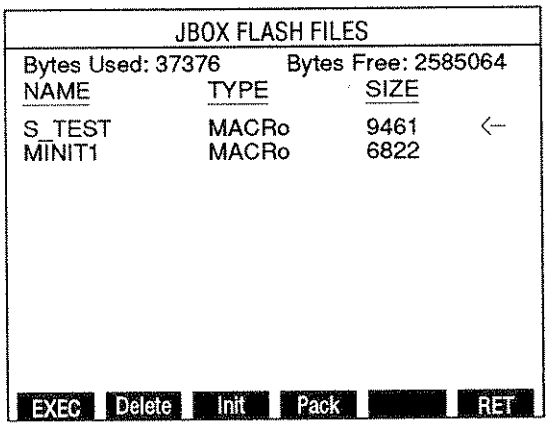
Examples:
```
mmem:store:macro "minit1","minit1" // Stores file, written in 6-2 and
                                   // and uploaded in 6-3-1, in Flash
                                   // Memory.

mmem:store:macro "sense_main","s_test"
                                   // Stores file from 6-6 in Flash
                                   // Memory.
```

6-4 FRONT PANEL OPERATION

STEP PROCEDURE

1. Press DPLX MODE Key to display Duplex Operation Screen.
2. Press "Sp Tst" Soft Function Key F5 to display Dual Mode Cellular Menu.
3. Press "AUX2" Soft Function Key F5 to display Auxiliary Functions Menu.
4. Press 6 DATA ENTRY Key to display JBox Flash Files directory.



9117129

Figure 6-1 JBox Flash Files Directory

5. Use ↑ and ↓ FIELD SELECT Keys to select desired macro file (indicated by arrow).
6. Press "EXEC" Soft Function Key F1 to execute macro.

6-6 MACRO EXAMPLES

FILE NAME: stest.mac

PURPOSE: Demonstrate several capabilities of FM/AM-1600CSA TMAC including:

- FM/AM-1600CSA TMAC Graphics
- Cell Site Simulation
- Using multiple macros
- Storing macros in Flash Memory
- Running a TMAC program from the Front Panel.
- Using FM/AM-1600S TMAC commands through the **HOST** command.

FILE: (The following example uses the predefined constants F1 and F6 [see Appendix A].)

```
var key_code                    // Declares a global variable.

/*          MACRO: delay1

           SYNTAX: delay1 n
                   (n=delay time in ms.)

          PURPOSE: Provide a delay if the F1 Soft Function Key is not pressed. */
*dmc "delay1",begin
if (key_code!=F1)
  delay $1
endif
end

/*          MACRO: focc_setup

           SYNTAX: focc_setup

          PURPOSE: Set up the System Parameter Overhead message sent on the Forward
                   Control Channel. */
*dmc "focc_setup",begin
css:setup                       // Sets CSA for Cell Site Simulation.
css:chan 333                    // Sets Forward Control Channel to 333 (879.99 MHz).
css:rflvl -60.0                 // Sets output RF Level to -60.0 dBm.
css:focc:pci 1                  // Sets Protocol Capability Indicator.
css:focc:rcf 0                  // Sets Read Control Filler bit.
css:focc:sid 0                  // Sets System Identification Number.
css:focc:n 1                    // Sets Number of Paging Channels.
css:focc:cmax 1                 // Sets Number of Access Channels.
css:focc:auth 0                 // Sets authentication bit.
css:start
end
```

```
/*          MACRO: call_setup
           SYNTAX: call_setup
          PURPOSE: Set up parameters used in making a digital call to the Mobile Station
                   (cellular phone). */
*dmc "call_setup",begin
css:call:type 1                          // Sets call type to digital.
css:call:chan 10                         // Sets Mobile Digital Traffic Channel
                                         // assignment.
css:call:dmac 5                          // Sets Digital Mode Attenuation Code.
css:call:slot 2                          // Sets to Timeslot 2.
css:call:pm 0                            // Sets Privacy Mode bit.
css:call:mem 0                           // Sets Message Encryption Mode to 0.
css:call:ef 0                            // Sets Extended Protocol Forware Channel
                                         // Indicator.
css:fdtc:enable:signal 1                 // Enables Signal field.
css:fdtc:signal:pitch 0;cadence 1        // Sets pitch and pattern of Alert tone.
css:fdtc:enable:calling:num 1            // Enables Calling Party Number field.
css:fdtc:calling:type 0                  // Sets Calling Party Type.
css:fdtc:calling:plan 0;pi 0;si 0        // Sets Calling Party Numbering Plan
                                         // Identification, Presentation Indicator
                                         // and Screening Indicator.
css:fdtc:enable:dmac 0;ta 1;dtx 0;dic 0
                                         // Disables Digital Mobile Attenuation
                                         // Code, Discontinuous Transmission bit
                                         // and Delay Interval.  Enables Time
                                         // Alignment field.
css:fdtc:ta 2                            // Sets Time Alignment to 2.
end

/*          MACRO: registration
           SYNTAX: registration
          PURPOSE: Force Mobile Station to register. */
*dmc "registration",begin
center "Registering Mobile",0,35,640
call_setup
do
  css:call:proc:reg
  delay1 1000
  key_code=val(host? "syst:key?")
until (key_code=F1) or css:recc:status?
print "MIN of mobile is ",recc:min?      // Prints information out CSA RS-232
print "ESN of mobile is "%h,recc:esn?    // Connector.
end
```

```
/*          MACRO: page1
            SYNTAX: page1
           PURPOSE: Page Mobile Station until Page Response is received. */
*dmc "page1",begin
erase:text 0,35,640
center "Paging Mobile",0,35,640
css:call:proc:page
do
  delay1 1000
  key_code=val(host? "syst:key?")
until (key_code=F1) or css:recc:status?
erase:text 0,35,640
center "Page Response Received",0,35,640
end

/*          MACRO: assign
            SYNTAX: assign
           PURPOSE: Assign Mobile Station to a digital voice channel. */
*dmc "assign",begin
css:call:proc:assign
delay1 1000                              // Waits for channel change.
erase:text 0,35,640
center "Digital Voice Channel Assigned",0,35,640
end

/*          MACRO: plc
            SYNTAX: plc
           PURPOSE: Send Physical Layer Control message to the Mobile Station until a Physical
                    Layer Control Ack message is received. */
*dmc "plc",begin
do
  css:fdtc:facch:plc
  delay1 500
  key_code=val(host? "syst:key?")
until((key_code=F1) or (rdtc:facch:msg? = "PLC ACK"))
erase:text 0,35,640
center "Physical Layer Control",0,35,640
end
```

```
/*          MACRO: alert
            SYNTAX: alert
           PURPOSE: Send Alert message to Mobile Station until Alert with Info Ack is received.
                    Wait until Connect message is received. */

*dmc "alert",begin
do
  css:fdtc:facch:alert
  delay1 500
  key_code=val(host? "syst:key?")
until((key_code=F1) or (rdtc:facch:amt? = "ALERT"))
erase:text 0,35,640
center "Answer Phone",0,35,640
while !(key_code=F1) and (rdtc:facch:msg? != "CONNECT")
  key_code=val(host? "syst:key?")
wend
erase:text 0,35,640
center "Connect",0,35,640
end
```

```
/*          MACRO: sense
            SYNTAX: sense
            PURPOSE: Vary the output RF Level of the FM/AM-1600CSA until the Mobile Station
                     (phone) reports a BER of 2% to 4%. */
*dmc "sense",begin
var timeout,ber
i=0
j=0
erase:text 0,35,640
center "Finding Sensitivity of Mobile",0,35,640
do
  css:fdtc:facch:meas                          // Sends Measurement Order until
  delay1 400                                   // Measurement Order Ack message is
  key_code=val(host? "syst:key?")              // received.
until (key_code=F1) or (rdtc:facch:msg?="MEAS ACK")
css:rflvl -90                                  // Starts with RF Level set at -90 dBm.
delay1 3000
do
  timeout=25
  css:rflvl -90 - i
  erase:text 0,140,640
  xyprint 40,140,"RFLVL: ",%d,-90 - i
  while ((rdtc:sacch:msg? != "CHAN QUAL1") and (--timeout))
    delay1 100
  wend
  ber=rdtc:sacch:ber?
  erase:text 0,160,640
  xyprint 40,160,"BER: ",%d,ber
  if ber=0                                     // BER < 0.01%
    i=i+2
  elif ber=1                                   // 0.01 < BER < 0.1
    i=i+1
  elif ber=2                                   // 0.1 < BER < 0.5
    i=i+.2
  elif ber=3                                   // 0.5 < BER < 1.0
    i=i+.1
  elif ber=4                                   // 1.0 < BER < 2.0
    i=i+.1
  elif ber=5                                   // 2.0 < BER < 4.0
    j=j+1
  elif ber=6                                   // 4.0 < BER < 8.0
    i=i-.1
  else                                         // BER > 8.0
    i=i-.2
  endif
  key_code=val(host? "syst:key?")
until j=6 or !(timeout) or key_code=F1         // Goes on when BER=5 is found 6 times.
```

(The sense macro continues on the next page.)

```
if key_code != F1                                    // Checks for STOP.
  print "Sensitivity of Mobile is ",%4d,-90-i        // Prints information out CSA
  print ""                                           // RS-232 Connector.
  erase:text 0,35,640
  center "Test Done, Press End or STOP",0,35,640
  while (key_code != F1) and (rdtc:facch:msg? != "RELEASE")
    key_code=val(host? "syst:key?")
    delay1 100
  wend
  erase:text 0,35,640
  center "Test Completed",0,35,640
endif
css:stop
delay 500
end
```

```
/*          MACRO: sense_main
           SYNTAX: sense_main
          PURPOSE: Provide the main calling routine for the sensitivity test. */
*dmc "sense_main",begin
var done=0
do
  focc_setup
  user
  host ":keypad:soft"
  color 1,15
  keypad:label 1,"START"
  keypad:label 6,"Ret"
  color 1,11
  center "Sensitivity Test",0,5,640
  do
    key_code=val(host? "syst:key?")
  until (key_code=F6) or (key_code=F1)    // Waits for valid key.
  if key_code=F6
    done=1                                // Goes to end if Ret key is pressed.
  else
    keypad:label 1,""                     // Erases F1 label.
    keypad:label 1,"STOP"
    keypad:label 6,""                     // Erases F6 label.
    registration
    if(key_code != F1)
      page1
    endif
    if(key_code != F1)
      assign
    endif
    if(key_code != F1)
      plc
    endif
    if(key_code != F1)
      alert
    endif
    if(key_code != F1)
      delay 3000
    endif
    if(key_code != F1)
      sense
    endif
  endif
until done
end
```

SECTION 7 - CSA SPECIFIC TMAC COMMANDS

7-1 GENERAL

This Section lists the FM/AM-1600CSA specific commands by Operation Mode. Commands are used remotely (Section 2) through GPIB (if option is installed) or RS-232, or as part of a TMAC program downloaded to the FM/AM-1600CSA. There are nine major modes of operation:

FOCC	Forward Control Channel Monitor
FVC	Forward Voice Channel Monitor
FDTC	Forward Digital Traffic Channel Monitor
RECC	Reverse Control Channel Monitor
RVC	Reverse Voice Channel Monitor
RDTC	Reverse Digital Traffic Channel Monitor
CSS	Cell Site Simulation
BER	Bit Error Rate for RDTC
MODACC	Modulation Accuracy for FDTC

The short forms of commands are shown in uppercase letters and the long forms are finished in lower case. When entering commands, it is not necessary to use a particular letter case. The TMAC compiler/interpreter is not case sensitive. Commands in each subsection are presented in logical operating order. Values are in decimal unless specified otherwise.

For remote operation, activate one of the 1600S operation modes and do not run Sp Tst. Refer to Section 6 for creating and downloading programs. Refer to Section 9 for programming examples.

Monitor commands, used to monitor forward channels from a Cell Site (Base Station) or reverse channels from a Mobile Phone (Station), implement the same functions available in the screens entered under Cell Site Data Monitor. **CSS:** commands, used to simulate a Cell Site, implement many of the same functions available in the screens entered under Cell Site Simulation. **CSS:CALL:PROCess:** commands add more flexibility than the front panel when setting up remote call processing. **BER:** commands implement the same functions available in the Base Station Digital Traffic BER screen. **MODacc:** commands implement the same functions available in the Modulation Accuracy screen.

7-2 HOST COMMANDS

HOST commands allow execution of 1600S specific commands and general commands limited to 1600S remote operation, when operating as the FM/AM-1600CSA.

HOST *"string"*
Issues commands, as strings, to the host FM/AM-1600S.

Example:
```
HOST "SETUP:DUPL"     // Passes the SETUP:DUPL command to the 1600S.
                      // 1600S configures Test Set routing and displays
                      // screen for Duplex Operation.
```

HOST? *"string?"*
Issues queries, as strings, to the host FM/AM-1600S.

Example:
```
HOST? "MEAS:POW?"     // Passes the MEAS:POW? query to the 1600S.
                      // 1600S returns a power meter reading (if
                      // configured to take power meter readings).
```

7-3 FORWARD CONTROL CHANNEL (FOCC) MONITOR COMMANDS

7-3-1 FORWARD CONTROL CHANNEL CONTROL

FOCC:

SETup
Sets up the FM/AM-1600CSA as when entering the Forward Control Channel screen. (Screen is not displayed.) The FM/AM-1600S is forced into Duplex Mode through selection of Duplex screen. The Antenna is selected as the input source.

CHANnel *n*
Selects Forward Control Channel to monitor. Range of *n* is 1 to 1023 (800 to 990 not used).

STARt
Starts decoding Forward Control Channel data. (Data returned by query. Refer to 7-3-2.)

STOP
Stops decoding Forward Control Channel data.

REMote:

STARt
Stops decoding and redirects the received Forward Control Channel data (10 kbit) as ASCII characters out the RS-232 Connector. Each character represents one nibble (4 bits) of data. (Set Baud Rate to 38400 prior to command execution to allow RS-232 to keep up with data received.)

STOP
Stops redirection of Forward Control Channel data out the RS-232 Connector.

WORD:

A
Selects Stream A to decode.

B
Selects Stream B to decode.

BOTH
Selects Streams A and B to decode. Responses come randomly from either Stream A or B, unless the **FOCC:STREAM** command is used.

STREAM:

A
Used prior to a decode data query, to select data from Stream A when both streams are being decoded (**FOCC:WORD:BOTH**).

B
Used prior to a decode data query, to select data from Stream B when both streams are being decoded (**FOCC:WORD:BOTH**).

CAPTure?
Used with Capture commands to test if a specific message (ORDER) and/or communication to specific Mobile Station (MIN) has occurred (returns 1). Returns 0 otherwise.

FOCC:

CAPTure:

CLEAR
Restarts Capture Function (same as RECAP from front panel).

SELect:

BOTH
Sets Capture Mode to BOTH (MIN and ORDER) to capture a specific message from communication with specific Mobile Station.

MIN
Sets Capture Mode to MIN to capture communication with specific Mobile Station.

NONE
Sets Capture Mode to NONE.

ORDER
Sets Capture Mode to ORDER to capture a specific message.

MODE?
Returns current Capture Mode setting (0 for NONE, 1 for ORDER, 2 for MIN or 3 for BOTH [MIN and ORDER]).

Messages (Orders) for capture (used when **FOCC:CAPTure:SELect:** command is set to ORDER or BOTH):

FOCC:

CAPTure:

A_ALERT
Selects Abbreviated Alert message as order for Capture.

AUDIT
Selects Audit message as order for Capture.

AUT_REG
Selects Autonomous Registration Confirmation message as order for Capture.

BSCHALCON
Selects Base Station Challenge Confirmation Message as order for Capture.

DIR_RTRY
Selects Directed-Retry message as order for Capture.

INTRCPT
Selects Intercept message as order for Capture.

LC
Selects Local Control message as order for Capture.

MSG_WTG
Selects Message Waiting message as order for Capture.

FOCC:

CAPTure:

N_AUT_REG
Selects Non-autonomous Registration Confirmation message as order for Capture.

PAGE
Selects Page as order for Capture.

RELease
Selects Release message as order for Capture.

REORDER
Selects Reorder message as order for Capture.

SLOT_1
Selects Slot 1 Handoff message as order for Capture.

SLOT_2
Selects Slot 2 Handoff message as order for Capture.

SLOT_3
Selects Slot 3 Handoff message as order for Capture.

SSD_UPdate
Selects Shared Secret Data Update as order for Capture.

UCHAL
Selects Unique Challenge message as order for Capture.

VC_DES
Selects Voice Channel Designation message as order for Capture.

ORDer?
Returns name (string) of order currently selected for Capture.

Communication to specific Mobile Station for capture (used when **FOCC:CAPTure:SELect:** command is set to MIN or BOTH):

FOCC:

CAPTure:

MIN *"n"*
Selects Mobile Identification Number to Capture. The Mobile Identification Number (*n*) is entered as a string, with quotation marks. Wild cards are indicated using a tilde (~).

Examples:
```
FOCC:CAPTure:MIN "316/522-4981"
FOCC:CAPTure:MIN "~ ~ ~/522-~9~1"
```

MIN?
Returns MIN currently selected for Capture.

7-3-2 FORWARD CONTROL CHANNEL DECODE DATA

The FOCC decode data queries return the specific monitored data items. If the specific data item is not available or has already been read, the query returns -1. Prior to initiating FOCC decode data queries, set FM/AM-1600CSA to FOCC initial setup (**FOCC:SET**) and start decoding (**FOCC:STAR**). If the queries are used as part of a TMAC Program, a **TPAUSE** command in conjunction with the queries may be needed to allow the data processing to take place. Refer to 9-2-1.

FOCC:

ACT?
Returns Global Action field value (four bit value).

AUTH?
Returns Authentication (one bit value).

AUTHBS?
Returns AUTHBS (18 bit value).

BIdle?
Returns Busy-Idle bit. Busy-Idle bit, multiplexed in with FOCC data, is 0 if Reverse Control Channel is busy or 1 if Reverse Control Channel is idle.

BIS?
Returns Busy-Idle Status bit from the Access Type Parameters Global Action message. (Busy-Idle Status bit is 0 if monitoring the Busy-Idle bit is not required by the Mobile Station, otherwise 1.)

CHAN?
Returns Channel Number field indicating designated RF channel (1 to 1023).

CHANPOS1?
Returns Channel Position 1 (seven bit value).

CHANPOS2?
Returns Channel Position 2 (seven bit value).

CHANPOS3?
Returns Channel Position 3 (seven bit value).

CHANPOS4?
Returns Channel Position 4 (seven bit value).

CHANPOS5?
Returns Channel Position 5 (seven bit value).

CHANPOS6?
Returns Channel Position 6 (seven bit value).

CMAC?
Returns Control Mobile Attenuation Code (three bit value).

CMAX_1?
Returns number of Access Channels minus one.

FOCC:

CPA?
Returns Combined Paging/Access (one bit value).

DCC?
Returns Digital Color Code (two bit value).

DMAC?
Returns Digital Mobile Attenuation Code (four bit value).

DTX?
Returns Discontinuous Transmission (one bit value).

DVCC?
Returns Digital Verification Color Code (eight bit value).

E?
Returns Extended Address (one bit value).

EF?
Returns Extended Protocol Forward Channel Indicator (one bit value).

END?
Returns End Indication (one bit value).

EP?
Returns Extended Protocol (one bit value).

LOC CONTRL?
Returns Local Control message (16 bit value).

LOCAID?
Returns Location Area Identity (12 bit value).

LOCAL_MT?
Returns Local Control (Local Control message)/Message Type field (five bit value).

LREG?
Returns Location Registration (one bit value).

MBUSY:

OTH?
Returns Maximum Number of Busy Occurrences Allowed for Other Accesses (four bit value).

PGR?
Returns Maximum Number of Busy Occurrences Allowed for Page Responses (four bit value).

MEM?
Returns Message Encryption Mode (one bit value).

FOCC:

MIN?
Returns Mobile Identification Number (string).

MSZTR:

OTH?
Returns Maximum Number of Seizure Attempts Allowed for Other Accesses (four bit value).

PGR?
Returns Maximum Number of Seizure Attempts Allowed for Page Responses (four bit value).

N_1?
Returns number of Paging Channels minus one, to be scanned by the Mobile Station (five bit value).

NAWC?
Returns Number of Additional Words Coming (four bit value).

NEWACC?
Returns New Access Channel Starting Point (11 bit value).

OLC?
Returns Overload Control Class (15 bit value).

ORDER?
Returns name (string) of received order.

ORDERCD?
Returns Order Code (five bit value).

ORDQ?
Returns Order Qualifier (three bit value).

PCI?
Returns Protocol Capability Indicator (one bit value).

PDREG?
Returns Power Down Registration (one bit value).

PM?
Returns Privacy Mode (one bit value).

PUREG?
Returns Power Up Registration (one bit value).

RANDSSD_1?
Returns value of 24 most significant bits of RANDSSD (24 bit value).

RANDSSD_2?
Returns value of bits 8 through 31 of RANDSSD (24 bit value).

FOCC:

RANDSSD_3?
Returns value of eight least significant bits of RANDSSD (eight bit value).

RANDU?
Returns RANDU received in the Unique Challenge message (24 bit value).

RAND1_A?
Returns value of 16 most significant bits of RAND (16 bit value).

RAND1_B?
Returns value of 16 least significant bits of RAND (16 bit value).

RCF?
Returns Read Control-Filler (one bit value).

REGH?
Returns Registration for Home Mobile Stations (one bit value).

REGID?
Returns Registration Identification (20 bit value).

REGINCR?
Returns Registration Increment field (12 bit value).

REGR?
Returns Registration for Roaming Mobile Stations (one bit value).

S?
Returns Serial Number (one bit value).

SCC?
Returns Supervisory Audio Tone Color Code (two bit value).

SDCC1?
Returns Supplementary Digital Color Code 1 (two bit value).

SDCC2?
Returns Supplementary Digital Color Code 2 (two bit value).

SID?
Returns System Identification Number (14 bit value).

VMAC?
Returns Voice Mobile Attenuation Code (three bit value).

WFOM?
Returns Wait For Overhead Message (one bit value).

7-3-3 FORWARD CONTROL CHANNEL RAW DATA

The FM/AM-1600CSA contains a 100 word data buffer to capture raw data (undecoded bit streams) received on the Forward Control Channel. Prior to initiating FOCC raw data queries, set the FM/AM-1600CSA to initial FOCC setup (**FOCC:SET**) and select the RF Channel (**FOCC:CHAN** *n*). Refer to 9-2-2.

FOCC:

RAW:

WORD:

A
Selects raw data from Stream A to monitor.

B
Selects raw data from Stream B to monitor.

BOTH
Selects raw data from Streams A and B, randomly, to monitor.

Messages (Orders) for raw data capture:

FOCC:

RAW:

CAPTure:

NONE
Sets Capture Mode to NONE.

A_ALERT
Selects Abbreviated Alert Message as order for Capture.

AUDIT
Selects Audit Message as order for Capture.

AUT_REG
Selects Autonomous Registration Confirmation message as order for Capture.

BSCHALCON
Selects Base Station Challenge Confirmation Message as order for Capture.

DIR_RTRY
Selects Directed-Retry message as order for Capture.

INTRCPT
Selects Intercept message as order for Capture.

LC
Selects Local Control Message as order for Capture.

FOCC:

RAW:

CAPTure:

MSG_WTG
Selects Message Waiting message as order for Capture.

N_AUT_REG
Selects Non-autonomous Registration Confirmation message as order for Capture.

PAGE
Selects Page as order for Capture.

RELease
Selects Release message as order for Capture.

REORDER
Selects Reorder message as order for Capture.

SLOT_1
Selects Slot 1 Handoff message as order for Capture.

SLOT_2
Selects Slot 2 Handoff message as order for Capture.

SLOT_3
Selects Slot 3 Handoff message as order for Capture.

SSD_UPdate
Selects Shared Secret Data Update as order for Capture.

UCHAL
Selects Unique Challenge message as order for Capture.

VC_DES
Selects Voice Channel Designation message as order for Capture.

ORDer?
Returns name (string) of order currently selected for Capture.

INDex?
Returns position (0 to 99) of Captured Order in the 100 word data buffer.

TRIGger *n*
Selects position of Captured Order in the 100 word data buffer. Range of *n* is 0 to 4.

STARt
Starts raw data Capture.

STOP
Stops raw data Capture.

FOCC:

RAW:

CAPTure?
Returns 1 if Capture condition has occurred; 0 otherwise.

FULL?
Returns 1 if 100 word data buffer is full; 0 otherwise.

A:

CHECK? *n*
Returns CRC Check result for selected data word from Stream A (0 [good], 1 [bad]). Range of *n* is 0 to 99.

DATA? *n*
Returns selected raw data word from Stream A. Range of *n* is 0 to 99.

PARITY? *n*
Returns Parity for selected data word from Stream A. Range of *n* is 0 to 99.

B:

CHECK? *n*
Returns CRC Check result for selected data word from Stream B (0 [good], 1 [bad]). Range of *n* is 0 to 99.

DATA? *n*
Returns Raw Data word from Stream B. Range of *n* is 0 to 99.

PARITY *n*
Returns Parity for selected data word from Stream B. Range of *n* is 0 to 99.

B_I? *n*
Returns Busy-Idle bit for selected data word. Range of *n* is 0 to 99.

TS? *n*
Returns Time Stamp in ms of selected data word. Range of *n* is 0 to 99.

ADDENDUM TO SECTION 7-3

7-3A FOCC

FOCC:

CONFigure:

USER
[FOCC:CONGigure:USER]
This command is identical to the **FOCC:SETup** command except that it selects the USER screen.

NONE
[FOCC:CONGigure:NONE]
This command is also identical to the **FOCC:SETup** command except that it does not select a screen.

PVI?
[FOCC:PVI?]
Returns current state of Protocol Version Indicator (1 bit value).

ORDER?
[FOCC:ORDER?]
Add the following orders:

G3FAX WTG	SMS WTG
G3 PAGE	ASYNC PAGE
MSG WTG	ANA VC DES

7-4 FORWARD VOICE CHANNEL (FVC) MONITOR COMMANDS

7-4-1 FORWARD VOICE CHANNEL CONTROL

FVC:

SETup
Sets up the FM/AM-1600CSA as when entering the Forward Voice Channel screen. (Screen is not displayed.)

STARt
Starts decoding Forward Voice Channel data. (Data returned by query. Refer to 7-4-2.)

STOP
Stops decoding Forward Voice Channel data.

CHANnel *n*
Selects Forward Voice Channel to monitor. Range of *n* is 1 to 1023 (800 to 900 not used).

SCC *n*
Specifies the SCC by providing the SAT frequency in Hz (*n*) corresponding to the SAT Color Code (SCC). Range of *n* is 5955 to 6044. The FVC decode task only decodes messages having a specified SCC. Sending the command as **FVC:SCC :MEAS:SAT?** insures the SCC matches the SAT frequency on the channel being monitored. The **:MEAS:SAT?** query returns the SAT frequency in Hz and the **FVC:SCC** sets up the SCC match as follows:

SAT Frequency	SCC
5955-5984 Hz	0 (00)
5985-6014 Hz	1 (01)
6015-6044 Hz	2 (10)

Using the command in this way insures only messages on the selected channel, and not an adjacent channel, are decoded.

7-4-2 FORWARD VOICE CHANNEL DECODE DATA

The FVC Decode Data queries return the last value decoded or a -1, if value has not been decoded since the last time requested. Prior to initiating FVC decode data queries, set FM/AM-1600CSA to FVC initial setup (**FVC:SET**) and start decoding data (**FVC:STAR**). If queries are used as part of a TMAC Program, a **TPAUSE** command in conjunction with the queries may be needed to allow the background decoding task time to run. Refer to 9-3-1.

FVC:

AUTHBS?
Returns Output Response of Base Station Authentication Algorithm (18 bit value).

CHAN?
Returns Channel Number field indicating designated RF channel (1 to 1023).

CHAR1?
Returns first Characters in Called Party Number message (Flash with Information) or Calling Party Number message (Alert with Information or Flash with Information) (up to 16 digits).

CHAR2?
Returns last Characters in Called Party Number message (Flash with Information) or Calling Party Number message (Alert with Information or Flash with Information) (up to 16 digits).

FVC:

CPN_RL?
Returns Calling Party Number Remaining Length (six bit value).

DMAC?
Returns Digital Mobile Attenuation Code (four bit value).

DVCC?
Returns Digital Verification Color Code (eight bit value).

EF?
Returns Extended Protocol Forward Channel Indicator (one bit value).

LOCAL_MT?
Returns Local Control (Local Control message)/Message Type field (five bit value).

MEM?
Returns Message Encryption Mode (one bit value).

ORDER?
Returns name (string) of received Order.

ORDERCD?
Returns Order Code (five bit value).

ORDQ?
Returns Order Qualifier (three bit value).

PI?
Returns Calling Party Number Presentation Indicator (two bit value).

PM?
Returns Privacy Mode (one bit value).

PSCC?
Returns Present SAT Color Code (two bit value).

PWRL?
Returns Power Level requested of Mobile Station in Power Level message (same as ORDQ).

RANDSSD1?
Returns value of 24 most significant bits of RANDSSD (24 bit value).

RANDSSD2?
Returns value of bits 8 through 31 of RANDSSD (24 bit value).

RANDSSD3?
Returns eight least significant bits of RANDSSD (eight bit value).

RANDU?
Returns RANDU received in the Unique Challenge message (24 bit value).

RL_W?
Returns Remaining Length in Words (five bit value).

FVC:

SBI?
Returns Shortened Burst Indicator (two bit value).

SCC?
Returns Supervisory Audio Tone Color Code (two bit value).

SI?
Returns Calling Party Screening Indicator (two bit value).

SIGnal?
Returns Signal field (eight bit value).

TA?
Returns Time Alignment offset (five bit value).

VMAC?
Returns Voice Mobile Attenuation Code (three bit value).

7-4-3 FORWARD VOICE CHANNEL RAW DATA

The FM/AM-1600CSA contains a variable capacity data buffer to capture raw data (undecoded bit stream) received on the Forward Voice Channel. Prior to initiating FVC raw data commands, set the FOCC Capture Mode to NONE (**FOCC:CAPT:SEL:NONE**) and set FM/AM-1600CSA to initial FVC setup (**FVC:SET**). Obtain new data information by setting *n* from 0 to current **FVC:RAW:DEPTH** setting minus one, when using raw data information queries. Refer to 9-3-2.

FVC:

RAW:

DEPTH *n*
Selects size of data buffer in data words. Range of *n* is 1 to 100 data words.

STARt
Starts receiving raw data into data buffer.

STOP
Stops receiving raw data into data buffer.

COUNT?
Returns number of data words received (increments with each new data word). Used to know when to query for new data information.

DATA? *n*
Returns selected raw data word (28 bits). Range of *n* is 0 to 99.

PARITY? *n*
Returns Parity for selected data word (12 bits). Range of *n* is 0 to 99.

CHECK? *n*
Returns CRC Check result for selected data word (0 [good], 1 [bad]). Range of n is 0 to 99.

TS? *n*
Returns Time Stamp of selected data word in seconds from 0 data word. Range of *n* is 0 to 99.

ADDENDUM TO SECTION 7-4

7-4A FVC

FVC:

CONFigure:

USER
[FVC:CONFigure:USER]
This command is identical to the **FVC:SETup** command except that it selects the USER screen.

NONE
[FVC:CONFigure:NONE]
This command is also identical to the **FVC:SETup** command except that it does not select a screen.

PVI?
[FVC:PVI?]
Returns current state of Protocol Version Indicator (1 bit value).

HYPERband?
[FVC:HYPERband?]
Returns current value of Hyperband (2 bit value).

ORDER?
[FVC:ORDER?]
Add the following orders:

G3FAX WTG	SMS WTG
G3 PAGE	ASYNC PAGE
MSG WTG	RELEASE W/INFO

7-5 FORWARD DIGITAL TRAFFIC CHANNEL (FDTC) MONITOR COMMANDS

7-5-1 FORWARD DIGITAL TRAFFIC CHANNEL CONTROL

FDTC:

SETup
Sets up the FM/AM-1600CSA as when entering the Forward Digital Traffic Channel screen. (Screen is not displayed.)

STARt
Starts decoding Forward Digital Traffic Channel data. (Data returned by query. Refer to 7-5-2.)

STOP
Stops decoding Forward Digital Traffic Channel data.

CHANnel *n*
Selects Forward Digital Traffic Channel to monitor. Range of *n* is 1 to 1023. (800 to 900 not used.)

SLOT *n*
Selects Digital Traffic Channel Timeslot. Range of *n* is 1 to 3.

DVCC?
Returns Digital Verification Color Code.

R0?
Returns VSELP frame energy value (0 to 31).

7-5-2 FORWARD DIGITAL TRAFFIC CHANNEL DECODE DATA

FDTC:FACCH: queries pertain to the blank and burst Fast Associated Control Channel (FACCH). **FDTC:SACCH:** queries pertain to the continuous Slow Associated Control Channel (SACCH).

FDTC:

FACCH:

MSGtype?
Returns Message Type (string):

ALERT	HANDOFF	RELEASE
AUDIT	LC	SBDA
BS ACK	MAINT	SCDA
FILL OCT	MEAS	S MEAS
FLASH	PLC	SR
FLASH ACK	PU	

AMT?
Returns Acknowledge Message Type (string).

AUTHBS?
Returns AUTHBS (18 bit value).

FDTC:

FACCH:

CALLING:

NUM?
Returns number of the calling party (string).

PI?
Returns Calling Party Presentation Indicator (two bit value).

PLANid?
Returns Calling Party Numbering Plan Identification (four bit value).

SI?
Returns Calling Party Screening Indicator (two bit value).

SPare?
Returns value of Calling Party Number spare bits (five bit value).

TYpe?
Returns Calling Party Number Type (three bit value).

CNPC?
Returns the Calling Number Presentation Code (four bit value).

DIC?
Returns Delay Interval Compensation (one bit value).

DIGits? *n*
Returns the 11 digits (string) of the selected digit set. Setting for *n* is 0 (DIGITS1), 1 (DIGITS2) or 2 (DIGITS3).

DMAC?
Returns Digital Mobile Attenuation Code (four bit value).

DTX?
Returns Discontinuous Transmission (one bit value).

HDVCC?
Returns Handoff Digital Verification Color Code (eight bit value).

LC?
Returns Local Control (five bit value).

LDP?
Returns Last Decoded Parameter (four bit value).

MEM?
Returns Message Encryption Mode (one bit value).

NOMW?
Returns Number of Messages Waiting (six bit value).

FDTC:

FACCH:

NV? *n*
Returns Number of Values (six bit value) for selected optional information element. Range of *n* is 0 (1st element) to 5 (6th element).

PD?
Returns Protocol Discriminator (two bit value).

PT? *n*
Returns Parameter Type (four bit value) for selected optional information element. Range of *n* is 0 (1st element) to 5 (6th element).

RANDSSD1?
Returns value of 24 most significant bits of RANDSSD (24 bit value).

RANDSSD2?
Returns value of bits 8 through 31 of RANDSSD (24 bit value).

RANDU?
Returns RANDU received in Unique Challenge message (24 bit value).

RATE?
Returns Channel Rate (one bit value).

RFCHAN? *n*
Returns RF Channel (1 to 1023) for the selected index (1 to 1023). Range of *n* is 0 to 11.

RL? *n*
Returns Remaining Length as number of octets (six bit value) remaining in selected message segment. Range of *n* is 0 to 2. (Segments pertain to the 0 to 2 instances the Calling Party Number may be sent in the Alert with Information or Flash with Information messages.)

RN?
Returns Request Number (four bit value).

SBI?
Returns Shortened Burst Indicator (two bit value).

SIGnal?
Returns Signal field (eight bit value).

TA?
Returns Time Alignment (five bit value).

TI?
Returns Timeslot Indicator (0 to 6). (0 is analog.)

VPM?
Returns Voice Privacy Mode (one bit value).

FDTC:

SACCH:

MSGtype?
Returns Message Type (string):

ALERT	HANDOFF	RELEASE
AUDIT	LC	SBDA
BS ACK	MAINT	SCDA
FILL OCT	MEAS	S MEAS
FLASH	PLC	SR
FLASH ACK	PU	

AMT?
Returns Acknowledge Message Type (string).

AUTHBS?
Returns AUTHBS (18 bit value).

CALLING:

NUM?
Returns number of the calling party (string).

PI?
Returns Calling Party Presentation Indicator (two bit value).

PLANid?
Returns Calling Party Numbering Plan Identification (four bit value).

SI?
Returns Calling Party Screening Indicator (two bit value).

SPare?
Returns value of Calling Party Number spare bits (five bit value).

TYpe?
Returns Calling Party Number Type (three bit value).

CNPC?
Returns Calling Number Presentation Code (four bit value).

DIC?
Returns Delay Interval Compensation (one bit value).

DIGits? *n*
Returns the 11 digits (string) of the selected digit set. Setting for *n* is 0 (DIGITS1), 1 (DIGITS2) or 2 (DIGITS3).

DMAC?
Returns Digital Mobile Attenuation Code (four bit value).

DTX?
Returns Discontinuous Transmission (one bit value).

FDTC:

SACCH:

HDVCC?
Returns Handoff Digital Verification Color Code value (eight bit value).

LC?
Returns Local Control (five bit value).

LDP?
Returns Last Decoded Parameter (14 bit value).

MEM?
Returns Message Encryption Mode (one bit value).

NOMW?
Returns Number of Messages Waiting (six bit value).

NV? *n*
Returns Number of Values (six bit value) for selected optional information element. Range of *n* is 0 (1st element) to 5 (6th element).

PD?
Returns Protocol Discriminator (two bit value).

PT? *n*
Returns Parameter Type (two bit value) for selected optional information element. Range of *n* is 0 (1st element) to 5 (6th element).

RANDSSD1?
Returns value of 24 most significant bits of RANDSSD (24 bit value).

RANDSSD2?
Returns value of bits 8 through 31 of RANDSSD (24 bit value).

RANDU?
Returns RANDU received in Unique Challenge message (24 bit value).

RATE?
Returns Channel Rate (one bit value).

RFCHAN? *n*
Returns RF Channel (1 to 1023) for the selected index. Range of *n* is 0 to 11.

RL? *n*
Returns Remaining Length as number of octets (six bit value) remaining in selected message segment. Range of *n* is 0 to 2. (Segments pertain to the 0 to 2 instances the Calling Party Number may be sent in the Alert with Information or Flash with Information messages.)

RN?
Returns Request Number (four bit value).

FDTC:

SACCH:

SBI?
Returns Shortened Burst Indicator (two bit value).

SIGnal?
Returns Signal Field (eight bit value).

TA?
Returns Time Alignment (five bit value).

TI?
Returns Timeslot Indicator (0 to 6). (0 is analog.)

VPM?
Returns Voice Privacy Mode (one bit value).

7-5-3 FORWARD DIGITAL TRAFFIC CHANNEL RAW DATA

The FM/AM-1600CSA contains a variable capacity data buffer to capture raw data (undecoded bit stream) received on the Forward Digital Traffic Channel. Prior to initiating FDTC raw data commands, set FM/AM-1600CSA to initial FDTC setup (**FDTC:SET**). Obtain new data information by setting *n* from 0 to current **FDTC:RAW:DEPTH** setting minus one, when using raw data information queries. Refer to 9-4-2.

FDTC:

RAW:

START
Starts raw data Capture.

STOP
Stops raw data Capture.

SELect:

FACCH
Selects Fast Associated Control Channel for raw data.

SACCH
Selects Slow Associated Control Channel for raw data.

DEPTH *n*
Selects size (number of data words) of data buffer. Range of *n* is 1 to 100.

COUNt?
Returns number of data words received (increments with each new data word). Used to know when to query for new data information.

CF? *n*
Returns Continuation Flag bit from selected data word. Range of *n* is 0 to 99.

FDTC:

RAW:

RSVD? *n*
Returns Reserved for Future Use bits from selected data word. Range of *n* is 0 to 99.

MESSage? *n,x*
Returns selected message byte from selected data word. Range of *n* is 0 to 99. Range of *x* (byte) is 0 to 5.

DVCC? *n*
Returns Digital Verification Color Code from selected data word. Range of *n* is 0 to 99.

TIME? *n*
Returns Time Stamp in ms of selected data word. Range of *n* is 0 to 99.

7-5-4 IS-54 RAW DATA

The FM/AM-1600CSA contains a fixed capacity data buffer (100 data words) to capture raw timeslot data (undecoded bit streams) received on the Forward Digital Traffic Channel. Prior to initiating FDTC raw data commands, set FM/AM-1600CSA to initial FDTC setup (**FDTC:SET**). Refer to 9-4-3.

FDTC:

IS54:

START
Starts IS-54 timeslot raw data Capture

STOP
Stops IS-54 timeslot raw data Capture.

COUNt?
Returns number of data words received (increments with each new data word). Used to know when to query for new data information.

SACCH? *n*
Returns 12 bits of the SACCH message in selected data word. Range of *n* is 0 to 99.

SYNC? *n*
Returns the 28 Synchronization bits from selected data word. Range of *n* is 0 to 99.

DATA? *n,x*
Returns SACCH character data from selected data word. Range of *n* is 0 to 99. Range of *x* (character) is 1 to 65.

CDVCC? *n*
Returns the 12 bit Coded Digital Color Code from selected data word. Range of *n* is 0 to 99.

TIME? *n*
Returns Time Stamp in ms of selected data word. Range of *n* is 0 to 99.

ADDENDUM TO SECTION 7-5

7-5A FDTC

FDTC:

CONFigure:

USER
[FDTC:CONFigure:USER]
This command is identical to the **FDTC:SETup** command except that it selects the USER screen.

NONE
[FDTC:CONFigure:NONE]
This command is also identical to the **FDTC:SETup** command except that it does not select a screen.

FACCH: or SACCH:

MSGtype?
[FDTC:FACCH: or SACCH:MSGtype?]
Add the following orders:

DEDICATED HANDOFF	CAPABILITY REQ	R-DATA ACCEPT
HYPERBAND MEAS	CAPABILITY RESP	R-DATA REJECT
BSMC	R-DATA	SOC

RFCHAN? *n*
[FDTC:FACCH: or SACCH:RFCHAN?]
Returns current value of RFCHAN (11 bit value) indexed by *n*. Range of *n* is (0-23).

PVI?
[FDTC:FACCH: or SACCH:PVI?]
Returns current state of Protocol Version Indicator (1 bit value)

DTXControl?
[FDTC:FACCH: or SACCH:DTXControl?]
Returns current state of DTX Control (1 bit value).

DCCHinfo:

HYPERband? *n*
[FDTC:FACCH: or SACCH:DCCHinfo:HYPERband?]
Returns current value of DCCH Info Hyperband (2 bit value) indexed by *n*. Range of *n* is 0 to 2.

CHANnel? *n*
[FDTC:FACCH: or SACCH:DCCHinfo:CHANnel?]
Returns current value of DCCH info Channel (8 bit value) indexed by *n*. Range of *n* is 0 to 2.

FDTC:

FACCH: or SACCH:

DCCHinfo:

DVCC? *n*
[FDTC:FACCH: or SACCH:DCCHinfo:DVCC?]
Returns current value of DCCH info DVCC (8 bit value) indexed by *n*. Range of *n* is 0 to 2.

MSGWTG:

TYPE? *n*
[FDTC:FACCH: or SACCH:MSGWTG:TYPE?]
Returns current value of Other Messages Waiting Info Type (4 bit value) indexed by *n*. Range of *n* is 0 to 14.

NUMBer? *n*
[FDTC:FACCH: or SACCH:MSGWTG:NUMBer?]
Returns current value of Number of Other Messages Waiting (6 bit value) indexed by *n*. Range of *n* is 0 to 14.

TASK?
[FDTC:FACCH: or SACCH:TASK?]
Returns current value of Task Status (3 bit value).

ATS?
[FDTC:FACCH: or SACCH:ATS?]
Returns current value of ATS (4 bit value).

PV?
[FDTC:FACCH: or SACCH:PV?]
Returns current value of Protocol Version (4 bit value).

CHANGE:

SOC?
[FDTC:FACCH: or SACCH:CHANGE:SOC?]
Returns current state of SOC Change Indicator (1 bit value).

BSMC?
[FDTC:FACCH: or SACCH:CHANGE:BSMC?]
Returns current state of BSMC Change Indicator (1 bit value).

DELTA:

TIME?
[FDTC:FACCH: or SACCH:DELTA:TIME?]
Returns current value of Delta Time (11 bit value).

FDTC:

FACCH: or SACCH:

VMI:

VC?
[FDTC:FACCH: or SACCH:VMI:VC?]
Returns current value of Voice Mode Voice coder (3 bit value).

PM_V?
[FDTC:FACCH: or SACCH:VMI:PM_V?]
Returns current value of Voice Privacy Mode (3 bit value).

DPM?
[FDTC:FACCH: or SACCH:DPM?]
Returns current state of Data Privacy Mode (1 bit value).

MEMC:

MEA?
[FDTC:FACCH: or SACCH:MEMC:MEA?]
Returns current value of Message Encryption Mode C Algorithm (2 bit value).

MED?
[FDTC:FACCH: or SACCH:MEMC:MED?]
Returns current value of Message Encryption Mode C Domain (2 bit value).

MEK?
[FDTC:FACCH: or SACCH:MEMC:MEK?]
Returns current value of Message Encryption Mode C Key (2 bit value).

MAP:

VPM?
[FDTC:FACCH: or SACCH:MAP:VPM?]
Returns current value of Voice Privacy Mode Map (4 bit value).

CODER?
[FDTC:FACCH: or SACCH:MAP:CODER?]
Returns current value of Voice Coder Map (6 bit value).

MEA:

DOMAIN?
[FDTC:FACCH: or SACCH:MAP:MEA:DOMAIN?]
Returns current value of Message Encryption Algorithm Map Domain (8 bit value).

Algorithms? *n*
[FDTC:FACCH: or SACCH:MAP:MEA:ALGORithms?]
Returns current value of Message Encryption Algorithm Map (4 bit value) indexed by *n*. Range of *n* is 0 to 7.

FDTC:

FACCH: or SACCH:

MAP:

MEK?
[FDTC:FACCH: or SACCH:MAP:MEK?]
Returns current value of Message Encryption Key Map (4 bit value).

ARQ?
[FDTC:FACCH: or SACCH:MAP:ARQ?]
Returns current state of FACCH/SACCH ARQ Map (1 bit value).

SMS?
[FDTC:FACCH: or SACCH:MAP:SMS?]
Returns current value of SMS Map (2 bit value).

SUPPort:

IRA?
[FDTC:FACCH: or SACCH:SUPPort:IRA?]
Returns current state of RA Support (1 bit value).

SOC?
[FDTC:FACCH: or SACCH:SOC?]
Returns current value of SOC (12 bit value).

BSMC?
[FDTC:FACCH: or SACCH:BSMC?]
Returns current value of BSMC (8 bit value).

RTRANSaction?
[FDTC:FACCH: or SACCH:RTRANSaction?]
Returns current value of R-Transaction Identifier (8 bit value).

RDATA_UNIT:

LENGth?
[FDTC:FACCH: or SACCH:RDATA_UNIT:LENGth?]
Returns current value of R-Data Unit Extended Remaining Length (8 bit value).

HLP:

IDentifier?
[FDTC:FACCH: or SACCH:RDATA_UNIT:HLP:IDentifier?]
Returns current value of R-Data Unit Higher Layer Protocol Identifier (8 bit value).

DATA? *n*
[FDTC:FACCH: or SACCH:RDATA_UNIT:HLP:DATA?]
Returns current value of R-Data Unit Higher Layer Data Unit (8 bit value) indexed by *n*. Range of *n* is 0 to 255.

FDTC:

FACCH: or SACCH:

MESSage:

CENTer:

LENGth?
[FDTC:FACCH: or SACCH:MESSage:CENTer:LENGth?]
Returns current value of Message Center Address Extended Remaining Length (8 bit value).

TYPE?
[FDTC:FACCH: or SACCH:MESSage:CENTer:TYPE?]
Returns current value of Message Center Address Type of Number (3 bit value).

PLANid?
[FDTC:FACCH: or SACCH:MESSage:CENTer:PLANid?]
Returns current value of Message Center Address Number Plan Identification (4 bit value).

ENCoding?
[FDTC:FACCH: or SACCH:MESSage:CENTer:ENCoding?]
Returns current state of Message Center Address Encoding (1 bit value).

ADDRess?
[FDTC:FACCH: or SACCH:MESSage:CENTer:ADDRess?]
Returns current string value of Message Center Address (ASCII string).

USER:

DEST:

LENGth?
[FDTC:FACCH: or SACCH:USER:DEST:LENGth?]
Returns current value of User Destination Address Extended Remaining Length (8 bit value).

TYPE?
[FDTC:FACCH: or SACCH:USER:DEST:TYPE?]
Returns current value of User Destination Address Type of Number (3 bit value).

PLANid?
[FDTC:FACCH: or SACCH:USER:DEST:PLANid?]
Returns current value of User Destination Address Number Plan Identification (4 bit value).

ENCoding?
[FDTC:FACCH: or SACCH:USER:DEST:ENCoding?]
Returns current state of User Destination Address Encoding (1 bit value).

FDTC:

FACCH: or SACCH:

USER:

DEST:

ADDRess?
[FDTC:FACCH: or SACCH:USER:DEST:ADDRess?]
Returns current string value of User Destination Address (ASCII string).

SUBaddress:

LENGth?
[FDTC:FACCH: or SACCH:USER:DEST:SUBaddress:LENGth?]
Returns current value of User Destination Subaddress Extended Remaining length (8 bit value).

ODD_EVEN?
[FDTC:FACCH: or SACCH:USER:DEST:SUBaddress:ODD_EVEN?]
Returns current state of User Destination Subaddress Odd/Even Indicator (1 bit value).

TYPE?
[FDTC:FACCH: or SACCH:USER:DEST:SUBaddress:TYPE?]
Returns current value of User Destination Type of Subaddress (3 bit value).

REServed?
[FDTC:FACCH: or SACCH:USER:DEST:SUBaddress:REServed?]
Returns current value of User Destination Subaddress Reserved bits (4 bit value).

ADDRess? *n*
[FDTC:FACCH: or SACCH:USER:DEST:SUBaddress:ADDRess?]
Returns current value of User Destination Subaddress (8 bit value) indexed by *n*. Range of *n* is 0 to 19.

ORIG:

LENGth?
[FDTC:FACCH: or SACCH:USER:ORIG:LENGth?]
Returns current value of User Originating Address Extended Remaining Length (8 bit value).

TYPE?
[FDTC:FACCH: or SACCH:USER:ORIG:TYPE?]
Returns current value of User Originating Address Type of Number (3 bit value).

PLANid?
[FDTC:FACCH: or SACCH:USER:ORIG:PLANid?]
Returns current value of User Originating Address Number Plan Identification (4 bit value).

FDTC:

FACCH: or SACCH:

USER:

ORIG:

ENCoding?
[FDTC:FACCH: or SACCH:USER:ORIG:ENCoding?]
Returns current state of User Originating Address Encoding (1 bit value).

ADDRess?
[FDTC:FACCH: or SACCH:USER:ORIG:ADDRess?]
Returns current value of User Originating Address (ASCII string).

SUBaddress:

LENGth?
[FDTC:FACCH: or SACCH:USER:ORIG:SUBaddress:LENGth?]
Returns current value of User Originating Subaddress Extended Remaining length (8 bit value).

ODD_EVEN?
[FDTC:FACCH: or SACCH:USER:ORIG:SUBaddress:ODD_EVEN?]
Returns current state of User Originating Subaddress Odd/Even Indicator (1 bit value).

TYPE?
[FDTC:FACCH: or SACCH:USER:ORIG:SUBaddress:TYPE?]
Returns current value of User Originating Type of Subaddress (3 bit value).

REServed?
[FDTC:FACCH: or SACCH:USER:ORIG:SUBaddress:REServed?]
Returns current value of User Originating Subaddress Reserved bits (4 bit value).

ADDRess? *n*
[FDTC:FACCH: or SACCH:USER:ORIG:SUBaddress:ADDRess?]
Returns current value of User Originating Subaddress (8 bit value) indexed by *n*. Range of *n* is 0 to 19.

PRESentation:

LENGth?
[FDTC:FACCH: or SACCH:USER:ORIG:LENGth?]
Returns current value of User Originating Address Presentation Indicator Extended Remaining Length (8 bit value).

PI?
[FDTC:FACCH: or SACCH:USER:ORIG:PI?]
Returns current value of User Originating Address Presentation Indicator (2 bit value).

FDTC:

FACCH: or SACCH:

USER:

ORIG:

SI?
[FDTC:FACCH: or SACCH:USER:ORIG:SI?]
Returns current value of User Originating Address Screening Indicator (2 bit value).

REServed?
[FDTC:FACCH: or SACCH:USER:ORIG:REServed?]
Returns current value of User Originating Address Presentation Indicator reserved bits (4 bit value).

RCAUSE?
[FDTC:FACCH: or SACCH:RCAUSE?]
Returns current value of R-Cause (8 bit value).

CUSTOM:

LENGth?
[FDTC:FACCH: or SACCH:CUSTOM:LENGth?]
Returns current value of Length of Custom Control in Octets (8 bit value).

CONTrol? *n*
[FDTC:FACCH: or SACCH:CUSTOM:CONTrol?]
Returns current value of Custom Control (8 bit value) indexed by *n*. Range of *n* is 0 to 255.

HYPERband:

NUMBer?
[FDTC:FACCH: or SACCH:HYPERband:NUMBer?]
Returns current value of RF Channel and Hyperband, Number of Hyperband channels (5 bit value).

BAND? *n*
[FDTC:FACCH: or SACCH:HYPERband:BAND?]
Returns current value of RF Channel and Hyperband, Hyperband (2 bit value) indexed by *n*. Range of *n* is 0 to 23.

CHANnel? *n*
[FDTC:FACCH: or SACCH:HYPERband:CHANnel?]
Returns current value of RF Channel and Hyperband, Channel (11 bit value) indexed by *n*. Range of *n* is 0 to 23.

TARGet?
[FDTC:FACCH: or SACCH:HYPERband:TARGet?]
Returns current value of Target Hyperband (2 bit value).

7-6 REVERSE CONTROL CHANNEL (RECC) MONITOR COMMANDS

7-6-1 REVERSE CONTROL CHANNEL CONTROL

RECC:

SETup
Sets up the FM/AM-1600CSA as when entering the Reverse Control Channel Screen. (Screen is not displayed.)

STARt
Starts monitoring Reverse Control Channel data. (Data returned by query. Refer to 7-6-2.)

STOP
Stops monitoring Reverse Control Channel data.

CHANnel *n*
Selects Reverse Control Channel to monitor. Range of *n* is 1 to 1023 (800 to 900 not used).

7-6-2 REVERSE CONTROL CHANNEL DECODE DATA

RECC:

AUTHR?
Returns AUTHR from Mobile Station Authentication Algorithms (18 bit value).

AUTHU?
Returns AUTHU used in Unique Challenge Order Confirmation (18 bit value).

COUNt?
Returns modulo-64 count sent from Mobile Station.

DCC?
Returns Digital Color Code (seven bit value).

DIGITS1?
Returns up to 16 digits (string).

DIGITS2?
Returns up to 16 digits (string).

E?
Returns Extended Address (one bit value).

EP?
Returns Extended Protocol (one bit value).

ER?
Returns Extended Protocol Reverse Channel Indicator (one bit value).

ESN?
Returns Electronic Serial Number (32 bit value).

ADDENDUM TO SECTION 7-6

7-6A RECC

RECC:

CONFigure:

USER
[RECC:CONFigure:USER]
This command is identical to the **RECC:SETup** command except that it selects the USER screen.

NONE
[RECC:CONFigure:NONE]
This command is also identical to the **RECC:SETup** command except that it does not select a screen.

SERVice?
[RECC:SERVice?]
Returns current value of Service code (4 bit value).

PM_D?
[RECC:PM_D?]
Returns current value of Selected Privacy Mode (3 bit value).

SAP?
[RECC:SAP?]
Returns current state of Service Access Point for data/fax call (1 bit value).

CRC?
[RECC:CRC?]
Returns 16 or 24 bit CRC setting on data/fax call (1 bit value).

DATA:

PART?
[RECC:DATA:PART?]
Returns current value of Data Part (3 bit value).

ACKED?
[RECC:DATA:ACKED?]
Returns current state of Acked Data (1 bit value).

RLP?
[RECC:RLP?]
Returns current value of Layer 2 Radio Link Protocol used for a data/fax call (2 bits value).

TORDER?
[RECC:TORDER?]
The new orders for RECC are:

ORIGIN W/SERVICE	PAGE RESP W/SERVICE

RECC:

LOCAL_MT?
Returns Local Control (Local Control message)/Message Type (five bit value).

LT?
Returns Last Try (one bit value).

MIN?
Returns Mobile Identification Number (string) of Mobile Station.

MPCI?
Returns Mobile Protocol Capability Indicator (one bit value).

ORDERCD?
Returns Order Code (five bit value).

ORDQ?
Returns Order Qualifier (three bit value).

RANDBS?
Returns RANDBS used in Base Station Challenge Order (32 bit value).

RANDC?
Returns RANDC (eight bit value).

S?
Returns Serial Number (one bit value).

SCM?
Returns Station Class Mark (four bit value).

SDCC1?
Returns Supplementary Digital Color Code 1 (two bit value).

SDCC2?
Returns Supplementary Digital Color Code 2 (two bit value).

TORDer?
Returns combination of the T and Order Fields (string):

A ALERT CNF	ORIGIN	REG
AUDIT CNF	PAGE RESP	

7-7 REVERSE VOICE CHANNEL (RVC) MONITOR COMMANDS

7-7-1 REVERSE VOICE CHANNEL CONTROL

RVC:

SETup
Sets up the FM/AM-1600CSA as when entering the Reverse Voice Channel Screen. (Screen is not displayed.)

STARt
Starts decoding Reverse Voice Channel data. (Data returned by query. Refer to 7-7-2.)

STOP
Stops decoding Reverse Voice Channel data.

CHANnel *n*
Selects Reverse Voice Channel to monitor. Range of *n* is 1 to 1023 (800 to 900 not used).

7-7-2 REVERSE VOICE CHANNEL DECODE DATA

RVC:

AUTHu?
Returns AUTHU used in Unique Challenge Order Confirmation (18 bit value).

DIGITS?
Returns up to 32 digits (string).

ESN?
Returns Electronic Serial Number (32 bit value).

LOCAL_MT?
Returns Local Control (Local Control message)/Message Type (five bit value).

ORDERCD?
Returns Order Code (five bit value).

ORDQ?
Returns Order Qualifier (three bit value).

RANDbs?
Returns RANDBS used in Base Station Challenge Order (32 bit value).

TORDer?
Returns combination of the T and Order Fields (string):

CALLED ADDR	PAGE RESP
ORDER CNF	SNR RES

ADDENDUM TO SECTION 7-7

7-7A RVC

RVC:

CONFigure:

USER
[RVC:CONFigure:USER]
This command is identical to the **RVC:SETup** command except that it selects the USER screen.

NONE
[RVC:CONFigure:NONE]
This command is also identical to the **RVC:SETup** command except that it does not select a screen.

TORDER?
[RVC:TORDER?]
New order for RVC is:

RELEASE

7-8 REVERSE DIGITAL TRAFFIC CHANNEL (RDTC) MONITOR COMMANDS

7-8-1 REVERSE DIGITAL TRAFFIC CHANNEL CONTROL

RDTC:

SETup
Sets up the FM/AM-1600CSA as when entering the Reverse Digital Traffic Channel Screen. (Screen is not displayed.)

STARt
Starts decoding Reverse Digital Traffic Channel data. (Data returned by query. Refer to 7-8-2.)

STOP
Stops decoding Reverse Digital Traffic Channel data.

CHANnel *n*
Selects Reverse Digital Traffic Channel to monitor. Range of *n* is 1 to 1023 (800 to 900 not used).

SLOT *n*
Selects Timeslot. Range of *n* is 1 to 3.

REMote:

STARt
Monitoring stops and VSELP data is redirected as ASCII characters out RS-232 Connector. (Baud Rate should be set to 38400 prior to command execution to allow RS-232 to keep up with data received.) Each line contains 54 ASCII Characters representing 27 bytes, followed by a carriage return. Refer to Table 7-1.

STOP
Stops redirection of VSELP data out RS-232 Connector.

R0?
Returns the VSELP frame energy value (0 to 31).

Byte	Description	Byte	Description	Byte	Description
R0	Frame Energy	LPC9	9th Reflection Coefficient	CODE1_4	1st Code,I,4th Subframe
LPC1	1st Reflection Coefficient	LPC10	10th Reflection Coefficient	CODE2_1	2nd Code,H,1st Subframe
LPC2	2nd Reflection Coefficient	LAG_1	Lag,1st Subframe	CODE2_2	2nd Code,H,2nd Subframe
LPC3	3rd Reflection Coefficient	LAG_2	Lag,2nd Subframe	CODE2_3	2nd Code,H,3rd Subframe
LPC4	4th Reflection Coefficient	LAG_3	Lag,3rd Subframe	CODE2_4	2nd Code,H,4th Subframe
LPC5	5th Reflection Coefficient	LAG_4	Lag,4th Subframe	GSP0_1	GSP0P1 Code,1st Subframe
LPC6	6th Reflection Coefficient	CODE1_1	1st Code,I,1st Subframe	GSP0_2	GSP0P1 Code,2nd Subframe
LPC7	7th Reflection Coefficient	CODE1_2	1st Code,I,2nd Subframe	GSP0_3	GSP0P1 Code,3rd Subframe
LPC8	8th Reflection Coefficient	CODE1_3	1st Code,I,3rd Subframe	GSP0_4	GSP0P1 Code,4th Subframe

Table 7-1 Speech Parameter (27 bytes)

7-8-2 REVERSE DIGITAL TRAFFIC CHANNEL DECODE DATA

RDTC:FACCH: queries pertain to the blank and burst Fast Associated Control Channel (FACCH).
RDTC:SACCH: queries pertain to the continuous Slow Associated Control Channel (SACCH).

RDTC:

FACCH:

AMT?
Returns Acknowledge Message Type (string):

ALERT	LC	RELEASE
AUDIT	MAINTNC	STOP MEAS
HANDOFF		

AUTHU?
Returns AUTHU used in Unique Challenge Order Confirmation (18 bit value).

BER?
Returns Bit Error Rate (three bit value). Refer to Table 7-2.

Bit Value	BER Interval (%)	Bit Value	BER Interval (%)
0	0.00 to 0.01	4	1.0 to 2.0
1	0.01 to 0.1	5	2.0 to 4.0
2	0.1 to 0.5	6	4.0 to 8.0
3	0.5 to 1.0	7	≥8.0

Table 7-2 BER Bit Definition

CALLED:

NUM?
Returns number of Called Party (string).

PLANid?
Returns Called Party Numbering Plan Identification (four bit value).

SPare?
Returns value of Called Party Number spare bit (one bit value).

TYpe?
Returns Called Party Number Type (three bit value).

CALLING:

NUM?
Returns number of Calling Party (string).

PLANid?
Returns Calling Party Numbering Plan Identification (four bit value).

SPare?
Returns value of Calling Party Number spare bits (five bit value).

TYpe?
Returns Calling Party Number Type (three bit value).

RDTC:

FACCH:

CM?
Returns Call Mode (five bit value).

DIC?
Returns the Delay Interval Compensation (one bit value).

DIGits?
Returns up to 32 digits (string).

DMAC?
Returns Digital Mobile Attenuation Code (four bit value).

DTX?
Returns Discontinuous Transmission (one bit value).

ESN?
Returns Electronic Serial Number (32 bit value).

FI?
Returns Feature Indicator (four bit value).

KF?
Returns Keypad Facility (32 digits).

LDP?
Returns Last Decoded Parameter (four bit value).

MEM?
Returns Memory Encryption Mode (one bit value).

MSGtype?
Returns Message Type received from Mobile Station (string):

CHAN QUAL1	MEAS ACK	RELEASE
CHAN QUAL2	MOBILE ACK	SBDTMF
CONNECT	PLC ACK	SCDTMF
FLASH	PU ACK	STATUS
FLASH ACK		

NV?
Returns Numbers of Values (string of numbers, one for each optional information element in the message, with each number derived from the corresponding six bit value).

PD?
Returns the Protocol Discriminator (two bit value).

PT?
Returns Parameter Types (string of numbers, one for each optional information element in the message, with each number derived from the corresponding six bit value).

RANDBS?
Returns RANDBS used in the Base Station Challenge Order (32 bit value).

RDTC:

FACCH:

RFCHAN? *n*
Returns RF Channel (1 to 1023) for the selected index. Range of *n* is 0 to 11.

RL?
Returns Remaining Length (six bit value).

RN?
Returns Request Number (four bit value).

RR?
Returns Release Reason (four bit value).

RSSI? *n*
Returns Received Signal Strength Indicator (five bit value) for the selected index. (0 indicates -113 dBm and 31 indicates -51 dBm.) Range of *n* (index) is 0 to 11.

RSSIC?
Returns Received Signal Strength Indicator (five bit value) of Current RF Channel. (0 indicates -113 dBm and 31 indicates -51 dBm.)

SSDUP?
Returns Shared Secret Data Update (one bit value).

TA?
Returns Time Alignment offset (five bit value).

TERMInf?
Returns Terminal Information of Mobile Station (32 bit value).

VPM?
Returns Voice Privacy Mode (one bit value).

SACCH:

AMT?
Returns Acknowledge Message Type (string):

ALERT	LC	RELEASE
AUDIT	MAINTNC	STOP MEAS
HANDOFF		

AUTHU?
Returns AUTHU used in Unique Challenge Order Confirmation (18 bit value).

BER?
Returns Bit Error Rate (three bit value). Refer to Table 7-2.

CALLED:

NUM?
Returns number of Called Party (string).

PLANid?
Returns Called Party Numbering Plan Identification (four bit value).

RDTC:

SACCH:

CALLED:

SPare?
Returns value of Called Party Number spare bit (one bit value).

TYpe?
Returns Called Party Number Type (three bit value).

CALLING:

NUM?
Returns number of Calling Party (string).

PLANid?
Returns Calling Party Numbering Plan Identification (four bit value).

SPare?
Returns value of Calling Party Number spare bits (five bit value).

TYpe?
Returns Calling Party Number Type (three bit value).

CM?
Returns Call Mode (five bit value).

DIC?
Returns Delay Interval Compensation (one bit value).

DIGits?
Returns up to 32 digits (string).

DMAC?
Returns Digital Mobile Attenuation Code (four bit value).

DTX?
Returns Discontinuous Transmission (one bit value).

ESN?
Returns Electronic Serial Number (32 bit value).

FI?
Returns Feature Indicator (four bit value).

KF?
Returns Keypad Facility (32 digits).

LDP?
Returns Last Decoded Parameter (four bit value).

MEM?
Returns Memory Encryption Mode (one bit value).

RDTC:

SACCH:

MSGtype?
Returns Message Type received from Mobile Station (string):

CHAN QUAL1	MEAS ACK	RELEASE
CHAN QUAL2	MOBILE ACK	SBDTMF
CONNECT	PLC ACK	SCDTMF
FLASH	PU ACK	STATUS
FLASH ACK		

NV?
Returns Numbers of Values (string of numbers, one for each optional information element in the message, with each number derived from the corresponding six bit value).

PD?
Returns the Protocol Discriminator (two bit value).

PT?
Returns Parameter Types (string of numbers, one for each optional information element in the message, with each number derived from the corresponding six bit value).

RANDBS?
Returns RANDBS used in the Base Station Challenge Order (32 bit value).

RFCHAN? *n*
Returns RF Channel (1 to 1023) for the selected index. Range of *n* is 0 to 11.

RL?
Returns Remaining Length (six bit value).

RN?
Returns Request Number (four bit value).

RR?
Returns Release Reason (four bit value).

RSSI? *n*
Returns Received Signal Strength Indicator (five bit value) for selected index. (0 indicates -113 dBm and 31 indicates -51 dBm.) Range of *n* (index) is 0 to 11.

RSSIC?
Returns Received Signal Strength Indicator (five bit value) of Current RF Channel. (0 indicates -113 dBm and 31 indicates -51 dBm.)

SSDUP?
Returns Shared Secret Data Update (one bit value).

TA?
Returns Time Alignment offset (five bit value).

TERMinf?
Returns Terminal Information of Mobile Station (32 bit value).

VPM?
Returns Voice Privacy Mode (one bit value).

ADDENDUM TO SECTION 7-8

7-8A RDTC

RDTC:

CONFigure:

USER
[RDTC:CONFigure:USER]
This command is identical to the **RDTC:SETup** command except that it selects the USER screen.

NONE
[RDTC:CONFigure:NONE]
This command is also identical to the **RDTC:SETup** command except that it does not select a screen.

FACCH: or SACCH:

MSGtype?
[FACCH: or SACCH:MSGtype?]
Add the following message types.

CHAN QUAL3	SOC
CHAN QUAL4	BSMC
HYPER MEAS ACK	R-DATA
CAPABILITY RESP	R-DATA REJECT
CAPABILITY REQ	R-DATA ACCEPT

RFCHAN? *n*
[FACCH: or SACCH:RFCHAN?]
Returns current value of RFCHAN (8 bit value) indexed by *n*. Range of *n* is 0 to 23.

RSSI? *n*
[FACCH: or SACCH:RSSI?]
Returns current value of RSSI (8 bit value) indexed by *n*. Range of *n* is 0 to 23.

AMT?
[FACCH: or SACCH:AMT?]
Add the following message types.

DTC HANDOFF

TASK?
[FACCH: or SACCH:TASK?]
Returns current value of Task Status (3 bit value).

RDTC:

FACCH: or SACCH:

RTRANSaction?
[RDTC:FACCH: or SACCH:RTRANSaction?]
Returns current value of R-Transaction Identifier (8 bit value).

RDATA_UNIT:

LENGth?
[RDTC:FACCH: or SACCH:RDATA_UNIT:LENGth?]
Returns current value of R-Data Unit Extended Remaining Length (8 bit value).

HLP:

IDentifier?
[RDTC:FACCH: or SACCH:RDATA_UNIT:HLP:IDentifier?]
Returns current value of R-Data Unit Higher Layer Protocol Identifier (8 bit value).

DATA? ***n***
[RDTC:FACCH: or SACCH:RDATA_UNIT:HLP:DATA?]
Returns current value of R-Data Unit Higher Layer Protocol Data Unit (8 bit value) indexed by *n*. Range of *n* is 0 to 254.

MESSage:

CENTer:

LENGth?
[RDTC:FACCH: or SACCH:MESSage:CENTer:LENGth?]
Returns current value of Message Center Address Extended Remaining Length (8 bit value).

TYPE?
[RDTC:FACCH: or SACCH:MESSage:CENTer:TYPE?]
Returns current value of Message Center Address Type of Number (3 bit value).

PLANid?
[RDTC:FACCH: or SACCH:MESSage:CENTer:PLANid?]
Returns current value of Message Center Address Number Plan Identification (4 bit value).

ENCoding?
[RDTC:FACCH: or SACCH:MESSage:CENTer:ENCoding?]
Returns current state of Message Center Address Encoding (1 bit value).

ADDRess?
[RDTC:FACCH: or SACCH:MESSage:CENTer:ADDRess?]
Returns current string value of Message Center Address (ASCII string).

RDTC:

FACCH: or SACCH:

USER:

DEST:

LENGth?
[RDTC:FACCH: or SACCH:USER:DEST:LENGth?]
Returns current value of User Destination Address Extended Remaining Length (8 bit value).

TYPE?
[RDTC:FACCH: or SACCH:USER:DEST:TYPE?]
Returns current value of User Destination Address Type of Number (3 bit value).

PLANid?
[RDTC:FACCH: or SACCH:USER:DEST:PLANid?]
Returns current value of User Destination Address Number Plan Identification (4 bit value).

ENCoding?
[RDTC:FACCH: or SACCH:USER:DEST:ENCoding?]
Returns current state of User Destination Address Encoding (1 bit value).

ADDRess?
[RDTC:FACCH: or SACCH:USER:DEST:ADDRess?]
Returns current string value of User Destination Address (ASCII string).

SUBaddress:

LENGth?
[RDTC:FACCH: or SACCH:USER:DEST:SUBaddress:LENGth?]
Returns current value of User Destination Subaddress Extended Remaining length (8 bit value).

ODD_EVEN?
[RDTC:FACCH: or SACCH:USER:DEST:SUBaddress:ODD_EVEN?]
Returns current state of User Destination Subaddress Odd/Even Indicator (1 bit value).

TYPE?
[RDTC:FACCH: or SACCH:USER:DEST:SUBaddress:TYPE?]
Returns current value of User Destination Type of Subaddress (3 bit value).

REServed?
[RDTC:FACCH: or SACCH:USER:DEST:SUBaddress:REServed?]
Returns current value of User Destination Subaddress Reserved bits (4 bit value).

RDTC:

FACCH: or SACCH:

USER:

DEST:

SUBaddress:

ADDRess? *n*
[RDTC:FACCH: or SACCH:USER:DEST:SUBaddress:ADDRess?]
Returns current value of User Destination Subaddress (8 bit value) indexed by *n*. Range of *n* is 0 to 19.

ORIG:

LENGth?
[RDTC:FACCH: or SACCH:USER:ORIG:LENGth?]
Returns current value of User Originating Address Extended Remaining Length (8 bit value).

TYPE?
[RDTC:FACCH: or SACCH:USER:ORIG:TYPE?]
Returns current value of User Originating Address Type of Number (3 bit value).

PLANid?
[RDTC:FACCH: or SACCH:USER:ORIG:PLANid?]
Returns current value of User Originating Address Number Plan Identification (4 bit value).

ENCoding?
[RDTC:FACCH: or SACCH:USER:ORIG:ENCoding?]
Returns current state of User Originating Address Encoding (1 bit value).

ADDRess?
[RDTC:FACCH: or SACCH:USER:ORIG:ADDRess?]
Returns current string value of User Originating Address (ASCII string).

SUBaddress:

LENGth?
[RDTC:FACCH: or SACCH:USER:ORIG:SUBaddress:LENGth?]
Returns current value of User Originating Subaddress Extended Remaining length (8 bit value).

ODD_EVEN?
[RDTC:FACCH: or SACCH:USER:ORIG:SUBaddress:ODD_EVEN?]
Returns current state of User Originating Subaddress Odd/Even Indicator (1 bit value).

RDTC:

FACCH: or SACCH:

USER:

ORIG:

SUBaddress:

TYPE?
[RDTC:FACCH: or SACCH:USER:ORIG:SUBaddress:TYPE?]
Returns current value of User Originating Type of Subaddress (3 bit value).

REServed?
[RDTC:FACCH: or SACCH:USER:ORIG:SUBaddress:REServed?]
Returns current value of User Originating Subaddress Reserved bits (4 bit value).

ADDRess? *n*
[RDTC:FACCH: or SACCH:USER:ORIG:SUBaddress:ADDRess?]
Returns current value of User Originating Subaddress (8 bit value) indexed by *n*. Range of *n* is 0 to 19.

PRESentation:

LENGth?
[RDTC:FACCH: or SACCH:USER:ORIG:PRESentation:LENGth?]
Returns current value of User Originating Address Presentation Indicator Extended Remaining Length (8 bit value).

PI?
[RDTC:FACCH: or SACCH:USER:ORIG:PRESentation:PI?]
Returns current value of User Originating Address Presentation Indicator (2 bit value).

SI?
[RDTC:FACCH: or SACCH:USER:ORIG:PRESentation:SI?]
Returns current value of User Originating Address Screening Indicator (2 bit value).

REServed?
[RDTC:FACCH: or SACCH:USER:ORIG:PRESentation:REServed?]
Returns current value of User Originating Address Presentation Indicator reserved bits (4 bit value).

RCAUSE?
[RDTC:FACCH: or SACCH:RCAUSE?]
Returns current value of R-Cause (8 bit value).

PV?
[RDTC:FACCH: or SACCH:PV?]
Returns current value of Protocol Version (8 bit value).

RDTC:

FACCH: or SACCH:

MAP:

VPM?
[RDTC:FACCH: or SACCH:MAP:VPM?]
Returns current value of Voice Privacy Mode Map (4 bit value).

CODER?
[RDTC:FACCH: or SACCH:MAP:CODER?]
Returns current value of Voice Coder Map (6 bit value).

MEA:

DOMAIN?
[RDTC:FACCH: or SACCH:MAP:MEA:DOMAIN?]
Returns current value of Message Encryption Algorithm map domain (8 bit value).

ALGORithms? *n*
[RDTC:FACCH: or SACCH:MAP:MEA:ALGORithms?]
Returns current value of Message Encryption Algorithms (4 bit value) indexed by *n*. Range of *n* is 0 to 7.

MEK?
[RDTC:FACCH: or SACCH:MAP:MEK?]
Returns current value of Message Encryption Key Map (4 bit value).

ARQ?
[RDTC:FACCH: or SACCH:MAP:ARQ?]
Returns current state of FACCH/SACCH ARQ Map (1 bit value).

SMS?
[RDTC:FACCH: or SACCH:MAP:SMS?]
Returns current value of SMS Map (2 bit value).

SOC?
[RDTC:FACCH: or SACCH:SOC?]
Returns current value of SOC (12 bit value).

BSMC?
[RDTC:FACCH: or SACCH:BSMC?]
Returns current value of BSMC (8 bit value).

RDTC:

FACCH: or SACCH:

SUPPort:

IRA?
[RDTC:FACCH: or SACCH:SUPPort:IRA?]
Returns current state of IRA Support (1 bit value).

FREQuency:

BANDS?
[RDTC:FACCH: or SACCH:SUPPort:FREQuency:BANDS?]
Returns current value of Supported Frequency Bands (8 bit value).

ANAlog?
[RDTC:FACCH: or SACCH:SUPPort:ANAlog?]
Returns current state of 800 MHz Analog Speech Support (1 bit value).

CUSTOM:

LENGth?
[RDTC:FACCH: or SACCH:CUSTOM:LENGth?]
Returns current value of Length of custom control in octets (8 bit value).

CONTrol? *n*
[RDTC:FACCH: or SACCH:CUSTOM:CONTrol?]
Returns current value of Custom Control (8 bit value) indexed by *n*. Range of *n* is 0 to 255.

HYPERband:

NUMBer?
[RDTC:FACCH: or SACCH:HYPERband:NUMBer?]
Returns current value of Number of Hyperband Channels (5 bit value).

BAND? *n*
[RDTC:FACCH: or SACCH:HYPERband:BAND?]
Returns current value of Hyperband band (2 bit value) indexed by *n*. Range of *n* is 0 to 23.

CHANnel? *n*
[RDTC:FACCH: or SACCH:HYPERband:CHANnel?]
Returns current value of Hyperband Channel (11 bit value) indexed by *n*. Range of *n* is 0 to 23.

7-9 CELL SITE SIMULATION COMMANDS

Refer to 9-8 for command guidelines and programming examples.

7-9-1 GENERAL

CSS:

SETup
Sets up the FM/AM-1600CSA as when entering the first Cell Site Simulation screen. (Screen is not displayed and the FM/AM-1600S is not transmitting at this point.)

STARt
Starts Cell Site Simulation transmission (Overhead Message on FOCC).

STOP
Stops Cell Site Simulation transmission.

CHANnel *n*
Selects Forward Control Channel for sending Overhead Messages. Range of *n* is 1 to 1023.

RFLvl *n*
Sets RF Output Level, in dBm, of Overhead Message. Range of *n* is -30.0 to -127.0.

7-9-2 FOCC OVERHEAD MESSAGE PARAMETERS

CSS:

FOCC:

AUTH *n*
Sets Authentication bit (1 or 0).

AUTH?
Returns Authentication bit setting.

B_I *n*
Sets Busy-Idle bit (1 or 0).

B_I?
Returns Busy-Idle bit setting.

CMAC *n*
Sets Control Mobile Attenuation Code. Range of *n* is 0 to 7.

CMAC?
Returns Control Mobile Attenuation Code setting.

CMAX *n*
Sets maximum Number of Channels to be scanned by Mobile Station when accessing a system. Range of *n* is 1 to 32.

CMAX?
Returns maximum Number of Channels to be scanned by Mobile Station setting.

CSS:

FOCC:

DCC *n*
Sets Digital Color Code. Range of *n* is 0 to 3.

DCC?
Returns Digital Color Code setting.

E *n*
Sets Extended Address bit (1 or 0).

E?
Returns Extended Address bit setting.

EP *n*
Sets Extended Protocol bit (1 or 0).

EP?
Returns Extended Protocol bit setting.

N *n*
Sets Number of Paging Channels to be scanned by Mobile Station. Range of *n* is 1 to 32.

N?
Returns Number of Paging Channels to be scanned by Mobile Station setting.

PCI *n*
Sets Protocol Capability Indicator (1 or 0).

PCI?
Returns Protocol Capability Indicator setting.

RCF *n*
Sets Read Control Filler bit (1 or 0).

RCF?
Returns Read Control Filler bit setting.

REGH *n*
Sets Home Registration bit (1 or 0).

REGH?
Returns Home Registration bit setting.

REGR *n*
Sets Roaming Registration bit (1 or 0).

REGR?
Returns Roaming Registration bit setting.

CSS:

FOCC:

S *n*
Sets Serial Number bit (1 or 0).

S?
Returns Serial Number bit setting.

SDCC1 *n*
Sets Supplementary Digital Color Code 1. Range of *n* is 0 to 3.

SDCC1?
Returns Supplementary Digital Color Code 1 setting.

SDCC2 *n*
Sets Supplementary Digital Color Code 2. Range of *n* is 0 to 3.

SDCC2?
Returns Supplementary Digital Color Code 2 setting.

SID *n*
Sets System Identification Number (14 most significant digits). Range of *n* is 0 to 32767.

SID?
Returns System Identification Number setting.

WFOM *n*
Sets Wait For Overhead Message bit (1 or 0).

WFOM?
Returns Wait For Overhead Message bit setting.

7-9-3 CALLING SETUP

CSS:

CALL:

CHANnel *n*
Selects Digital Traffic Channel or Voice Channel the call to the Mobile Station is assigned to. Range of *n* is 1 to 1023.

CHANnel?
Returns Digital Traffic Channel or Voice Channel the call to the Mobile Station is assigned to (1 to 1023).

DEViation *n*
Sets SAT Deviation in kHz. Range of *n* is 0.0 to 4.0.

DEViation?
Returns SAT Deviation in kHz setting.

DMAC *n*
Selects Digital Mobile Attenuation Code. Range of *n* is 0 to 7.

DMAC?
Returns Digital Mobile Attenuation Code setting.

DVCC *n*
Selects Digital Verification Color Code. Range of *n* is 0 to 255.

DVCC?
Returns Digital Verification Color Code setting.

EF *n*
Sets Extended Protocol Forward Channel Indicator (1 or 0).

EF?
Returns Extended Protocol Forward Channel Indicator setting.

MEM *n*
Sets Message Encryption Mode (1 or 0).

MEM?
Returns Message Encryption Mode setting.

MIN *"n"*
Selects Mobile Identification Number to call. The Mobile Identification Number (*n*) is entered as a string, with quotation marks.

Example: `CSS:CALL:MIN "316/522-4981"`

MIN?
Returns MIN currently selected.

PM *n*
Sets Privacy Mode bit (1 or 0).

CSS:

CALL:

PM?
Returns Privacy Mode bit setting.

SAT *n*
Sets Supervisory Audio Tone in Hz. Range of *n* is 5965 to 6035.

SAT?
Returns Supervisory Audio Tone setting.

SLOT *n*
Selects Timeslot. Range of *n* is 1 to 3.

SLOT?
Returns Timeslot setting.

TYPE *n*
Selects channel type (1 [Digital] or 0 [Analog]).

TYPE?
Returns channel type.

VMAC *n*
Selects Voice Mobile Attenuation Code. Range of *n* is 0 to 7.

VMAC?
Returns Voice Mobile Attenuation Code setting.

7-9-4 PROCESS AND HANDOFF

CSS:

CALL:

PROCess:

ASSIGNment
Sends an initial channel designation command to the Mobile Station. After sending the channel designation, the FM/AM-1600CSA changes the channel and mode to transmit and receive on the assigned channel.

MOBINIT
Sets up the FM/AM-1600CSA for a call initiated by the Mobile Station. The Busy-Idle bit changes from a 1 to a 0 at the correct time to enable the access to occur. An access sends an SRQ=1 on the GPIB (if **GPIB:MASK 1** had been initiated before the **CSS:CALL:PROCess:MOBINIT**) and sets the status bit returned by the **CSS:RECC:STATus?** query.

PAGE
Simulates a Base Station (Cellsite) initiated call.

REGistration
Causes the Mobile Station to send a Registration message. An access by the Mobile Station sends an SRQ=1 on the GPIB (if **GPIB:MASK 1** had been initiated before the **CSS:CALL:PROCess:REGistration**) and sets the status bit returned by the **CSS:RECC:STATus?** query. The **RECC:TORDer?** query verifies the registration order was or was not received.

FVC:

HANDoff
Sends an analog to analog handoff order to the Mobile Station. The **CSS:FVC:HAND:CHAN** *n* command specifies the new analog Voice Channel. After the handoff order, the FM/AM-1600CSA changes to the specified Voice Channel.

SLOT1
Sends an analog to digital handoff order (to Timeslot 1) to the Mobile Station. The **CSS:FVC:HAND:CHAN** *n* command specifies the new Digital Traffic Channel. After the handoff order, the FM/AM-1600CSA changes to the specified Digital Traffic Channel.

SLOT2
Sends an analog to digital handoff order (to Timeslot 2) to the Mobile Station. The **CSS:FVC:HAND:CHAN** *n* command specifies the new Digital Traffic Channel. After the handoff order, the FM/AM-1600CSA changes to the specified Digital Traffic Channel.

SLOT3
Sends an analog to digital handoff order (to Timeslot 3) to the Mobile Station. The **CSS:FVC:HAND:CHAN** *n* command specifies the new Digital Traffic Channel. After the handoff order, the FM/AM-1600CSA changes to the specified Digital Traffic Channel.

CSS:

CALL:

PROCess:

FDTC:

HANDoff?
Performs a digital to digital (**CSS:FDTC:TI?** ≠ 0), or digital to analog (**CSS:FDTC:TI?** = 0) handoff and returns a 1 if successful or a 0 if not successful. If successful, the FM/AM-1600CSA changes to the channel specified by the **CSS:FDTC:HAND:CHAN** *n* command.

RECC:

STATus?
Returns access with the Mobile Station status (0 indicates no access has occurred; 1 indicates an access has occurred).

7-9-5 FORWARD VOICE CHANNEL (FVC)

Data commands set the data in the messages and order commands send the orders (messages).

A. FVC ORDERS

CSS:

FVC:

ORDER:

ALERT
Sends the Alert message.

ALERTWinfo
Sends the Alert with Information message.

AUDIT
Sends the Audit order.

BSCHALCON
Sends the Base Station Challenge Confirmation message.

CALLMODEACK
Sends the Call Mode Acknowledgment message.

DISDTMF
Sends the Disable DTMF message.

DISMEM
Sends the Message Encryption Mode order with disable indication.

ENAMEM
Sends the Message Encryption Mode order with enable indication.

FLASHWinfo
Sends the Flash with Information message.

HANDoff
Sends the Handoff message. (For complete handoff testing, use the **CSS:CALL:PROCess** commands.)

LC
Sends the Local Control message.

MAINTenance
Sends the Maintenance message.

MSGWTG
Sends the Message Waiting message.

PAGE
Sends the Page order.

CSS:

FVC:

ORDER:

PU
Sends the Parameter Update message.

PWRLVL
Sends the Power Level message.

RELease
Sends the Release message.

SALERT
Sends the Stop Alert message.

SLOT1
Sends the Handoff to Digital Traffic Channel on Timeslot 1 message. (For complete handoff testing, use the **CSS:CALL:PROCess** commands.)

SLOT2
Sends the Handoff to Digital Traffic Channel on Timeslot 2 message. (For complete handoff testing, use the **CSS:CALL:PROCess** commands.)

SLOT3
Sends the Handoff to Digital Traffic Channel on Timeslot 3 message. (For complete handoff testing, use the **CSS:CALL:PROCess** commands.)

SNDAddr
Sends the Send Called Address message.

SNRreq
Sends the Serial Number Request message.

SSDUP
Sends the Shared Secret Data Update message.

UCHAL
Sends the Unique Challenge message.

B. FVC DATA FIELDS

CSS:

FVC:

AUTHBS *n*
Sets the AUTHBS value. Range of *n* is 0 to 262143.

AUTHBS?
Returns AUTHBS value setting.

CSS:

FVC:

CALLING:

NUM *"n"*
Sets Calling Party Number. The number *n* is entered as a string, with quotation marks (i.e., "316/522-4981").

NUM?
Returns Calling Party Number setting (string).

PI *n*
Sets Calling Party Number Presentation Indicator. Range of *n* is 0 to 3.

PI?
Returns Calling Party Number Presentation Indicator setting.

SI *n*
Sets Calling Party Screening Indicator. Range of *n* is 0 to 3.

SI?
Returns Calling Party Screening Indicator setting.

DMAC *n*
Sets Digital Mobile Attenuation Code. Range of *n* is 0 to 10.

DMAC?
Returns Digital Mobile Attenuation Code setting.

DVCC *n*
Sets Digital Verification Color Code. Range of *n* is 1 to 255.

DVCC?
Returns Digital Verification Color Code setting.

EF *n*
Sets Extended Protocol Forward Channel Indicator (0 or 1).

EF?
Returns Extended Protocol Forward Channel Indicator setting.

ENABLE:

VOICEPrivacy *n*
Sets the Voice Privacy in the Call Mode Acknowledgment message (0 or 1).

VOICEPrivacy?
Returns the Voice Privacy setting.

CSS:

FVC:

HANDoff:

CHANnel *n*
Sets analog Voice or Digital Traffic Channel for Handoff. Range of *n* is 1 to 1023. (Use the **CSS:CALL:PROCess** commands to initiate Handoff.)

CHANnel?
Returns Channel for Handoff setting.

LOCAL *n*
Sets the Local Control field used in the Local Control message. Range of *n* is 0 to 31.

LOCAL?
Returns the Local Control field setting.

MEM *n*
Sets Message Encryption Mode bit (0 or 1).

MEM?
Returns Message Encryption Mode bit setting.

MT *n*
Sets the Message Type field. Range of *n* is 0 to 31.

MT?
Returns the Message Type field setting.

PM *n*
Sets Privacy Mode bit (0 or 1).

PM?
Returns Privacy Mode bit setting.

PSCC *n*
Sets Present SAT Color Code. Range of *n* is 0 to 2.

PSCC?
Returns Present SAT Color Code setting.

PWRLVL *n*
Sets Power Level requested of Mobile Station in the Power Level message. Range of *n* is 0 to 7.

PWRLVL?
Returns requested Power Level setting.

RANDSSD *"n"*
Sets the 56 bit Random Number sent in the Shared Secret Data Update message. The number *n* is entered in hexadecimal as a string, with quotation marks (i.e., "4A59BE232F9C26").

CSS:

FVC:

RANDSSD?
Returns the Shared Secret Data 56 bit Random Number setting (string).

RANDU *n*
Sets the 24 bit Random Number sent in the Unique Challenge message. Range of *n* is 0 to 16777215.

RANDU?
Returns the Unique Challenge 24 bit Random Number setting.

SBI *n*
Sets Shortened Burst Indicator. Range of *n* is 0 to 3.

SBI?
Returns Shortened Burst Indicator setting.

SCC *n*
Sets Supervisory Audio Tone Color Code. Range of *n* is 0 to 2.

SCC?
Returns Supervisory Audio Tone Color Code setting.

SIGNAL:

CADENCE *n*
Sets the on, off pattern of the Alert tone. Range of *n* is 0 to 63.

CADENCE?
Returns the Alert tone on, off pattern setting.

PITCH *n*
Sets the pitch of the Alert tone. Range of *n* is 0 to 3.

PITCH?
Returns Alert tone pitch setting.

TA *n*
Sets Time Alignment. Range of *n* is 0 to 31.

TA?
Returns Time Alignment setting.

VMAC *n*
Sets Voice Mobile Attenuation Code. Range of *n* is 0 to 7.

VMAC?
Returns Voice Mobile Attenuation Code setting.

ADDENDUM TO SECTION 7-9-5

7-9-5A FVC

CSS:

FVC:

STARt
[CSS:FVC:STARt]
Starts the 1600CSA transmitting on the Forward Voice Channel.

STOP
[CSS:FVC:STOP]
Stops the Forward Voice Channel.

SAT ***n***
[CSS:FVC:SAT]
Specifies the SAT frequency on the FVC. Range of *n* is 0 or 5965 to 6035.

SAT?
Returns current value of SAT.

PVI ***n***
[CSS:FVC:PVI]
Enables (*n* = 1) or disables (*n* = 0) Protocol Version Indicator.

PVI?
Returns current state of PVI.

HYPERband ***n***
[CSS:FVC:HYPERband]
Specifies Hyperband. Range of *n* is 0 to 3.

HYPERband?
Returns current value of Hyperband.

ORDER:

RELEASE_Winfo
[CSS:FVC:ORDER:RELEASE_Winfo]
Sends the Release with DCCH Information message.

VOICE_MSG_WTG
[CSS:FVC:ORDER:VOICE_MSG_WTG]
Sends the Voice Message Waiting message.

SMS_MSG_WTG
[CSS:FVC:ORDER:SMS_MSG_WTG]
Sends the SMS Message Waiting message.

G3_MSG_WTG
[CSS:FVC:ORDER:G3_MSG_WTG]
Sends the G3-Fax Message Waiting message.

CSS:

FVC:

ORDER:

G3_PAGE
[CSS:FVC:ORDER:G3_PAGE]
Sends the Page message (Group 3 Fax).

ASYNC_PAGE
[CSS:FVC:ORDER:ASYNC_PAGE]
Sends the Page message (Async Data).

7-9-6 FORWARD DIGITAL TRAFFIC CHANNEL (FDTC)

Data commands set the data in the messages and order commands send the orders (messages). **CSS:FDTC:ENABLE** commands enable or disable optional fields. **CSS:FDTC:FACCH:** commands pertain to the blank and burst Fast Associated Control Channel (FACCH). **CSS:FDTC:SACCH:** commands pertain to the continuous Slow Associated Control Channel (SACCH).

A. FDTC ORDERS

CSS:

FDTC:

FACCH:

ALERT
Sends the Alert with Information message.

AUDIT
Sends the Audit message.

BSACK
Sends the Base Station Acknowledgment message. The **CSS:FDTC:AMT** *n* command sets acknowledgment type.

BSCHALCON
Sends the Base Station Challenge Confirmation message.

FLASH
Sends the Flash with Information message.

FLASHACK
Sends the Flash Acknowledgment message.

HANDoff
Sends the Handoff message. (For complete handoff testing, use the **CSS:CALL:PROCess** commands.)

LC
Sends the Local Control message.

MAINTenance
Sends the Maintenance message.

MEASure
Sends the Measurement message.

PLC
Sends the Physical Layer Control.

PU
Sends the Parameter Update message.

CSS:

FDTC:

FACCH:

RELease
Sends the Release message.

SBDA
Sends the Send Burst DTMF Acknowledge message.

SCDA
Sends the Send Continuous DTMF Acknowledge message.

SMEASure
Sends the Stop Measurement message.

SR
Sends the Status Request message.

SSDUP
Sends the Shared Secret Data Update message.

UCHAL
Sends the Unique Challenge message.

SACCH:

ALERT
Sends the Alert with Information message.

AUDIT
Sends the Audit message.

BSACK
Sends the Base Station Acknowledgment message. The **CSS:FDTC:AMT** *n* command sets acknowledgment type.

BSCHALCON
Sends the Base Station Challenge Confirmation message.

FLASH
Sends the Flash with Information message.

FLASHACK
Sends the Flash Acknowledgment message.

HANDoff
Sends the Handoff message. (For complete handoff testing, use the **CSS:CALL:PROCess** commands.)

LC
Sends the Local Control message.

CSS:

FDTC:

SACCH:

MAINTenance
Sends the Maintenance message.

MEASure
Sends the Measurement message.

PLC
Sends the Physical Layer Control.

PU
Sends the Parameter Update message.

RELease
Sends the Release message.

SBDA
Sends the Send Burst DTMF Acknowledge message.

SCDA
Sends the Send Continuous DTMF Acknowledge message.

SMEASure
Sends the Stop Measurement message.

SR
Sends the Status Request message.

SSDUP
Sends the Shared Secret Data Update message.

UCHAL
Sends the Unique Challenge message.

B. FDTC DATA FIELDS

CSS:

FDTC:

AMT:

CONNect
Acknowledges a Connect message from the Mobile Station.

RELease
Acknowledges a Release message from the Mobile Station.

STATus
Acknowledges a Status message from the Mobile Station.

CSS:

FDTC:

AMT?
Returns Acknowledge Message Type setting (string).

AUTHBS *n*
Sets the AUTHBS value. Range of *n* is 0 to 262143.

AUTHBS?
Returns AUTHBS value setting.

CALLING:

NUM *"n"*
Sets Calling Party Number. The number *n* is entered as a string, with quotation marks (i.e., "316/522-4981").

NUM?
Returns Calling Party Number setting (string).

PI *n*
Sets Calling Party Number Presentation Indicator. Range of *n* is 0 to 3.

PI?
Returns Calling Party Number Presentation Indicator setting.

PLANId *n*
Sets Calling Party Numbering Plan Identification. Range of *n* is 0 to 15.

PLANId?
Returns Calling Party Numbering Plan Identification setting.

SI *n*
Sets Calling Party Screening Indicator. Range of *n* is 0 to 3.

SI?
Returns Calling Party Screening Indicator setting.

TYpe *n*
Sets Calling Party Type. Range of *n* is 0 to 7.

TYpe?
Returns Calling Party Type setting.

CONTROL *n*
Sets the Local Control field used in the Local Control message. Range of *n* is 0 to 31.

CONTROL?
Returns the Local Control field setting.

DIC *n*
Sets Delay Interval Compensation bit (0 or 1).

CSS:

FDTC:

DIC?
Returns Delay Interval Compensation bit setting.

DMAC *n*
Sets Digital Mobile Attenuation Code. Range of *n* is 0 to 10.

DMAC?
Returns Digital Mobile Attenuation Code setting.

DTX *n*
Sets Discontinuous Transmission bit (0 or 1).

DTX?
Returns Discontinuous Transmission bit setting.

DVCC *n*
Sets Digital Verification Color Code. Range of *n* is 0 to 255.

DVCC?
Returns Digital Verification Color Code setting.

ENABLE:

CALLING:

NUM *n*
Enables or disables Calling Party Number field (1 or 0).

NUM?
Returns the Calling Party Number Enable setting.

DIC *n*
Enables or disables Delay Interval Compensation bit (1 or 0).

DIC?
Returns Delay Interval Compensation bit Enable setting.

DMAC *n*
Enables or disables Digital Mobile Attenuation Code (1 or 0).

DMAC?
Returns Digital Mobile Attenuation Code Enable setting.

DTX *n*
Enables or disables Discontinuous Transmission bit (1 or 0).

DTX?
Returns Discontinuous Transmission bit Enable setting.

CSS:

FDTC:

ENABLE:

LDP:

BSACK *n*
Enables or disables Last Decoded Parameter in Base Station Acknowledgment message (1 or 0).

BSACK?
Returns Last Decoded Parameter Enable setting for BSACK message.

FLASHACK *n*
Enables or disables Last Decoded Parameter in Flash Acknowledgment message (1 or 0).

FLASHACK?
Returns Last Decoded Parameter Enable setting for FLASHACK message.

SBDA *n*
Enables or disables Last Decoded Parameter in Send Burst DTMF Acknowledge message (1 or 0).

SBDA?
Returns Last Decoded Parameter Enable setting for SBDA message.

NOMW *n*
Enables or disables Number of Messages Waiting field (1 or 0).

NOMW?
Returns Number of Messages Waiting field Enable setting.

RFCHAN *n,x*
Enables or disables selected RF Channel index. Range of *n* is 0 to 11 (index). Range of *x* is 1 or 0 (enable or disable).

RFCHAN? *n*
Returns selected RF Channel index Enable setting. Range of *n* is 0 to 11.

SIGNAL *n*
Enables or disables Signal field (1 or 0).

SIGNAL?
Returns Signal field Enable setting.

STATUS:

CMODE *n*
Enables or disables Call Mode field in Status Request message (1 or 0).

CMODE?
Returns Call Mode field Enable setting.

CSS:

FDTC:

ENABLE:

STATUS:

ESN *n*
Enables or disables Electronic Serial Number field in Status Request message (1 or 0).

ESN?
Returns Electronic Serial Number field Enable setting.

MEM *n*
Enables or disables Message Encryption Mode field (1 or 0).

MEM?
Returns Message Encryption Mode field Enable setting.

TI *n*
Enables or disables Terminal Information field (1 or 0).

TI?
Returns Terminal Information field Enable setting.

VPM *n*
Enables or disables Voice Privacy Mode bit (1 or 0).

VPM?
Returns Voice Privacy Mode bit Enable setting.

TA *n*
Enables or Disables Time Alignment field (1 or 0)

TA?
Returns Time Alignment field Enable setting.

HANDoff:

CHANnel *n*
Sets analog Voice or Digital Traffic Channel for Handoff. Range of *n* is 1 to 1023. (Use the **CSS:CALL:PROCess** commands to initiate Handoff.)

CHANnel?
Returns channel for Handoff setting.

LDP *n*
Sets Last Decoded Parameter. Range of *n* is 0 to 15.

LDP?
Returns Last Decoded Parameter setting.

CSS:

FDTC:

MEM *n*
Sets Message Encryption Mode (0 or 1).

MEM?
Returns Message Encryption Mode setting.

NOMW *n*
Sets Number of Messages Waiting field. Range of *n* is 0 to 63.

NOMW?
Returns Number of Messages Waiting field setting.

RANDSSD *"n"*
Sets the 56 bit Random Number sent in the Shared Secret Data Update message. The number *n* is entered in hexadecimal as a string, with quotation marks (i.e., "4A59BE232F9C26").

RANDSSD?
Returns the Shared Secret Data 56 bit Random Number setting (string).

RANDU *n*
Sets the 24 bit Random Number sent in the Unique Challenge message. Range of *n* is 0 to 16777215.

RANDU?
Returns the Unique Challenge 24 bit Random Number setting.

RATe *n*
Sets Channel Rate (0 [Full-Rate] or 1 [Half-Rate]).

RATe?
Returns Channel Rate setting.

RFCHAN *n,x*
Sets selected RF Channel index. Range of *n* is 0 to 11 (index). Range of *x* is 1 to 1023 (RF Channel setting).

RFCHAN? *n*
Returns selected RF Channel index setting. Range of *n* is 0 to 11.

RN *n*
Sets Request Number. Range of *n* is 0 to 15.

RN?
Returns Request Number setting.

SBI *n*
Sets Shortened Burst Indicator. Range of *n* is 0 to 3.

SBI?
Returns Shortened Burst Indicator setting.

CSS:

FDTC:

SIGNAL:

CADENCE *n*
Sets the on, off pattern of the Alert tone. Range of *n* is 0 to 63.

CADENCE?
Returns the Alert tone on, off pattern setting.

PITCH *n*
Sets the pitch of the Alert tone. Range of *n* is 0 to 3.

PITCH?
Returns Alert tone pitch setting.

SLOT *n*
Sets Timeslot. Range of *n* is 1 to 3.

SLOT?
Returns Timeslot setting.

TA *n*
Sets Time Alignment. Range of *n* is 0 to 31.

TA?
Returns Time Alignment setting.

TI *n*
Sets Timeslot Indicator. Range of *n* is 0 to 6. (0 is analog.)

TI?
Returns Timeslot Indicator setting.

VPM *n*
Sets Voice Privacy Mode bit (0 or 1).

VPM?
Returns Voice Privacy Mode bit setting.

ADDENDUM TO SECTION 7-9-6

7-9-6A FDTC

CSS:

FDTC:

STARt
[CSS:FDTC:STARr]
Starts the 1600CSA transmitting on the Forward Digital Traffic Channel.

STOP
[CSS:FDTC:STOP]
Stops the Forward Digital Traffic Channel.

FACCH: or SACCH:

New Message types for the FDTC.

DEDicated:

HANDoff
[CSS:FDTC:FACCH: or SACCH:DEDicated:HANDoff]
Sends Dedicated DTC Handoff message.

HYPERband:

MEASure
[CSS:FDTC:FACCH: or SACCH:HYPERband:MEASure]
Sends Hyperband Measurement message.

CAPability:

REQuest
[CSS:FDTC:FACCH: or SACCH:CAPability:REQuest]
Sends Capability Update Request message.

RESPonse
[CSS:FDTC:FACCH: or SACCH:CAPability:RESPonse]
Sends Capability Update Response message.

RDATA:

MESSage
[CSS:FDTC:FACCH: or SACCH:RDATA:MESSage]
Sends R-DATA message.

ACCept
[CSS:FDTC:FACCH: or SACCH:RDATA:ACCept]
Sends R-DATA ACCEPT message.

REJect
[CSS:FDTC:FACCH: or SACCH:RDATA:REJect]
Sends R-DATA REJECT message.

CSS:

FDTC:

FACCH: or SACCH:

SOC
[CSS:FDTC:FACCH: or SACCH:SOC]
Sends SOC Message Delivery message.

BSMC
[CSS:FDTC:FACCH: or SACCH:BSMC]
Sends BSMC Message Delivery message.

RAW *x(1),x(2),x(3),x(4),x(5),x(6),...,x(n)*
[CSS:FDTC:FACCH: or SACCH:RAW]
This command is used to generate a user defined message. The parameters used with this command each setup 8 bits of the message. Since each word of a FACCH or SACCH message consists of 48 bits, it takes 6 parameters to make 1 word, 12 to make 2 words,.. etc. Therefore the number of parameters used with this message is variable, but it must be a multiple of 6.

RFCHAN *n,m*
[CSS:FDTC:RFCHAN]
Specifies RF Channel Number (*m*) indexed by *n*. Range of *n* is 0 to 23; range of *m* is 1 to 1023.

RFCHAN? *n*
Returns current value of RFCHAN indexed by *n*. Range of *n* is 0 to 23.

PVI *n*
[CSS:FDTC:PVI]
Enables (*n* = 1) or disables (*n* = 0) Protocol Version Indicator.

PVI?
Returns current state of PVI.

DTXControl *n*
[CSS:FDTC:DTXControl]
Enables (*n* = 1) or disables (*n* = 0) DTX Control.
Indicates the DTX mode supported on the channel to which a handoff is occurring.

DTXControl?
Returns current state of DTXControl.

CSS:

FDTC:

DCCHinfo:

Digital Control Channel Information.

NUMBer *n*
[CSS:FDTC:DCCHinfo:NUMBer]
Specifies Length of DCCH info content. Range of *n* is 0 to 2.

NUMBer?
Returns current value of NUMBer.

HYPERband *n,m*
[CSS:FDTC:DCCHinfo:HYPERband]
Specifies Hyperband (*m*) indexed by *n*. Range of *n* is 0 to 2; range of *m* is 0 to 3.

HYPERband? *n*
Returns current value of HYPERband indexed by *n*. Range of *n* is 0 to 2.

CHANnel *n,m*
[CSS:FDTC:DCCHinfo:CHANnel]
Specifies Digital Control Channel Information (*m*) indexed by *n*. Range of *n* is 0 to 2; range of *m* is 0 to 2047.

CHANnel? *n*
Returns current value of CHANnel indexed by *n*. Range of *n* is 0 to 2.

DVCC *n,m*
[CSS:FDTC:DCCHinfo:DVCC]
Specifies Digital Verification Color Code (*m*) indexed by *n*. Range of *n* is 0 to 2; range of *m* is 0 to 255.

DVCC? *n*
Returns current value of DVCC indexed by *n*. Range of *n* is 0 to 2.

MSGWTG:

NUMBer *n*
[CSS:FDTC:MSGWTG:NUMBer]
Specifies Length of Message Waiting info content. Range of *n* is 1 to 15.

NUMBer?
Returns current value of NUMBer.

CSS:

FDTC:

MSGWTG:

MESSage:

TYPE *n,m*
[CSS:FDTC:MSGWTG:MESSage:TYPE]
Specifies Messages Waiting Type (*m*) indexed by *n*. Range of *n* is 0 to 14; range of *m* is 0 to 15.

Indicates the type of messages that are waiting..

TYPE? *n*
Returns current value of TYPE indexed by *n*. Range of *n* is 0 to 14.

NUMBer *n,m*
[CSS:FDTC:MSGWTG:MESSage:NUMBer]
Specifies Number of Messages Waiting (*m*) indexed by *n*. Range of *n* is 0 to 14; range of *m* is 0 to 63.

Indicates the number of messages associated to the Message Waiting Type.

NUMBer? *n*
Returns current value of NUMBer indexed by *n*. Range of *n* is 0 to 14.

TASK *n*
[CSS:FDTC:TASK]
Specifies Task Status. Range of *n* is 0 to 7.

TASK?
Returns current value of TASK.

ATS *n*
[CSS:FDTC:ATS]
Specifies Assigned Time Slot. Range of *n* is 0 to 15.

ATS?
Returns current value of ATS.

PV *n*
[CSS:FDTC:PV]
Specifies Protocol Version. Range of *n* is 0 to 15.

PV?
Returns current value of PV.

CSS:

FDTC:

CHANGE:

SOC *n*
[CSS:FDTC:CHANGE:SOC]
Enables (*n* = 1) or disables (*n* = 0) System Operator Code Change Indicator.

Indicates whether or not the SOC associated with the current digital traffic channel is different from that associated with the digital traffic channel being assigned.

SOC?
Returns the value of SOC.

BSMC *n*
[CSS:FDTC:CHANGE:BSMC]
Enables (*n* = 1) or disables (*n* = 0) Base Station Manufacturer Code Change Indicator.

Indicates whether or not the BSMC associated with the current digital traffic channel is different from that associated with the digital traffic channel being assigned (0-1).

BSMC?
Returns current state of BSMC.

DELTA:

TIME *n*
[CSS:FDTC:DELTA:TIME]
Specifies Delta Time. Range of *n* is 0 to 2047.

Indicates timing advance in half symbols that shall be applied, relative to the current mobile station transmit time, on the assigned digital traffic channel.

TIME?
Returns current value of TIME.

VMI:

Used to provide voice mode operation information for the assigned digital traffic channel.

VC *n*
[CSS:FDTC:VMI:VC]
Specifies Voice Code. Range of *n* is 0 to 7.

VC?
Returns current value of VC.

CSS:

FDTC:

VMI:

PM_V *n*
[CSS:FDTC:VMI:PM_V]
Specifies Voice Privacy Mode. Range of *n* is 0 to 7.

PM_V?
Returns current value of PM_V.

DPM *n*
[CSS:FDTC:DPM]
Enables (*n* = 1) or disables (*n* = 0) Data Privacy Mode.

DPM?
Returns current state of DPM.

MEMC:

Identifies the message encryption mode of a mobile station.

MEA *n*
[CSS:FDTC:MEMC:MEA]
Specifies Message Encryption Mode C Algorithm. Range of *n* is 0 to 3.

MEA?
Returns current value of MEA.

MED *n*
[CSS:FDTC:MEMC:MED]
Specifies Message Encryption Mode C Domain. Range of *n* is 0 to 3.

MED?
Returns current value of MED.

MEK *n*
[CSS:FDTC:MEMC:MEK]
Specifies Message Encryption Mode C Key. Range of *n* is 0 to 3.

MEK?
Returns current value of MEK.

CSS:

FDTC:

MAP:

Identifies the forms of voice privacy supported by the BMI.

VPM *n*
[CSS:FDTC:MAP:VPM]
Specifies Voice Privacy Mode Map. Range of *n* is 0 to 15.

VPM?
Returns current value of VPM.

CODER *n*
[CSS:FDTC:MAP:CODER]
Specifies Voice Coder Map. Range of *n* is 0 to 63.

CODER?
Returns current value of CODER.

MEA:

DOMAIN *n*
[CSS:FDTC:MAP:MEA:DOMAIN]
Specifies Message Encryption Algorithm Map Domain. Range of *n* is 0 to 255.

DOMAIN?
Returns current value of DOMAIN.

ALGORithms *n,m*
[CSS:FDTC:MAP:MEA:ALGORithms]
Specifies Message Encryption Algorithm Map (*m*) indexed by *n*. Range of *n* is 0 to 7; range of *m* is 0 to 15.

ALGORithms? *n*
Returns current value of ALGORithms.

MEK *n*
[CSS:FDTC:MAP:MEK]
Specifies Message Encryption Key Map. Range of *n* is 0 to 15.

MEK?
Returns current value of MEK.

ARQ *n*
[CSS:FDTC:MAP:ARQ]
Enables (*n* = 1) or disables (*n* = 0) FACCH/SACCH ARQ Map.

ARQ?
Returns current state of ARQ.

CSS:

FDTC:

MAP:

SMS *n*
[CSS:FDTC:MAP:SMS]
Specifies SMS Map. Range of *n* is 0 to 3.

SMS?
Returns current value of SMS.

SUPPort:

IRA *n*
[CSS:FDTC:SUPPort:IRA]
Enables (*n* = 1) or disables (*n* = 0) IRA Support.

Indicates whether or not a mobile station or BMI supports IRA address encoding in the address field in the Message Center Address.

IRA?
Returns current state of IRA.

SOC *n*
[CSS:FDTC:SOC]
Enables (*n* = 1) or disables (*n* = 0) System Operator Code.

SOC?
Returns current state of SOC.

BSMC *n*
[CSS:FDTC:BSMC]
Enables (*n* = 1) or disables (*n* = 0) Base Station Manufacturer Code.

BSMC?
Returns current state of BSMC.

RTRANSaction *n*
[CSS:FDTC:RTRANSaction]
Specifies R-Transaction Identifier. Range of *n* is 0 to 255.

Used to uniquely associate a R-DATA ACCEPT or a R-DATA REJECT message with a specific R-DATA message.

RTRANSaction?
Returns current value of RTRANSaction.

CSS:

FDTC:

RDATA_UNIT:

Used to carry the higher layer SMS protocol data unit.

LENGth *n*
[CSS:FDTC:RDATA_UNIT:LENGth]
Specifies Length of the R-Data Unit info content. Range of *n* is 1 to 256.

LENGth?
Returns current value of LENGth.

HLP:

IDentifier *n*
[CSS:FDTC:RDATA_UNIT:HLP:IDentifier]
Specifies Higher Layer Protocol Identifier. Range of *n* is 0 to 255.

IDentifier?
Returns current value of IDentifier.

DATA *n,m*
[CSS:FDTC:RDATA_UNIT:HLP:DATA]
Specifies Higher Layer Protocol Data Unit (*m*) indexed by *n*. Range of *n* is 0 to 255; range of *m* is 0 to 255.

DATA? *n*
Returns current value of DATA. Range of *n* is 0 to 255.

MESSage:

CENTer:

TYPE *n*
[CSS:FDTC:MESSage:CENTer:TYPE]
Specifies Type of Number. Range of *n* is 0 to 7.

TYPE?
Returns current value of TYPE.

PLANid *n*
[CSS:FDTC:MESSage:CENTer:PLANid]
Specifies Numbering Plan Identification. Range of *n* is 0 to 15.

PLANid?
Returns current value of PLANid.

CSS:

FDTC:

MESSage:

CENTer:

ENCoding *n*
[CSS:FDTC:MESSage:CENTer:ENCoding]
Enables (*n* = 1) or disables (*n* = 0) Address Encoding.

ENCoding?
Returns current state of ENCoding.

ADDRess "*n*"
[CSS:FDTC:MESSage:CENTer:ADDRess]
Specifies Address (ASCII string).

Up to 60 characters in this field.

ADDRess?
Returns current string value of ADDRess.

USER:

DEST:

Used to identify the user destination address of a MS originated short message.

TYPE *n*
[CSS:FDTC:USER:DEST:TYPE]
Specifies Type of Number. Range of *n* is 0 to 7.

TYPE?
Returns current value of TYPE.

PLANid *n*
[CSS:FDTC:USER:DEST:PLANid]
Specifies Numbering Plan Identification. Range of *n* is 0 to 15.

PLANid?
Returns current value of PLANid.

ENCoding *n*
[CSS:FDTC:USER:DEST:ENCoding]
Enables (*n* = 1) or disables (*n* = 0) Address Encoding.

ENCoding?
Returns current state of ENCoding.

CSS:

FDTC:

USER:

DEST:

ADDRess "*n*"
[CSS:FDTC:USER:DEST:ADDRess]
Specifies Address (ASCII string).

Up to 60 characters may be sent.

ADDRess?
Returns current string value of ADDRess.

SUBaddress:

Used to identify the subaddress of the destination user of a short message.

LENGth *n*
[CSS:FDTC:USER:DEST:SUBaddress:LENGth]
Specifies Length of subaddress info content. Range of *n* is 0 to 21.

LENGth?
Returns current value of LENGth.

ODD_EVEN *n*
[CSS:FDTC:USER:DEST:SUBaddress:ODD_EVEN]
Enables (*n* = 1) or disables (*n* = 0) Odd/Even Indicator.

ODD_EVEN?
Returns current state of ODD_EVEN.

TYPE *n*
[CSS:FDTC:USER:DEST:SUBaddress:TYPE]
Specifies Type of subaddress. Range of *n* is 0 to 7.

TYPE?
Returns current value of TYPE.

REServed *n*
[CSS:FDTC:USER:DEST:SUBaddress:REServed]
Enables (*n* = 1) or disables (*n* = 0) Reserved.

REServed?
Returns current state of REServed.

CSS:

FDTC:

USER:

DEST:

SUBaddress:

ADDRess *n,m*
[CSS:FDTC:USER:DEST:SUBaddress:ADDRess]
Specifies User Destination Subaddress (*m*) indexed by *n*. Range of *n* is 0 to 19; range of *m* is 0 to 255.

ADDRess? *n*
Returns current value of User Destination Subaddress indexed by *n*. Range of *n* is 0 to 19.

ORIG:

Used to identify the originating address of a short message.

TYPE *n*
[CSS:FDTC:USER:ORIG:TYPE]
Specifies Type of Number. Range of *n* is 0 to 7.

TYPE?
Returns current value of TYPE.

PLANid *n*
[CSS:FDTC:USER:ORIG:PLANid]
Specifies Numbering Plan Identification. Range of *n* is 0 to 15.

PLANid?
Returns current value of PLANid.

ENCoding *n*
[CSS:FDTC:USER:ORIG:ENCoding]
Enables (*n* = 1) or disables (*n* = 0) Address Encoding.

ENCoding?
Returns current state of ENCoding.

ADDRess "*n*"
[CSS:FDTC:USER:ORIG:ADDRess]
Specifies Address (ASCII string).

Up to 60 characters may be sent.

ADDRess?
Returns current string value of ADDRess.

CSS:

FDTC:

USER:

ORIG:

SUBaddress:

Used to identify the subaddress of the originating user of a short message.

LENGth *n*
[CSS:FDTC:USER:ORIG:SUBaddress:LENGth]
Specifies Length of User Originating subaddress info content. Range of *n* is 0 to 21.

LENGth?
Returns current value of LENGth.

ODD_EVEN *n*
[CSS:FDTC:USER:ORIG:SUBaddress:ODD_EVEN]
Enables (*n* = 1) or disables (*n* = 0) Odd/Even Indicator.

ODD_EVEN?
Returns current state of ODD_EVEN.

TYPE *n*
[CSS:FDTC:USER:ORIG:SUBaddress:TYPE]
Specifies Type of subaddress. Range of *n* is 0 to 7.

TYPE?
Returns the value of TYPE.

REServed *n*
[CSS:FDTC:USER:ORIG:SUBaddress:REServed]
Enables (*n* = 1) or disables (*n* = 0) Reserved.

REServed?
Returns current state of REServed.

ADDRess *n,m*
[CSS:FDTC:USER:ORIG:SUBaddress:ADDRess]
Specifies User Originating Subaddress (*m*) indexed by *n*. Range of *n* is 0 to 19; range of *m* is 0 to 255.

ADDRess? *n*
Returns current value of User Originating Subaddress indexed by *n*. Range of *n* is 0 to 19.

CSS:

FDTC:

USER:

ORIG:

PRESentation:

PI ***n***
[CSS:FDTC:USER:ORIG:PRESentation:PI]
Specifies Presentation Indicator. Range of *n* is 0 to 3.

PI?
Returns current value of PI.

SI ***n***
[CSS:FDTC:USER:ORIG:PRESentation:SI]
Specifies Screening Indicator. Range of *n* is 0 to 3.

SI?
Returns current value of SI.

REServed ***n***
[CSS:FDTC:USER:ORIG:PRESentation:REServed]
Specifies Reserved. Range of *n* is 0 to 15.

REServed?
Returns current value of REServed.

RCAUSE ***n***
[CSS:FDTC:RCAUSE]
Specifies R-Cause. Range of *n* is 0 to 255.

Used to qualify an R-DATA REJECT message.

RCAUSE?
Returns current value of RCAUSE.

CUSTOM:

LENGth ***n***
[CSS:FDTC:CUSTOM:LENGth]
Specifies Length of Custom Control in octets. Range of *n* is 1 to 255.

LENGth?
Returns current value of LENGth.

CSS:

FDTC:

CUSTOM:

CONTrol *n,m*
[CSS:FDTC:CUSTOM:CONTrol]
Specifies Custom Control (*m*) indexed by *n*. Range of *n* is 0 to 255; range of *m* is 0 to 255.

CONTrol? *n*
Returns current value of CONTrol indexed by *n*. Range of *n* is 0 to 255.

HYPERband:

NUMBer *n*
[CSS:FDTC:HYPERband:NUMBer]
Specifies Number of Hyperband channels. Range of *n* is 0 to 24.

NUMBer?
Returns current value of NUMBer.

BAND *n,m*
[CSS:FDTC:HYPERband:BAND]
Specifies the Hyperband (*m*) indexed by *n*. Range of *n* is 0 to 23; range of *m* is 0 to 3.

BAND? *n*
Returns current value of BAND.

CHANnel *n,m*
[CSS:FDTC:HYPERband:CHANnel]
Specifies Hyperband channels (*m*) indexed by *n*. Range of *n* is 0 to 23; range of *m* is 0 to 2047.

CHANnel? *n*
Returns current value of CHANnel indexed by *n*. Range of *n* is 0 to 23

TARGet *n*
[CSS:FDTC:HYPERband:TARGet]
Specifies Target Hyperband. Range of *n* is 0 to 3.
Specifies the hyperband to which handoff is occurring.

TARGet?
Returns current value of TARGet.

CSS:

FDTC:

ENABLE:

The following commands enable or disable the optional Information Elements.

DCCHinfo ***n***
[CSS:FDTC:ENABLE:DCCHinfo]
Specifies Digital Control Channel Information.

DCCHinfo?
Returns current value of DCCHinfo.

MSGWTG ***n***
[CSS:FDTC:ENABLE:MSGWTG]
Specifies Other Messages Waiting Info.

MSGWTG?
Returns current value of MSGWTG.

TASK ***n***
[CSS:FDTC:ENABLE:TASK]
Specifies Task Status.

TASK?
Returns current value of TASK.

DELTA:

TIME ***n***
[CSS:FDTC:ENABLE:DELTA:TIME]
Specifies Delta Time.

TIME?
Returns current value of TIME.

VMI ***n***
[CSS:FDTC:ENABLE:VMI]
Specifies Voice Mode.

VMI?
Returns current value of VMI.

DPM ***n***
[CSS:FDTC:ENABLE:DPM]
Specifies Data Privacy Mode.

DPM?
Returns current value of DPM.

CSS:

FDTC:

ENABLE:

MEMC *n*
[CSS:FDTC:ENABLE:MEMC]
Specifies Message Encryption Mode C.

MEMC?

HYPERband:

TARGet *n*
[CSS:FDTC:ENABLE:HYPERband:TARGet]
Specifies Target Hyperband.

TARGet?
Returns current value of TARGet.

MESSage:

CENTer:

ADDRess *n*
[CSS:FDTC:ENABLE:MESSage:CENTer:ADDRess]
Specifies Message Center Address.

ADDRess?
Returns current value of ADDRess.

USER:

DEST:

ADDRess *n*
[CSS:FDTC:ENABLE:USER:DEST:ADDRess]
Specifies User Destination Address.

ADDRess?
Returns current value of ADDRess.

SUBaddress *n*
[CSS:FDTC:ENABLE:USER:DEST:SUBaddress]
Specifies User Destination Subaddress.

SUBaddress?
Returns current value of SUBaddress.

CSS:

FDTC:

ENABLE:

USER:

ORIG:

ADDRess *n*
[CSS:FDTC:ENABLE:USER:ORIG:ADDRess]
Specifies User Originating Address.

ADDRess?
Returns current value of ADDRess.

SUBaddress *n*
[CSS:FDTC:ENABLE:USER:ORIG:SUBaddress]
Specifies User Originating Subaddress.

SUBaddress?
Returns current value of SUBaddress.

PRESentation *n*
[CSS:FDTC:ENABLE:USER:ORIG:PRESentation]
Specifies User Originating Address Presentation Indicator.

PRESentation?
Returns current value of PRESentation.

7-9-7 TALKBACK

Talkback commands put data received on the RDTC into data fields of the FDTC. Data including VSELP loops back to the Mobile Station. The FM/AM-1600CSA performs no decoding or error correction with this operation.

CSS:

FDTC:

TALK:

DELAY *n*
Adds delay between receiving and transmitting in 20 ms intervals. Range of *n* is 0 to 250.

START
Starts Talkback operation.

STOP
Stops Talkback operation.

7-9-8 GLOBAL ACTION OVERHEAD MESSAGES

Global Action Overhead Messages are appended to the System Parameter Overhead message increasing the length of the overhead message train. Action (**CSS:GLACT:ACTion**) commands enable specific Global Action messages. Other commands define data and control operation.

CSS:

GLACT:

SEND
Starts Sending the Global Action as part of the Overhead message train.

STOP
Stops sending the Global Action as part of the Overhead message train.

REPEAT:

OFF
Sends the Global Action Overhead message once after starting (**CSS:GLACT:SEND**).

ON
Sends the Global Action Overhead message continuously after starting (**CSS:GLACT:SEND**) and stops when the **CSS:GLACT:STOP** command is initiated.

ACTion:

ACCess *n*
Enables or disables Access Attempt Parameters message (1 or 0).

ACCess?
Returns Access Attempt Parameters message Enable setting.

BIS *n*
Enables or disables Access Type Parameters message (1 or 0). The Access Type Parameters message contains the BIS bit.

BIS?
Returns Access Type Parameters message Enable setting.

LOCAID *n*
Enables or disables Location Area message (1 or 0).

LOCAID?
Returns Location Area message Enable setting.

LOCAL1 *n*
Enables or disables Local Control 1 message (1 or 0).

LOCAL1?
Returns Local Control 1 message Enable setting.

LOCAL2 *n*
Enables or disables Local Control 2 message (1 or 0).

CSS:

GLACT:

ACTion:

LOCAL2?
Returns Local Control 2 message Enable setting.

NEWACC *n*
Enables or disables New Access Channel Set message (1 or 0).

NEWACC?
Returns New Access Channel Set message Enable setting.

OLC *n*
Enables or disables Overload Control message (1 or 0).

OLC?
Returns Overload Control message Enable setting.

RANDA *n*
Enables or disables Random Challenge A message (1 or 0).

RANDA?
Returns Random Challenge A message Enable setting.

RANDB *n*
Enables or disables Random Challenge B message (1 or 0).

RANDB?
Returns Random Challenge B message Enable setting.

REGINCR *n*
Enables or disables Registration Increment message (1 or 0).

REGINCR?
Returns Registration Increment message Enable setting.

RESCAN *n*
Enables or disables Rescan message (1 or 0).

RESCAN?
Returns Rescan message Enable setting.

BIS *n*
Sets Busy-Idle Status bit (0 or 1).

BIS?
Returns Busy-Idle Status bit setting.

LOCAID *n*
Sets Cell Site Location Area Identification. Range of *n* is 0 to 4095.

CSS:

GLACT:

LOCAID?
Returns Cell Site Location Area Identification setting.

LOCALcntl *n*
Sets Local Control bits. Range of *n* is 0 to 65535.

LOCALcntl?
Returns value set for the Local Control bits.

LREG *n*
Sets Local Area Identification Registration bit (0 or 1).

LREG?
Returns Local Area Identification Registration bit setting.

MAXBusy:

OTHer *n*
Sets field indicating Maximum number of Busy occurrences allowed for Other than Page responses. Range of *n* is 0 to 15.

OTHer?
Returns Maximum number of Busy occurrences allowed for Other than Page responses setting.

PGR *n*
Sets field indicating Maximum number of Busy occurrences allowed for Page responses. Range of *n* is 0 to 15.

PGR?
Returns Maximum number of Busy occurrences allowed for Page responses field setting.

MAXSztr:

OTHer *n*
Sets field indicating Maximum number of Seizure Tries allowed for Other than Page responses. Range of *n* is 0 to 15.

OTHer?
Returns Maximum number of Seizure Tries allowed for Other than Page responses field setting.

PGR *n*
Sets field indicating Maximum number of Seizure Tries for Page responses. Range of *n* is 0 to 15.

PGR?
Returns Maximum number of Busy occurrences allowed for Page responses field setting.

CSS:

GLACT:

NEWACC *n*
Sets New Access Channel starting point field. Range of *n* is 0 to 2047.

NEWACC?
Returns New Access Channel starting point field setting.

OLC *n*
Sets Overhead Class field. Range of *n* is 0 to 32767.

OLC?
Returns Overhead Class field setting.

PDREG *n*
Sets Power Down Registration bit (0 or 1).

PDREG?
Returns Power Down Registration bit setting.

PUREG *n*
Sets Power Up Registration bit (0 or 1).

PUREG?
Returns Power Up Registration bit setting.

RAND1_A *n*
Sets 16 most significant bits of RAND. Range of *n* is 0 to 32767.

RAND1_A?
Returns value set for the 16 most significant bits of RAND.

RAND1_B *n*
Sets 16 least significant bits of RAND. Range of *n* is 0 to 32767.

RAND1_B?
Returns value set for the 16 least significant bits of RAND.

REGINCR *n*
Sets Registration Increment field. Range of *n* is 0 to 4095.

REGINCR?
Returns Registration Increment field setting.

7-9-9 MOBILE STATION CONTROL MESSAGES

Mobile Station Control messages, sent on the Forward Control Channel, replace the Overhead message train. **CSS:MSCM:ORDER:** commands select the Mobile Station Control Message.

CSS:

MSCM:

SEND
Starts sending the Mobile Station Control message.

STOP
Stops sending the Mobile Station Control message.

REPEAT:

OFF
Sends the Mobile Station Control message once after starting (**CSS:MSCM:SEND**).

ON
Sends the Mobile Station Control message continuously after starting (**CSS:MSCM:SEND**) and stops when the **CSS:MSCM:STOP** command is initiated.

ORDER:

A_ALERT
Selects Abbreviated Alert message.

AUDIT
Selects Audit message.

BSCHALCON
Selects Base Station Challenge Confirmation message.

DIR_RTRY
Selects Directed-Retry message.

INTRCPT
Selects Intercept message.

LC
Selects Local Control message.

MSG_WTG
Selects Message Waiting message.

REG_CNF
Selects Registration Confirmation message.

RELease
Selects Release message.

REORDER
Selects Reorder message.

CSS:

MSCM:

ORDER:

SLOT1
Selects Digital Channel Assignment to Timeslot 1 message.

SLOT2
Selects Digital Channel Assignment to Timeslot 2 message.

SLOT3
Selects Digital Channel Assignment to Timeslot 3 message.

SSD_UP
Selects Shared Secret Data Update message.

UCHAL
Selects Unique Challenge message.

VC_DES
Selects Voice Channel Designation message.

AUTHBS *n*
Sets AUTHBS value. Range of *n* is 0 to 262143.

AUTHBS?
Returns AUTHBS value setting.

CHAN *n*
Selects RF Channel. Range of *n* is 1 to 1023.

CHAN?
Returns RF Channel setting.

CHANPos *n,x*
Sets selected Channel Position field sent in Directed-Retry message. Range of *n* is 0 to 5 (indicating Channel Positions 1 to 6). Range of *x* is 0 to 127.

CHANPos? *n*
Returns selected Channel Position field setting. Range of *n* is 0 to 5.

DMAC *n*
Sets Digital Mobile Attenuation Code. Range of *n* is 0 to 10.

DMAC?
Returns Digital Mobile Attenuation Code setting.

DVCC *n*
Sets Digital Verification Color Code. Range of *n* is 0 to 255.

DVCC?
Returns Digital Verification Color Code setting.

CSS:

MSCM:

EF *n*
Sets Extended Protocol Forward Channel Indicator (0 or 1).

EF?
Returns Extended Protocol Forward Channel Indicator setting.

LOCAL *n*
Sets the Local Control (Local Control message)/Message Type field. Range of *n* is 0 to 31.

LOCAL?
Returns the Local Control/Message Type field setting.

MEM *n*
Sets Message Encryption Mode (0 or 1).

MEM?
Returns Message Encryption Mode setting.

MIN *"n"*
Selects Mobile Identification Number. The Mobile Identification Number (*n*) is entered as a string, with quotation marks (i.e., "316/522-4981").

MIN?
Returns Mobile Identification Number string setting.

ORDQ *n*
Sets Order Qualifier field. Range of *n* is 0 to 7.

ORDQ?
Returns Order Qualifier field setting.

PM *n*
Sets Privacy Mode Indicator bit (0 or 1).

PM?
Returns Privacy Mode Indicator bit setting.

RANDSSD1 *n*
Sets the 24 most significant bits of the Random Number sent in the SSD Update message (first order word). Range of *n* is 0 to 16777215.

RANDSSD1?
Returns the value set for the 24 most significant bits of the Random Number sent in the SSD Update message.

RANDSSD2 *n*
Sets the 24 intermediate bits of the Random Number sent in the SSD Update message (second order word). Range of *n* is 0 to 16777215.

CSS:

MSCM:

RANDSSD2?
Returns the value set for the 24 intermediate bits of the Random Number sent in the SSD Update message.

RANDSSD3 *n*
Sets the eight least significant bits of the Random Number sent in the SSD Update message (third order word). Range of *n* is 0 to 255.

RANDSSD3?
Returns the value set for the eight least significant bits of the Random Number sent in the SSD Update message.

RANDU *n*
Sets the 24 bit Random Number sent in the Unique Challenge message. Range of *n* is 0 to 16777215.

RANDU?
Returns the Unique Challenge 24 bit Random Number setting.

SBI *n*
Sets Shortened Burst Indicator. Range of *n* is 0 to 3.

SBI?
Returns Shortened Burst Indicator setting.

SCC *n*
Sets Supervisory Audio Tone Color Code. Range of *n* is 0 to 2.

SCC?
Returns Supervisory Audio Tone Color Code setting.

VMAC *n*
Sets Voice Mobile Attenuation Code. Range of *n* is 0 to 7.

VMAC?
Returns Voice Mobile Attenuation Code setting.

ADDENDUM TO SECTION 7-9-9

7-9-9A FOCC

CSS:

MSCM:

PVI *n*
[CSS:MSCM:PVI]
Enables (*n* = 1) or disables (*n* = 0) Protocol Version Indicator.

PVI?
Returns current state of PVI.

ORDER:

VOICE_MSG_WTG
[CSS:MSCM:ORDER:VOICE_MSG_WTG]
Selects Voice Message Waiting message.

SMS_MSG_WTG
[CSS:MSCM:ORDER:SMS_MSG_WTG]
Selects SMS Message Waiting message.

G3_MSG_WTG
[CSS:MSCM:ORDER:G3_MSG_WTG]
Selects G3-Fax Message Waiting message.

G3_PAGE
[CSS:MSCM:ORDER:G3_PAGE]
Selects Page (Group 3 Fax) message.

ASYNC_PAGE
[CSS:MSCM:ORDER:ASYNC_PAGE]
Selects Page (Async Data) message.

ANA_VC_DES
[CSS:MSCM:ORDER:ANA_VC_DES]
Selects Analog Voice Channel Assignment message.

7-9-10 OVERHEAD ENABLE COMMANDS

CSS:

ENABLE:

DCCH *n*
[CSS:ENABLE:DCCH]
Enables (*n* = 1) or disables (*n* = 0) DCCH information word.

REGID *n*
[CSS:ENABLE:REGID]
Enables (*n* = 1) or disables (*n* = 0) Registration ID word.

7-10 BER COMMANDS

BER commands measure the receive Bit Error Rate (BER) of a Base Station. BER is calculated for **BER:RDTC:DATA** and **BER:RDTC:PSUEDO**. **BER:DATA:LOOPBACK** and **BER:DATA:45MHZ_OFFset** enable the Base Station to measure own BER.

BER:

RDTC:

SETup
Sets up the FM/AM-1600CSA as when entering the Base Station Digital Traffic BER screen. (Screen is not displayed.)

CHANnel *n*
Sets RF Channel. Range of *n* is 1 to 1023.

SLOT *n*
Selects Digital Traffic Timeslot. Range of *n* is 1 to 3.

RFLVL *n*
Sets RF Level in dBm. Range of *n* is -30.0 to -127.0.

DATA:

USER
Sends user selected data (Reverse Digital Traffic Channel data set in the User Defined Data Field Setup screen). Pressing the SETUP Key in the Base Station Digital Traffic BER screen accesses the User Defined Data Field Setup screen. Data must be set prior to initiating this command.

PSUEDO
Sends pseudo-random data.

LOOPBACK
Sends data received from Base Station in the FDTC Slot format, back to the Base Station in the RDTC Slot format (for Base Stations with self BER test capabilities).

45MHZ_OFFset
Downconverts frequency 45 MHz and retransmits data.

GO
Starts Base Station Digital Traffic BER test.

STOP
Stops Base Station Digital Traffic BER test.

BITS?
Returns number of bits.

ERRORS?
Returns number of bit errors.

BER?
Returns Bit Error Rate (percentage).

BER:

RDTC:

CLEAR
Clears current results.

STATUS?
Returns synchronous data status (1 if Base Station cannot sync up to the data or 0).

7-11 MODULATION ACCURACY COMMANDS

Modulation Accuracy commands measure the π/4DQPSK modulation of Base Station signals.

MODacc:

FDTC:

SETup
Sets up the FM/AM-1600CSA as when entering the Modulation Accuracy screen. (Screen is not displayed.)

CHANnel *n*
Sets RF Channel. Range of *n* is 1 to 1023.

RUN?
Starts Modulation Accuracy measurements and returns adjusted AGC value.

COMPlete?
Returns test status (1 if complete or 0 if not complete). (Test takes ≈25 seconds.)

EVM?
Returns RMS Error Vector Magnitude in percent.

FREQ_ERRor?
Returns Frequency Error in hertz.

IQ_OFFset?
Returns I/Q Offset in dB.

MAG_ERRor?
Returns RMS Magnitude Error in percent.

PHASE_ERRor?
Returns RMS Phase Error in degrees.

7-12 GENERIC MEASURE COMMANDS

MEASure:

SAT?
Returns Supervisory Audio Tone frequency reading in Hz.

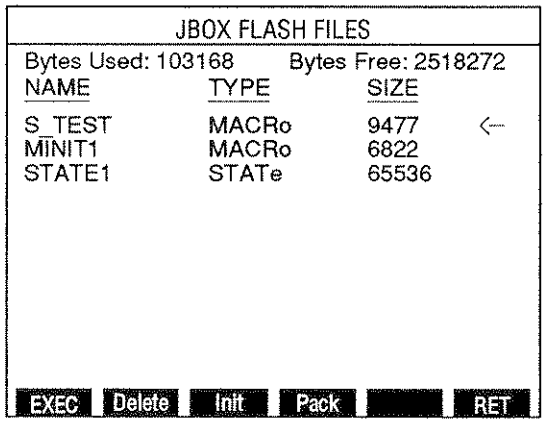
ST?
Returns Signal Tone frequency reading in Hz.

7-13 FLASH MEMORY COMMANDS

The mass memory (**MMEMory**) subsystem provides the Flash Memory storage capability of the FM/AM-1600CSA. Memory files are displayed in the JBox Flash Files Directory. 512 directory entries are available to store Test Set states, Macro programs, or Calibration data. File names are strings, limited to eight characters. The system promotes all lower case file name characters to upper case. Access and operate JBox Flash Files Directory as follows:

STEP PROCEDURE

1. Press DPLX MODE Key to display Duplex Operation Screen.
2. Press "Sp Tst" Soft Function Key F5 to display Dual Mode Cellular Menu.
3. Press "AUX2" Soft Function Key F5 to display Auxiliary Functions Menu.
4. Press 6 DATA ENTRY Key to display JBox Flash Files directory.



9117126

Figure 7-1 JBox Flash Files Directory

5. Use ↑ and ↓ FIELD SELECT Keys to select desired file (indicated by arrow).
6. To execute macro, press "EXEC" Soft Function Key F1.
7. To delete a file, press "Delete" Soft Function Key F2. Use ↑ and ↓ FIELD SELECT Keys to verify and press ENTER. After deletion, perform Step 8 to release Flash Memory space.
8. To perform Pack operation, press "Pack" Soft Function Key F4. Use ↑ and ↓ FIELD SELECT Keys to verify and press ENTER.

   NOTE: Do not power off Test Set during Pack operation as files may be lost.

9. To Initialize Flash Memory, press "Init" Soft Function Key F3. (Initializing clears Flash Memory and all files are lost.)
10. To return to Auxiliary Functions Menu (#2), press "RET" Soft Function Key F6.

Files are stored in Flash Memory using remote commands only. Calibration Data Sets and Test Set States are recalled using remote commands or as part of executable macros. Table 7-3 contains the remote commands used to operate the JBox Flash Files Directory. Error messages are returned to Host when errors occur. Refer to Table 4-3 for description of error messages.

COMMAND	RANGE/VALUE	DESCRIPTION
MMEMory:CATalog?		Returns Flash Memory status. First number returned is memory space used in bytes. Second number returned is memory space available in bytes. Remainder data is returned in sets of 3 consisting of file name, file type and file size for each file stored in Flash Memory.
MMEMory:CATalog:ENTRY? n	n is line number (index) in JBox Flash Files Directory. Set n from 0 to 512.	Returns file entry (file name, file type, file size) for given index. Returns $$$ if past end of directory or --- for deleted file.
MMEMory:CATalog:USED?		Returns file space used, in bytes.
MMEMory:CATalog:FREE?		Returns available file space, in bytes.
MMEMory:DELete "f"	f is file name.	Deletes file but does not release memory space until Pack operation is done.
MMEMory:INITialize		Erases all files stored in Flash Memory.
MMEMory:INITialize?		Returns 1 if file system has been initialized, 0 otherwise.
MMEMory:LOAD:MACRo "m","f"	m is name of designated macro. f is file name.	Loads macros and variables stored as the file name from Flash Memory into Test Set memory. If m is *, designated macro is executed. If m is macro name, that macro is executed. If m is omitted (""), no macro is executed.
MMEMory:LOAD:STATe "f"	f is file name.	Loads Test Set State stored as f from Flash Memory (sets CSA according to parameters stored in Flash Memory).
MMEMory:LOAD:CALibration "f"	f is file name.	Loads Calibration Data from Flash Memory into Test Set memory.
MMEMory:PACK		Packs Flash Memory and frees memory space from deleted files. Do not power off Test Set during Pack function (may lose files). Do not Pack with a WINDOW open.
MMEMory:STORe:MACRo "m","f"	m is name of designated macro. f is file name.	Stores all Test Set macros and variables (except free variables) in Flash Memory as f with macro specified as designated macro in JBox Flash Files Directory.
MMEMory:STORe:STATe "f"	f is file name.	Stores current CSA settings as f in Flash Memory (JBox Flash Files Directory).
MMEMory:STORe:CALibration "f"	f is file name.	Stores Test Set Calibration Data into Flash Memory (shown as BINary files in JBox Flash Files Directory.
MMEMory:TYPE? "f"	f is file name.	Returns file type (MACRo, STATe, BINary). Returns null string if file does not exist.

Table 7-3 CSA Flash Memory Remote Commands

7-14 DCCH MOBILE STATION SIMULATION COMMANDS

This section contains the TMAC commands necessary to simulate the transmit portion of a mobile station operating on the Digital Control Channel (DCCH). These commands are to be used, primarily, as "tools" in applications.

The FM/AM-1600CSA must be receiving data that contains valid sync words on the Forward Digital Control Channel (FDCCH) before it can transmit on the Reverse Digital Control Channel (RDCCH). The RDCCH transmitter waits for a sync word, in the slot to which it is assigned, before transmitting.

In order that the Test Set can understand the transmissions being received, the user must observe the following guidelines illustrated below:

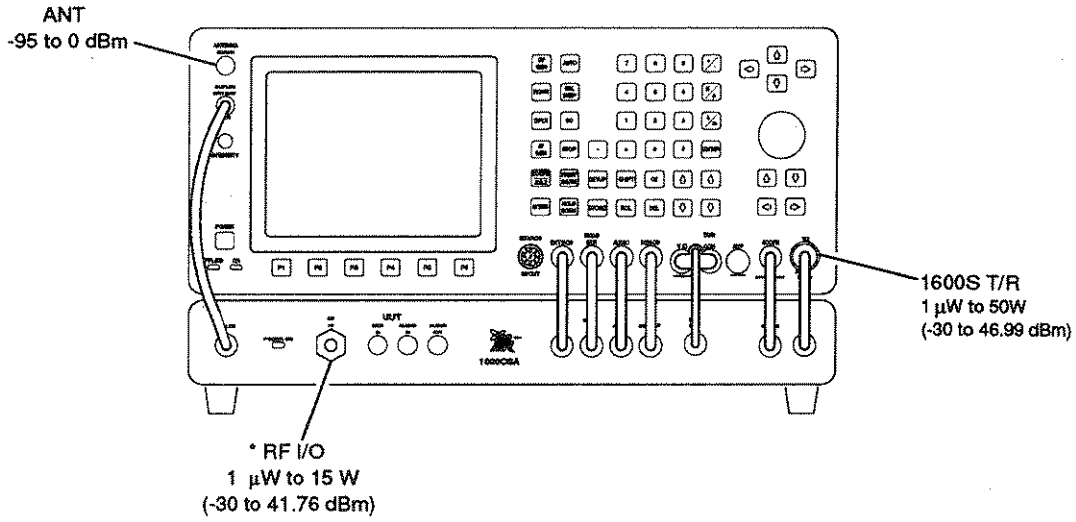


Figure 7-3 Connector Input Power Ranges

The user is to verify cable connections as depicted in Figure 7-3. There are three possible configurations for interfacing the Test Set to a mobile unit or cell site. See Table 7-4 for interface configurations:

CONFIGURATION	1600S ANT	1600S T/R	1600CSA RF I/0
1	I	*	O
2	*	I	O
3	*	*	I/O
I - Input O - Output * - High Impedance			

Table 7-4 Input/Output Connector Configurations

7-14-1 TDMA TRANSMISSION OVERVIEW

The type of TDMA Transmission (as discussed in each subsection of Section 7-14) is determined by three different factors figured in the setup:

- Length
- Mode
- Selection of type of data for data field

The Length can be Normal or Abbreviated. In a Normal length message, the User Data block is 101 bits long, while in an Abbreviated length message, the User Data block is only 79 bits long. In both cases the TDMA slot length is the same (324 bits); however, in the Abbreviated Length Message, extra Ramp Time bits and Guard Time bits are added that make up the difference. See Figure 7-4 for an illustration of a Normal Length Message. See Figure 7-5 for an illustration of an Abbreviated Length Message.

The Mode can be Contiguous or Sub Channel. In a Contiguous transmission mode, the FM/AM-1600CSA transmits at a Full-Rate TDMA in the slot determined by the **MSS:SLOT** ***n*** command discussed in 7-14-2. In a Sub Channel transmission mode, the FM/AM-1600CSA transmits in a Sub Channel slot. See Figure 7-6 for an illustration of a Contiguous transmission and a Sub Channel transmission.

The data field can consist of either random data or user-defined data.

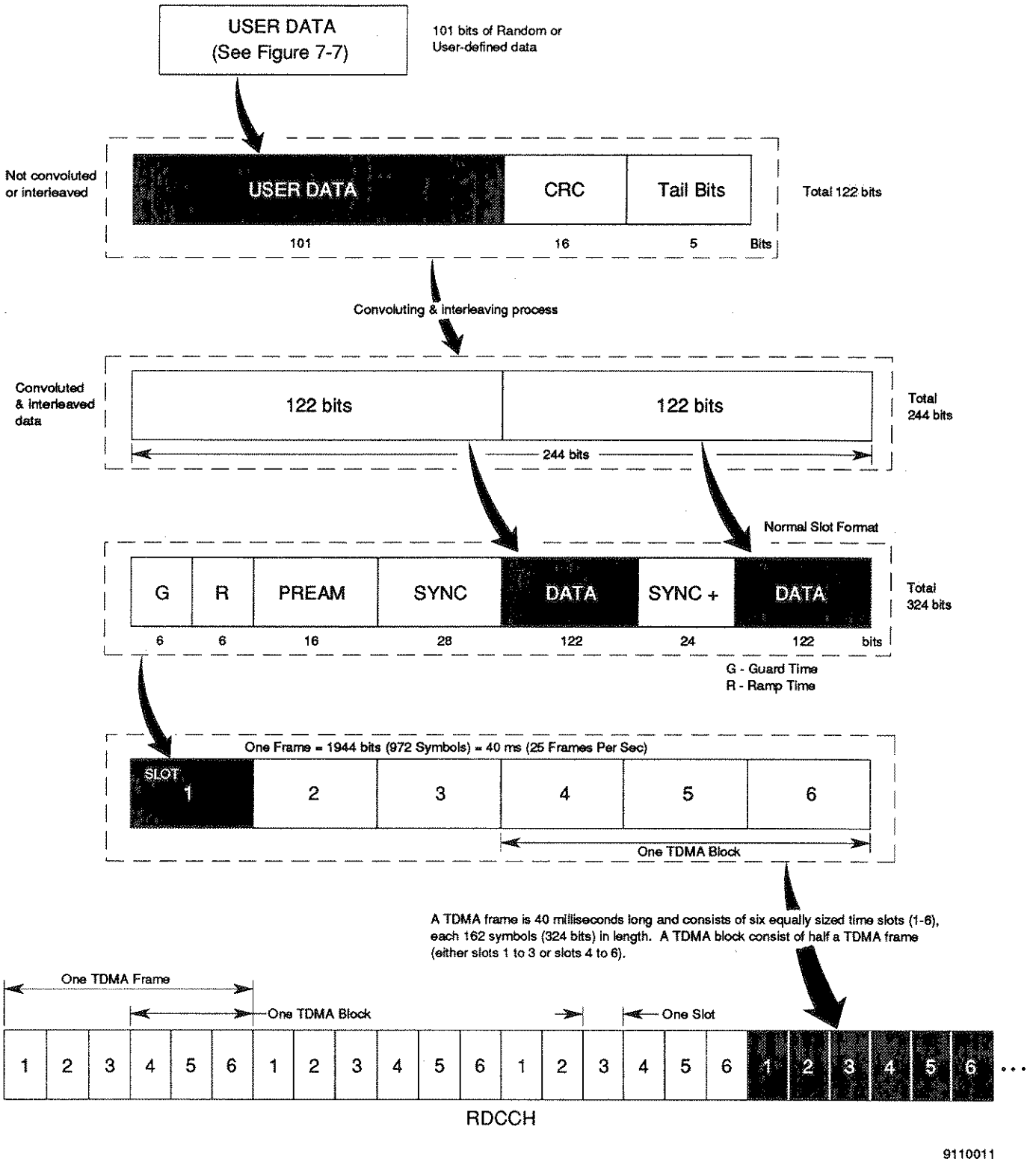


Figure 7-4 Normal Length Message in the RDCCH

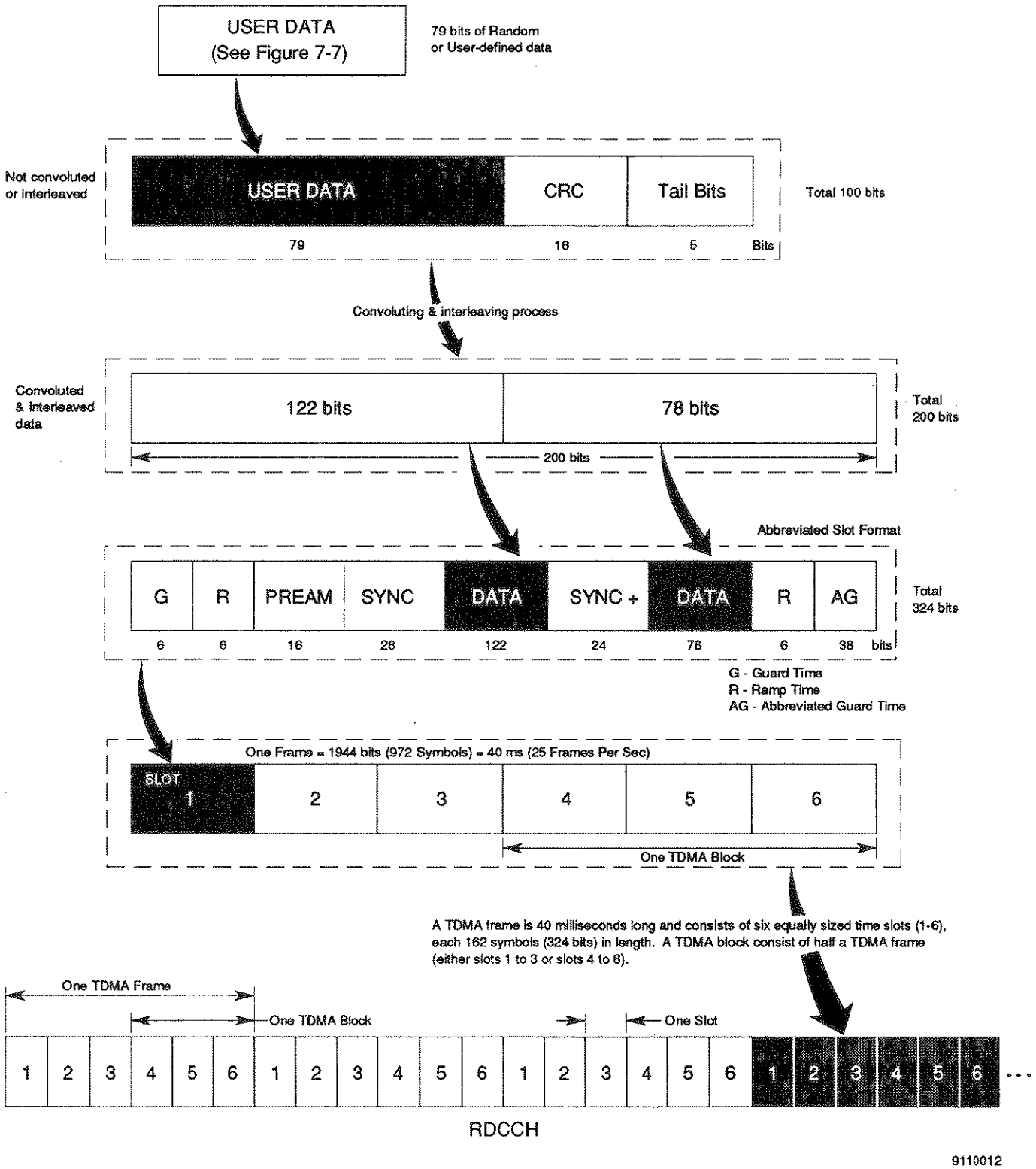


Figure 7-5 Abbreviated Length Message in the RDCCH

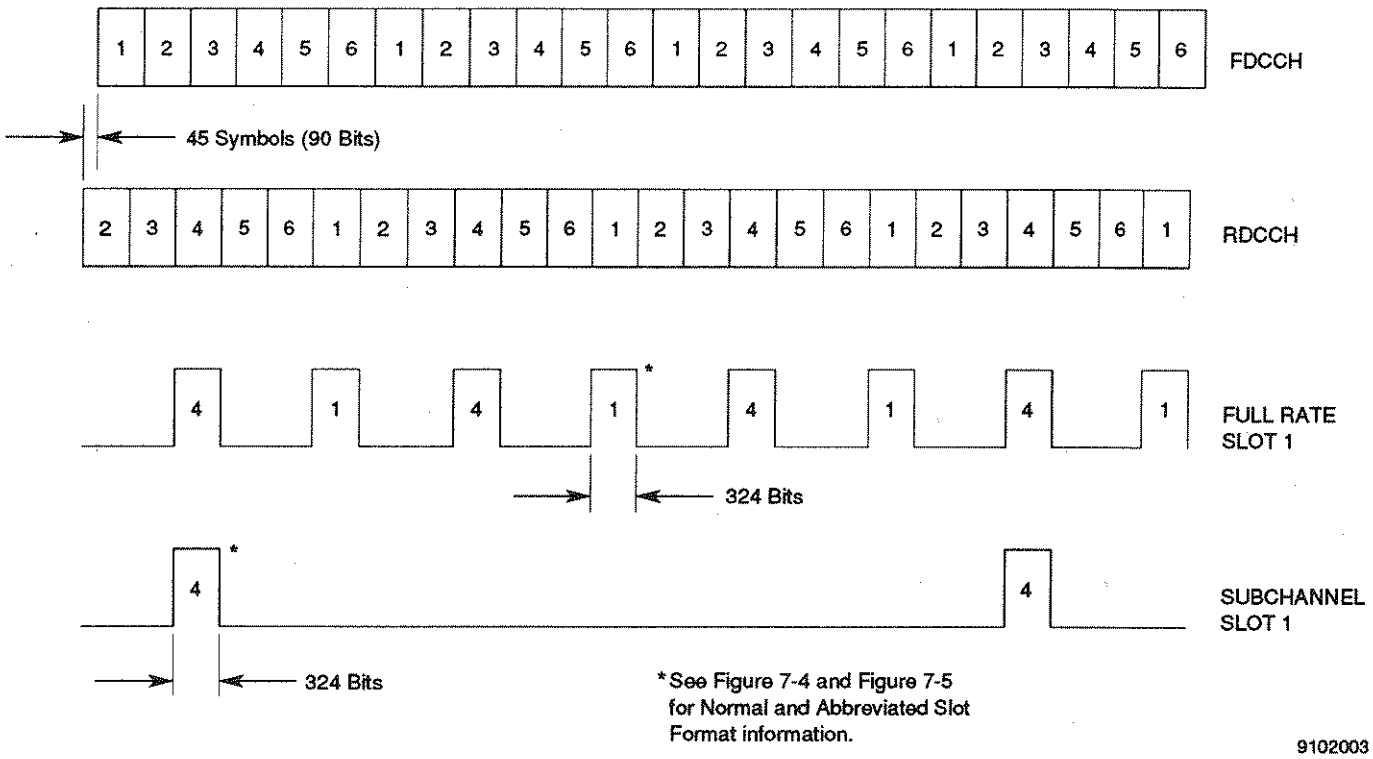


Figure 7-6 Contiguous and Sub Channel Transmissions

7-14-2 SETUP COMMANDS

To set up the FM/AM-1600CSA for Mobile Station Simulation (MSS), use the following TMAC commands:

- **MSS:SETup**
- **MSS:CHANnel *n***

} These two are to be used first in setting up an application

- **MSS:RFLVL *n***
- **MSS:SLOT *n***

MSS:

SETup
[MSS:SETup]
Configures the FM/AM-1600CSA to simulate a Mobile Station.

MSS:SETup performs the following functions to set up the Test Set:

- Places the 1600S into the Duplex Mode of operation.
- Selects the 1600S Modulation as "User Defined." The parameters defined are as follows:
  - Modulation as FM DATA
  - IF Filters as 30 kHz
  - Post Detection as 15 kHz Low-Pass Filter.
- Sets the output level of the 1600S Duplex Out Connector to +7 dBm.
- Selects the 1600S AGC Type as User Defined – High Speed.
- Turns off all 1600S Function Generators
- Sets the 1600S input connector to Antenna.
- Halts all tasks currently executing (i.e. if 1600CSA is decoding FOCC data, then that function is stopped).

MSS:

CHANnel *n*
[MSS:CHANnel]
Selects Reverse Channel on which to transmit. Range of *n* is 1 to 1023.

The 1600S Duplex Operation is set up to receive on the selected forward channel and to transmit out the 1600S Duplex Out Connector at 130 MHz below the selected reverse channel. This output is used as a local oscillator and is mixed with 130 MHz in the 1600CSA and transmitted out the RF I/O Connector on the selected reverse channel.

- Example:

  RDCCH Channel 1 is selected.

  Then, 1600S (in the Duplex mode of operation) receives on a frequency of 870.03 MHz and transmits, via the Duplex Out Connector, on a frequency of 695.03 MHz.

  And, the 1600CSA accepts the 695.03 MHz transmission, via the Duplex In Connector, and mixes with 130 MHz and the resultant frequency is 825.03 MHz.

  When setting up the channel, the Duplex Operation screen frequencies do not change. In addition, if the Duplex Operation is in Channel Mode and the screen is reselected (using the Front Panel of the 1600S), then, the 1600S Channel Mode overrides these channel settings.

CHANnel?
Returns current mobile simulation channel.

RATE *n*
[MSS:RATE]
Selects TDMA transmission rate: full (*n* = 0) or half (*n* = 1).

RATE?
Returns current state of TDMA transmission RATE.

RFLVL *n*
[MSS:RFLVL]
Specifies RF output level at which to transmit. Range of *n* is -127.9 to -30.0 dBm.

SLOT *n*
[MSS:SLOT]
Specifies DCCH Full Rate Slot in which to transmit. Range of *n* is 1 to 3.

SLOT?
Returns current value of DCCH slot selection.

7-14-3 RDCCH RAW GENERATOR

The following commands direct the RDCCH Raw Generator to transmit data on the RDCCH synchronous to the FDCCH. The RDCCH Raw Generator transmits user data in the following methods as specified by the **MSS:RDCCH:DVCC**, **MSS:RDCCH:LENGth**, **MSS:RDCCH:MODE** and **MSS:RDCCH:SELect** commands:

1. Full-Rate TDMA Channel as Normal or Abbreviated transmission bursts
   - Random Data
   - User-Defined Data
2. Within a Random Access Channel (RACH) Sub-Channel as Normal or Abbreviated transmission bursts
   - Random Data
   - User-Defined Data

The setup commands specified in section 7-14-2 are critical and must be taken into consideration when using this function.

The preamble, sync words, and sync+ as specified in IS-136 as well as the Cyclic Redundancy Check (CRC), convolutional encoding and interleaving are performed by the FM/AM-1600CSA.

The data in the DATA field is setup by the user as wither a user-defined pattern or random data.

Use the following commands to specify the Length and Mode of the Reverse Digital Control Channel:

MSS:

RDCCH:

LENGth:

ABBREViated
[MSS:RDCCH:LENGth:ABBREViated]
Selects Abbreviated length transmission bursts on RDCCH.

NORMal
[MSS:RDCCH:LENGth:NORMal]
Selects Normal length transmission bursts on RDCCH.

MODE:

CONTiguous
[MSS:RDCCH:MODE:CONTiguous]
Selects transmission in Full-Rate TDMA Channel.

SUBCHANnel
[MSS:RDCCH:MODE:SUBCHANnel]
Selects transmission in RACH Sub Channel.

MSS:

RDCCH:

SELect:

RANDom
[MSS:RDCCH:SELect:RANDom]
Selects Random Data for DATA Field in transmission bursts on RDCCH.

USER
[MSS:RDCCH:SELect:USER]
Selects a user-defined data pattern for DATA Field (See **MSS:RDCCH:USER**) in transmission bursts on RDCCH.

DVCC ***n***
[MSS:RDCCH:DVCC]
Specifies Digital Verification Color Code. Range of *n* is 1 to 255.

DVCC is required to calculate the correct CRC.

DVCC?
Returns current value of Digital Verification Color Code.

If the data selected is USER, then the data is specified by the following command:

MSS:

RDCCH:

USER *n*, *word*
[MSS:RDCCH:USER]
Specifies each 16 bit word indexed by *n*. Range of *n* is 0 to 6; range of *word* is 0 to #hFFFF.

USER data consists of 7 words (Normal Length) or 5 words (Abbreviated Length). Each word (*n*) consists of 16 bits. *n* = 0 selects the most significant word; *n* = 6 selects the least significant word. The first bit of *word* is the most significant bit, the last bit of *word* is the least significant bit (left justified).

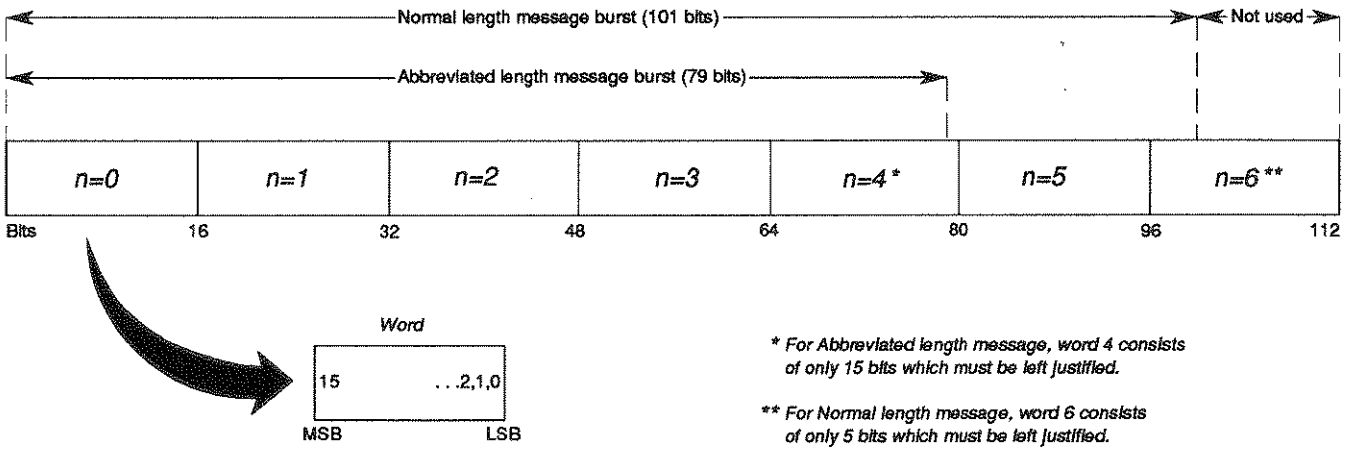


Figure 7-7 User Data Message Diagram

Data left justified is defined as bits aligned such that the most significant bit occurs first, followed by successively less significant bits.

Use the following commands to Start and Stop the data:

STARt
[MSS:RDCCH:STARt]
Starts transmission in RDCCH.

STOP
[MSS:RDCCH:STOP]
Stops transmission in RDCCH.

7-14-4 RANDOM ACCESS CONTROL CHANNEL (RACH) MESSAGE GENERATOR

To simulate a mobile access on the DCCH, the following commands direct the RACH Message Generator to transmit a fixed length, user-defined message as Normal or Abbreviated transmission bursts in a Full-Rate TDMA Channel or within a RACH Sub-channel on the RDCCH synchronous to the FDCCH (See Figure 7-9 and Figure 7-10).

The MSS:RDCCH:MODE, MSS:RDCCH:DVCC, MSS:RDCCH:LENGth and MSS:RDCCH:MESSage:LENGth commands are used to specify the type of transmission.

The setup commands specified in Section 7-14-2 are critical and must be taken into consideration when using this function.

MSS:

RDCCH:

MESSage:

LENGth *n*
[MSS:RDCCH:MESSage:LENGth]
Specifies the length (number of transmission bursts) of RDCCH message. Range of *n* is 1 to 320.

LENGth?
Returns current RDCCH message length.

MSS:

RDCCH:

MESSage:

DATA *m,n,word*
[MSS:RDCCH:MESSage:DATA]
Specifies a 16 bit data word (indexed by *n*) in selected transmission burst (*m*). The range of *m* is 0 to 319; range of *n* is 0 to 6; range of *word* is 0 to #hFFFF.

USER data consists of 1 to 320 transmission bursts. Each burst consists of 7 words (Normal Length) or 5 words (Abbreviated Length). *n* = 0 selects the most significant word; *n* = 6 selects the least significant word. The first bit of *word* is the most significant bit, the last bit of *word* is the least significant bit.

The MSS:RDCCH:MESSage:DATA and the MSS:RDCCH:MESSage:LENGth commands need to be used to fully define a RACH message.

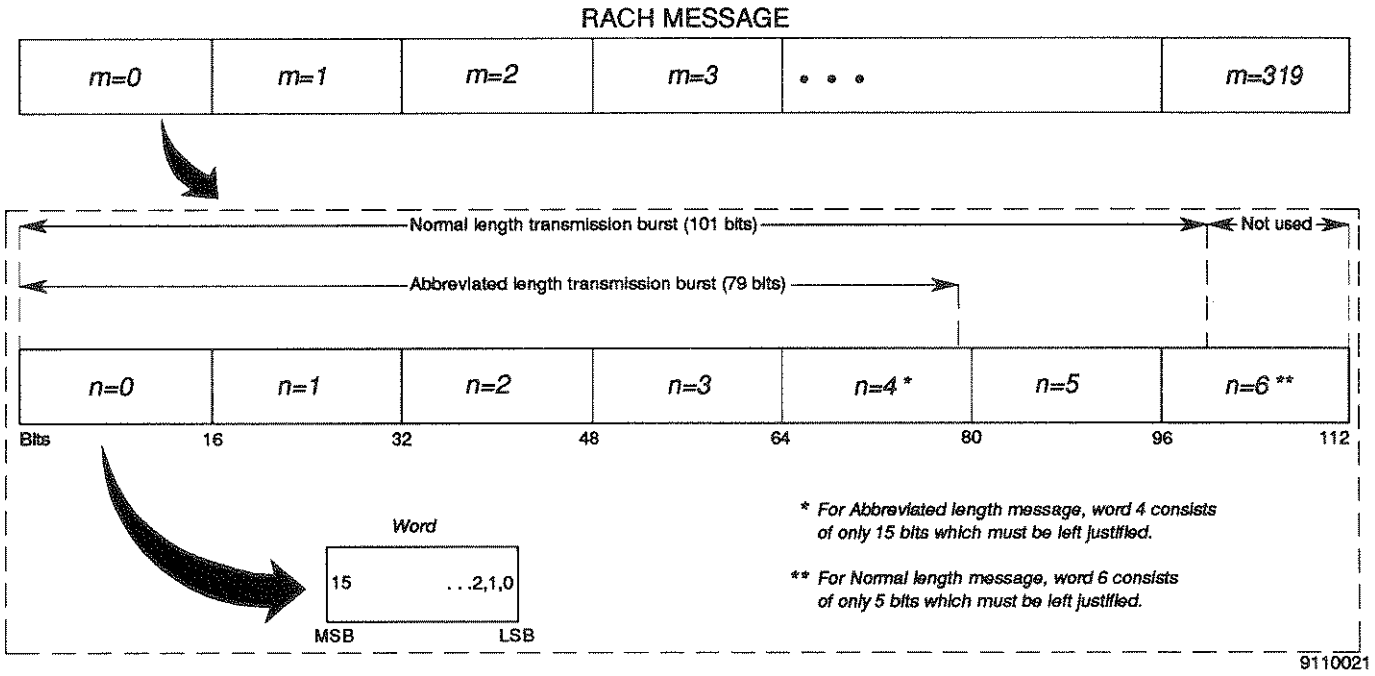


Figure 7-8 RDCCH Message Diagram

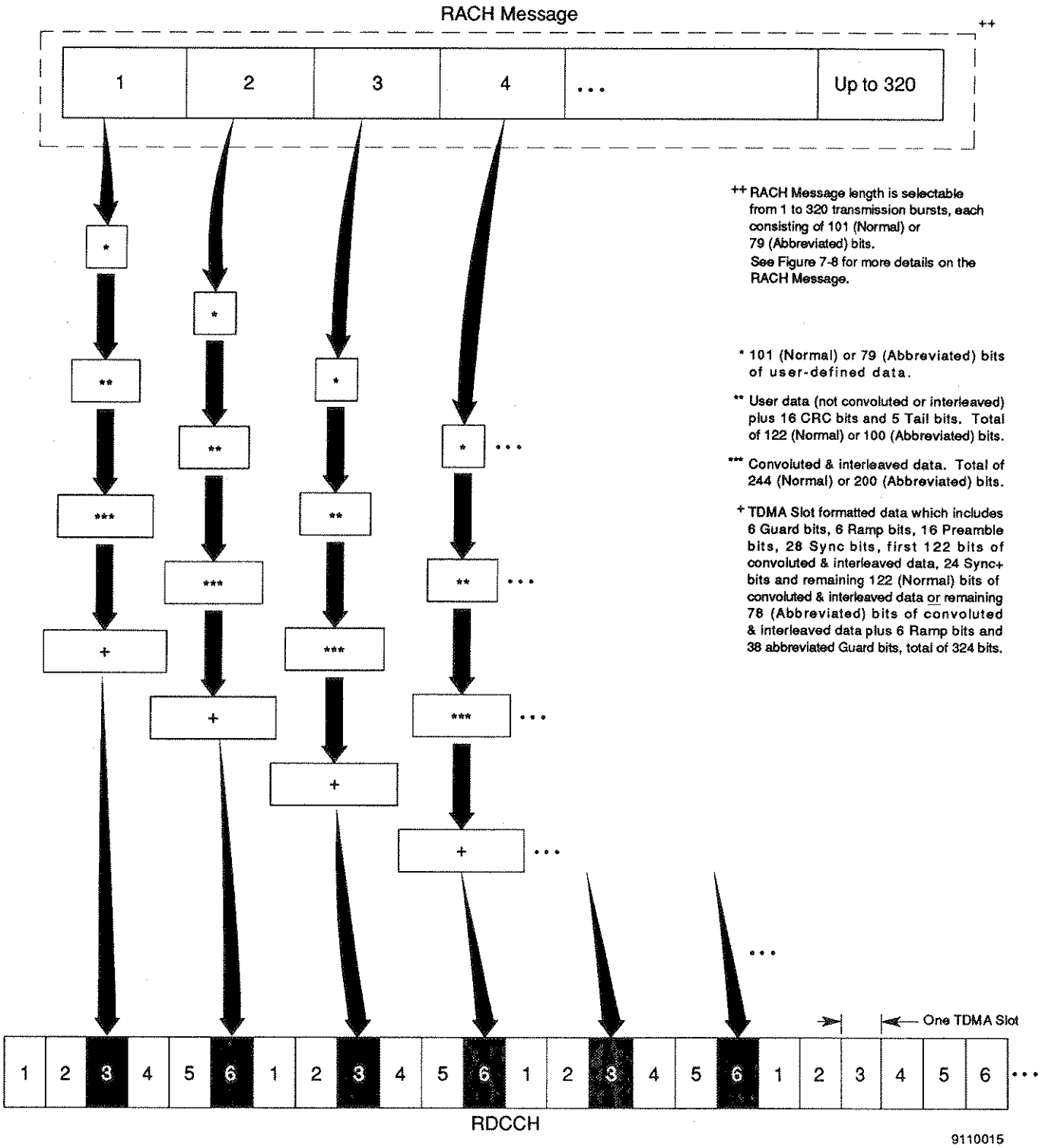


Figure 7-9 RACH Message Transmission (Contiguous)

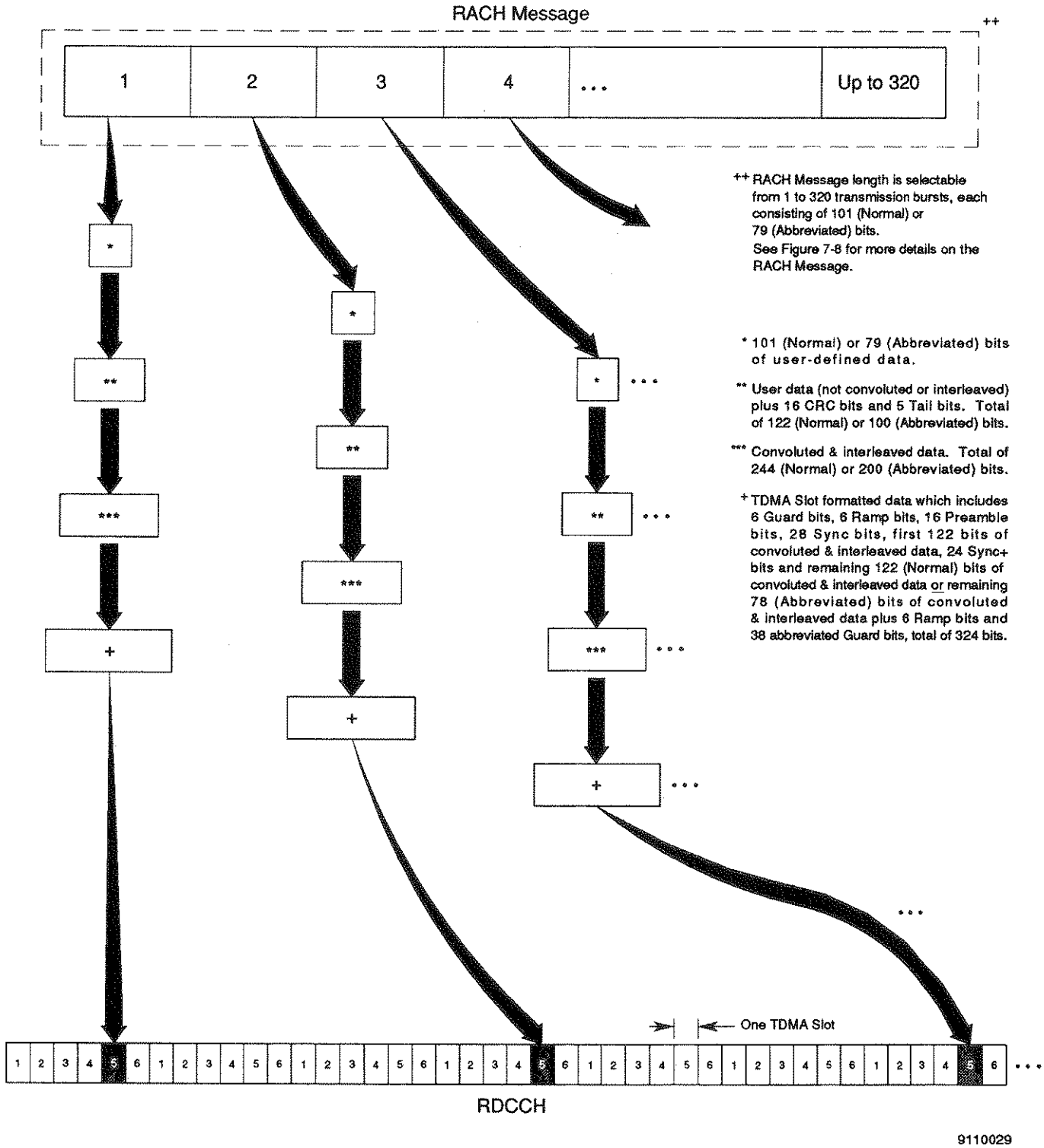


Figure 7-10 RACH Message Transmission (Sub Channel)

MSS:

RDCCH:

MESSage:

ACCESS:

SFP *n*
[MSS:RDCCH:MESSage:ACCESS:SFP]
Selects the Superframe Phase reference for transmission of user-defined message (See **MSS:RDCCH:MESSage:DATA**). Range of *n* is 0 to 31.

The first transmission burst of the RACH message is transmitted synchronous to the slot of the Superframe contained in the Superframe Phase selected by this command.

SFP?
Returns current Superframe Phase reference.

TYPE:

RANDom
[MSS:RDCCH:MESSage:ACCESS:TYPE:RANDom]
Selects Random access.

REServed
[MSS:RDCCH:MESSage:ACCESS:TYPE:REServed]
Selects Reserved access.

SFP
[MSS:RDCCH:MESSage:ACCESS:TYPE:SFP]
Selects Superframe Phase synchronization.

NONE
[MSS:RDCCH:MESSage:ACCESS:TYPE:NONE]
Selects no synchronization.

TYPE?
[MSS:RDCCH:MESSage:ACCESS:TYPE?]
Returns current value of Type of Access.

DELay *n*
[MSS:RDCCH:MESSage:ACCESS:DELay]
Specifies the delay in number of frames before retrying an access after an access failure. Range of *n* is 0 to 20.

DELay?
Returns current number of frames of the re-access delay.

MSS:

RDCCH:

MESSage:

REPeat:

ON
[MSS:RDCCH:MESSage:REPeat:ON]
Sends RACH message at Repeat Rate after initial transmission.

Each repeat of the RACH message is synchronized to the selected Superframe Phase (SFP). See **MSS:RDCCH:MESSage:SFP** command.

OFF
[MSS:RDCCH:MESSage:REPeat:OFF]
Sends RACH message once after initial transmission.

SYNC *n*
[MSS:RDCCH:MESSage:REPeat:SYNC]
Enables (*n* = 1) or disables (*n* = 0) Repeat Synchronizing.

If repeat synchronizing is enabled, then each repeat of the RACH message is synchronized to the designated superframe phase (SFP).

SYNC?
Returns current state of Repeat Synchronizing.

SEND
[MSS:RDCCH:MESSage:SEND]
Sends RACH message once (Repeat is disabled) or repeatedly (Repeat is enabled).

See MSS:RDCCH:MESSage:REPeat.

STOP
[MSS:RDCCH:MESSage:STOP]
Stops sending RACH message.

Applicable only when Repeat is turned on (See MSS:RDCCH:MESSage:REPeat:ON).

CORRUPT *n*
[MSS:RDCCH:MESSage:CORRUPT]
Selects the frame of the RACH message to corrupt. Range of *n* is 0 to 320.

If 0 is selected, then none of the frames of the message are corrupted.

CORRUPT?
Returns current corrupted frame.

7-14-5 RACH LAYER 3 MESSAGE BUILDER

The messages that are sent by the RACH Message Generator can be created with the RACH Layer 3 message builder. The process of building a RACH message can be broken down into five steps.

- Setup the Layer 2 data fields that cannot be determined from the layer 3 messages being sent.
- Select the Message Types that are to be included in the Layer 3 message (1 to 8).
- Set up the Elements that make up the specified messages.
- Enable or disable the optional fields.
- Build the RACH message.

A. Layer 2 Data Fields

Some of the Layer 2 data fields are determined by the Layer 3 message being sent. The ones that cannot be determined are set up by the following commands.

MSS:

RDCCH:

LAYER2:

IDT *n*
[MSS:RDCCH:LAYER2:IDT]
Specifies Identity Type. Range of *n* is 0 to 3.

IDT?
Returns current value of Identity Type.

EHI *n*
[MSS:RDCCH:LAYER2:EHI]
Enables (*n* = 1) or disables (*n* = 0) Extended Header Information.

EHI?
Returns current state of Extended Header Information.

MEA *n*
[MSS:RDCCH:LAYER2:MEA]
Specifies Message Encryption Algorithm. Range of *n* is 0 to 3.

MEA?
Returns current value of Message Encryption Algorithm..

MEK *n*
[MSS:RDCCH:LAYER2:MEK]
Specifies Message Encryption Key. Range of *n* is 0 to 3.

MEK?
Returns current value of Message Encryption Key.

NL3M *n*
[MSS:RDCCH:LAYER2:NL3M]
Specifies Number Layer 3 Messages Range of *n* is 0 to 7.

Besides being part of the Layer 2 frame, this command specifies the number of Layer 3 messages that are built into the RACH message by the MSS:RDCCH:BUILD command.

NL3M?
Returns current value of Number Layer 3 Messages.

MSS:

RDCCH:

LAYER2:

MSID:

MS *n*
[MSS:RDCCH:LAYER2:MSID:MS]
Specifies the 18 Most Significant Bits of Mobile Station Identification. Range of *n* is 0 to #h3FFFF.

MS?
Returns the 18 Most Significant Bits of Mobile Station Identification.

LS *n*
[MSS:RDCCH:LAYER2:MSID:LS]
Sets the 32 Least Significant Bits of Mobile Station Identification. Range of *n* is 0 to #hFFFFFFFF.

LS?
Returns the 32 Least Significant Bits of Mobile Station Identification.

MIN "*n*"
[MSS:RDCCH:LAYER2:MIN]
Specifies Mobile Identification Number. *n* is an ASCII string e.g. "316/522-4981".

This MIN is used to set up the MSID field when the Identity Type (IDT) is equal to 2.

This is for the convenience of the user since the MIN of the phone, and not the MSID, is usually known.

MIN?
Returns current Mobile Identification Number (ASCII string).

ARQ *n*
[MSS:RDCCH:LAYER2:ARQ]
Enables (*n* = 1) or disables (*n* = 0) ARQ status frame.

If ARQ is enabled, then the message built is a Layer 2 ARQ status frame.

ARQ?
Returns current state of ARQ status frame.

PEA *n*
[MSS:RDCCH:LAYER2:PEA]
Specifies Partial Echo Assigned. Range of *n* is 0 to 63.

PEA?
Returns current value of Partial Echo Assigned.

MSS:

RDCCH:

LAYER2:

FRNO *n*
[MSS:RDCCH:LAYER2:FRNO]
Specifies Frame Number Map. Range of *n* is 0 to #hFFFFFFFF.

FRNO?
Returns current value of Frame Number Map.

RSVD:

ARQ *n*
[RDCCH:RSVD:ARQ]
Specifies Automatic Retransmission Request RSVD. Range of *n* is 0 to 3.

ARQ?
Returns current value of ARQ. Returns -1 if not available.

EHI *n*
[RDCCH:RSVD:EHI]
Enables (*n* = 1) or disables (*n* = 0) Extended Header Indicator RSVD.

EHI?
Returns current state of Extended Header Indicator RSVD. Returns -1 if not available.

END *n*
[RDCCH:RSVD:END]
Enables (*n* = 1) or disables (*n* = 0) END frame RSVD.

END?
Returns current state of END frame RSVD. Returns -1 if not available.

The other Layer 2 data fields are setup as determined by the messages being sent in the Layer 3 data field.

BT — If the messages can be sent in one frame then BT = 3 for a BEGIN and END frame. For a multi-frame message, BT = 0 for the BEGIN frame, BT = 1 for the CONTINUE frame, and BT = 2 for the END frame. If the message is an ARQ status frame then BT = 4.

CI — The first CONTINUE frame sent has its CI field set to 0 and toggles for each additional CONTINUE frame. If the SCF information received on the FDCCH determines that a CONTINUE frame needs to be resent, then the CI field remains the same for that frame.

L3LI — This is determined by the length of the specified layer 3 message in octets.

RSVD — All reserved fields are set to zero.

B. RACH Message Types.

A RACH may include from 1 to 8 Layer 3 messages. The NL3M field determines the number of Layer 3 messages included in the RACH. The following commands select the message types sent as Layer 3 messages. The parameter n in each case determines the order and the selected messages types. As an example, if three messages are to be sent (NL3M = 2), then the message types selected with *n* = 0, *n* = 1 and *n* = 2 would be the messages built into the RACH message. These selected messages would be positioned in the RACH message, starting with the message type selected with *n* = 0.

MSS:

RDCCH:

MSGtype:

The user may specify from 1 to 8 Message Types.

AUDITcon *n*
[MSS:RDCCH:MSGtype:AUDITcon]
Selects Audit Confirmation to be transmitted as a Message Type. The position of the message is determined by *n*. Range of *n* is 0 to 7.

AUTHentication *n*
[MSS:RDCCH:MSGtype:AUTHentication]
Selects Authentication to be transmitted as a Message Type. The position of the message is determined by *n*. Range of *n* is 0 to 7.

BSCHAL *n*
[MSS:RDCCH:MSGtype:BSCHAL]
Selects Base Station Challenge Order to be transmitted as a Message Type. The position of the message is determined by *n*. Range of *n* is 0 to 7.

BSMC *n*
[MSS:RDCCH:MSGtype:BSMC]
Selects BSMC Message Delivery to be transmitted as a Message Type. The position of the message is determined by *n*. Range of *n* is 0 to 7.

CAPability *n*
[MSS:RDCCH:MSGtype:CAPability]
Selects Capability Report to be transmitted as a Message Type. The position of the message is determined by *n*. Range of *n* is 0 to 7.

MACA *n*
[MSS:RDCCH:MSGtype:MACA]
Selects MACA Report to be transmitted as a Message Type. The position of the message is determined by *n*. Range of *n* is 0 to 7.

ORIGination *n*
[MSS:RDCCH:MSGtype:ORIGination]
Selects Origination to be transmitted as a Message Type. The position of the message is determined by *n*. Range of *n* is 0 to 7.

MSS:

RDCCH:

MSGtype:

The user may specify from 1 to 8 Message Types.

PAGE_RESPonse *n*
[MSS:RDCCH:MSGtype:PAGE_RESPonse]
Selects Page Response to be transmitted as a Message Type. The position of the message is determined by *n*. Range of *n* is 0 to 7.

RDATA *n*
[MSS:RDCCH:MSGtype:RDATA]
Selects R-DATA to be transmitted as a Message Type. The position of the message is determined by *n*. Range of *n* is 0 to 7.

RDATA:

ACCept *n*
[MSS:RDCCH:MSGtype:RDATA:ACCept]
Selects R-DATA ACCEPT to be transmitted as a Message Type. The position of the message is determined by *n*. Range of *n* is 0 to 7.

REJect *n*
[MSS:RDCCH:MSGtype:RDATA:REJect]
Selects R-DATA REJECT to be transmitted as a Message Type. The position of the message is determined by *n*. Range of *n* is 0 to 7.

REGistration *n*
[MSS:RDCCH:MSGtype:REGistration]
Selects Registration to be transmitted as a Message Type. The position of the message is determined by *n*. Range of *n* is 0 to 7.

SERial *n*
[MSS:RDCCH:MSGtype:SERial]
Selects Serial Number to be transmitted as a Message Type. The position of the message is determined by *n*. Range of *n* is 0 to 7.

SOC *n*
[MSS:RDCCH:MSGtype:SOC]
Selects SOC Message Delivery to be transmitted as a Message Type. The position of the message is determined by *n*. Range of *n* is 0 to 7.

SPACHcon *n*
[MSS:RDCCH:MSGtype:SPACHcon]
Selects SPACH Confirmation to be transmitted as a Message Type. The position of the message is determined by *n*. Range of *n* is 0 to 7.

MSS:

RDCCH:

MSGtype:

The user may specify from 1 to 8 Message Types.

SSDUPcon *n*
[MSS:RDCCH:MSGtype:SSDUPcon]
Selects SSD Update Order Confirmation to be transmitted as a Message Type. The position of the message is determined by *n*. Range of *n* is 0 to 7.

TEST *n*
[MSS:RDCCH:MSGtype:TEST]
Selects Test Registration to be transmitted as a Message Type. The position of the message is determined by *n*. Range of *n* is 0 to 7.

UCHALcon *n*
[MSS:RDCCH:MSGtype:UCHALcon]
Selects Unique Challenge Order Confirmation to be transmitted as a Message Type. The position of the message is determined by *n*. Range of *n* is 0 to 7.

C. RACH Layer 3 Data Fields

The values of the elements that make up the Layer 3 messages are determined by the commands in this section. Generally, the name of each command closely matches or is an abbreviation or acronym of the associated data element specified in IS-136.

MSS:

RDCCH:

PD *n*
[MSS:RDCCH:PD]
Specifies Protocol Discriminator. Range of *n* is 0 to 3.

PD?
Returns current value of Protocol Discriminator.

PFC_1 *n*
[MSS:RDCCH:PFC]
Specifies Paging Frame Class Minus One. Range of *n* is 0 to 7.

PFC_1?
Returns current value of Paging Frame Class.

PSID_RSID:

SELect *n*
[MSS:RDCCH:PSID_RSID:SELect]
Specifies Selected PSID/RSID (Private System Identification/Residential System Identification). Range of *n* is 0 to 15.

SELect?
Returns current value of Selected PSID/RSID.

MAP *n*
[MSS:RDCCH:PSID_RSID:MAP]
Specifies PSID/RSID Map. Range of *n* is 0 to #hFFFF.

MAP?
Returns current value of PSID/RSID Map.

DISPlay:

LENGth *n*
[MSS:RDCCH:DISPlay:LENGth]
Specifies Length of Display Information. Range of *n* is 0 to 81.

LENGth?
Returns current value of Length of Display Information.

MSS:

RDCCH:

DISPlay:

CHARacter *m,n*
[MSS:RDCCH:DISPlay:CHARacter]
Specifies Display Character (*n*) selected by Index (*m*). Range of *m* is 0 to 81; range of *n* is 0 to 255.

The characters are setup one at a time.

CHARacter? *n*
Returns current value of Display Character selected by *n*. Range of *n* is 0 to 81.

COUNt *n*
[MSS:RDCCH:COUNt]
Specifies COUNT used for the Authentication process. Range of *n* is 0 to 63.

COUNt?
Returns current value of COUNt.

RANDC *n*
[MSS:RDCCH:RANDC]
Specifies RANDC used in the Authentication process. Range of *n* is 0 to 255.

RANDC?
Returns current value of RANDC.

AUTHR *n*
[MSS:RDCCH:AUTHR]
Specifies AUTHR used in the Authentication process. Range of *n* is 0 to #h3FFFF.

AUTHR?
Returns current value of AUTHR.

RANDBS *n*
[MSS:RDCCH:RANDBS]
Specifies RANDBS used in the Authentication process. Range of *n* is 0 to #hFFFFFFFF.

RANDBS?
Returns current value of RANDBS.

BSMC *n*
[MSS:RDCCH:BSMC]
Specifies Base Station Manufacture Code. Range of *n* is 0 to 255.

BSMC?
Returns current value of Base Station Manufacture Code.

MSS:

RDCCH:

PROTocol:

VERsion *n*
[MSS:RDCCH:PROTocol:VERsion]
Specifies Protocol Version. Range of *n* is 0 to 15.

VERsion?
Returns current value of Protocol Version.

SCM *n*
[MSS:RDCCH:SCM]
Specifies Station Class Mark. Range of *n* is 0 to 31.

SCM?
Returns current value of Station Class Mark.

VINtage:

SOFTware *n*
[MSS:RDCCH:VINtage:SOFTware]
Specifies Software Vintage. Range of *n* is 0 to 63.

SOFTware?
Returns current value of Software Vintage.

FIRMware *n*
[MSS:RDCCH:VINtage:FIRMware]
Specifies Firmware Vintage. Range of *n* is 0 to 63.

FIRMware?
Returns current value of Firmware Vintage.

MODEL *n*
[MSS:RDCCH:MODEL]
Specifies Model Number. Range of *n* is 0 to 15.

MODEL?
Returns current value of Model Number.

MANufacture *n*
[MSS:RDCCH:MANufacture]
Specifies Manufacture Code. Range of *n* is 0 to 255.

MANufacture?
Returns current value of Manufacture Code.

MSS:

RDCCH:

SUPPort:

MAX:

PFC *n*
[MSS:RDCCH:SUPPort:MAX:PFC]
Specifies MAX_SUPPORTED_PFC. Range of *n* is 0 to 7.

PFC?
Returns current value of MAX_SUPPORTED_PFC.

SOC *n*
[MSS:RDCCH:SUPPort:SOC]
Enables (*n* = 1) or disables (*n* = 0) SOC Support.

SOC?
Returns current state of SOC Support.

BSMC *n*
[MSS:RDCCH:SUPPort:BSMC]
Enables (*n* = 1) or disables (*n* = 0) BSMC Support.

BSMC?
Returns current state of BSCM Support.

ASYNC *n*
[MSS:RDCCH:SUPPort:ASYNC]
Enables (*n* = 1) or disables (*n* = 0) Async Data Support.

ASYNC?
Returns current state of Async Data Support.

G3fax *n*
[MSS:RDCCH:SUPPort:G3fax]
Enables (*n* = 1) or disables (*n* = 0) G3-Fax Support.

G3fax?
Returns current state of G3-Fax Support.

SMS *n*
[MSS:RDCCH:SUPPort:SMS]
Enables (*n* = 1) or disables (*n* = 0) SMS Broadcast Support.

SMS?
Returns current state of SMS Broadcast Support.

MSS:

RDCCH:

SUPPort:

SUBaddress *n*
[MSS:RDCCH:SUPPort:SUBaddress]
Enables (*n* = 1) or disables (*n* = 0) Subaddressing Support.

SUBaddress?
Returns current state of Subaddressing Support.

FREQuency:

BANDS *n*
[MSS:RDCCH:SUPPort:FREQuency:BANDS]
Specifies Supported Frequency Bands. Range of *n* is 0 to 255.

BANDS?
Returns current value of Supported Frequency Bands.

IA5 *n*
[MSS:RDCCH:SUPPort:IA5]
Enables (*n* = 1) or disables (*n* = 0) IA5 Support.

IA5?
Returns current state of IA5 Support.

USER *n*
[MSS:RDCCH:SUPPort:USER]
Enables (*n* = 1) or disables (*n* = 0) User Group Support.

USER?
Returns current state of User Group Support.

ANA800 *n*
[MSS:RDCCH:SUPPort:ANA800]
Enables (*n* = 1) or disables (*n* = 0) 800 MHz Analog Speech Support.

Indicates whether or not the mobile station supports analog speech in the 800 MHz Hyperband.

ANA800?
Returns current state of ANA800.

MSS:

RDCCH:

SUPPort:

HALF *n*
[MSS:RDCCH:SUPPort:HALF]
Enables (*n* = 1) or disables (*n* = 0) Half-Rate DTC Support.

Indicates whether or not the mobile station supports half-rate Digital Traffic Channels.

HALF?
Returns current state of HALF.

DOUBle *n*
[MSS:RDCCH:SUPPort:DOUBle]
Enables (*n* = 1) or disables (*n* = 0) Double Rate DTC Support.

Indicates whether or not the mobile station supports double rate Digital Traffic Channels.

DOUBle?
Returns current state of DOUBle.

TRIPle *n*
[MSS:RDCCH:SUPPort:TRIPle]
Enables (*n* = 1) or disables (*n* = 0) Triple Rate DTC Support.

Indicates whether or not the mobile station supports triple rate Digital Traffic Channels.

TRIPle?
Returns current state of TRIPle.

MEASurement:

LTM:

WER *n*
[MSS:RDCCH:MEASurement:LTM:WER]
Specifies LTM Measurement Word Error Rate. Range of *n* is 0 to 7.

WER?
Returns current value of LTM Measurement Word Error Rate.

MSS:

RDCCH:

MEASurement:

LTM:

BER *n*
[MSS:RDCCH:MEASurement:LTM:BER]
Specifies Word Error Rate LTM Measurement Bit Error Rate. Range of *n* is 0 to 7.

BER?
Returns current value of LTM Measurement Bit Error Rate.

RSS *n*
[MSS:RDCCH:MEASurement:LTM:RSS]
Specifies LTM Measurement Receive Signal Strength. Range of *n* is 0 to 7.

RSS?
Returns current value of LTM Measurement Receive Signal Strength.

FULL *n*
[MSS:RDCCH:MEASurement:LTM:FULL]
Enables (*n* = 1) or disables (*n* = 0) LTM Measurement Full Measurement Indicator.

FULL?
Returns current state of LTM Measurement Full Measurement Indicator.

STM:

NV *n*
[MSS:RDCCH:MEASurement:STM:NV]
Specifies STM Measurement Number of Values. Range of *n* is 0 to 15.

NV?
Returns current value of STM Measurement Number of Values.

RSS *m,n*
[MSS:RDCCH:MEASurement:STM:RSS]
Specifies value of ST_RSS (*n*) selected by *m*. Range of *m* is 0 to 15; Range of *n* is 0 to 31.

Specifies 1 of 16 STM Measurement Receive Signal Strengths.

RSS? *n*
Returns current value of STM Measurement Receive Signal Strength selected by *n*. Range of *n* is 0 to 15.

MSS:

RDCCH:

EMERgency *n*
[MSS:RDCCH:EMERgency]
Enables (*n* = 1) or disables (*n* = 0) Emergency Call.

EMERgency?
Returns current state of Emergency Call.

LT *n*
[MSS:RDCCH:LT]
Enables (*n* = 1) or disables (*n* = 0) Last Try flag.

If a mobile station receives a Directed Retry and attempts a new access on another DCCH, it shall set the Last Try flag in the Origination or the Page Response message to the value of the Last Try flag received in the Directed Retry message. Otherwise the mobile station shall reset the Last Try flag at system access.

LT?
Returns current state of Last Try flag.

SERVice *n*
[MSS:RDCCH:SERVice]
Specifies Service Code. Range of *n* is 0 to 15.

SERVice?
Returns current value of Service Code.

MODE:

VOICe:

VC *n*
[MSS:RDCCH:MODE:VOICe:VC]
Specifies Voice Mode VC. Range of *n* is 0 to 7.

VC?
Returns current value of Voice Mode VC.

PM *n*
[MSS:RDCCH:MODE:VOICe:PM]
Specifies Voice Mode PM_V. Range of *n* is 0 to 7.

PM?
Returns current value of Voice Mode PM_V.

MSS:

RDCCH:

MODE:

DATA:

PM *n*
[MSS:RDCCH:MODE:DATA:PM]
Specifies Data Mode PM_D. Range of *n* is 0 to 7.

PM?
Returns current value of Data Mode PM_D.

SAP *n*
[MSS:RDCCH:MODE:DATA:SAP]
Enables (*n* = 1) or disables (*n* = 0) Data Mode SAP.

SAP?
Returns current state of Data Mode SAP.

ACKED *n*
[MSS:RDCCH:MODE:DATA:ACKED]
Enables (*n* = 1) or disables (*n* = 0) Data Mode Acked Data.

ACKED?
Returns current state of Data Mode Acked Data.

CRC *n*
[MSS:RDCCH:MODE:DATA:CRC]
Enables (*n* = 1) or disables (*n* = 0) Data Mode CRC.

CRC?
Returns current state of Data Mode CRC.

PART *n*
[MSS:RDCCH:MODE:DATA:PART]
Enables (*n* = 1) or disables (*n* = 0) Data Mode Data Part.

PART?
Returns current value of Data Mode Data Part.

RLP *n*
[MSS:RDCCH:MODE:DATA:RLP]
Specifies Data Mode RLP. Range of *n* is 0 to 3.

RLP?
Returns current value of Data Mode RLP.

MSS:

RDCCH:

MEM:

Message Encryption Mode - Identifies the selected message encryption algorithm, key and domain.

MEA *n*
[MSS:RDCCH:MEM:MEA]
Specifies Message Encryption Algorithm. Range of *n* is 0 to 7.

MEA?
Returns current value of Message Encryption Algorithm.

MED *n*
[MSS:RDCCH:MEM:MED]
Specifies Message Encryption Domain. Range of *n* is 0 to 7.

MED?
Returns current value of Message Encryption Domain.

MEK *n*
[MSS:RDCCH:MEM:MEK]
Specifies Message Encryption Key. Range of *n* is 0 to 7.

MEK?
Returns current value of Message Encryption Key.

BANDWidth *n*
[MSS:RDCCH:BANDWidth]
Specifies Bandwidth. Range of *n* is 0 to 7.

BANDWidth?
Returns current value of Bandwidth.

CALLED:

ADDRess "*n***"**
[MSS:RDCCH:CALLED:ADDRess]
Specifies Called Address (ASCII String).

ADDRess?
Returns current ASCII string value of Called Address.

MSS:

RDCCH:

CALLED:

ADDRess:

ENCoding *n*
[MSS:RDCCH:CALLED:ADDRess:ENCoding]
Enables (*n* = 1) or disables (*n* = 0) Called Address Encoding.

ENCoding?
Returns current state of Called Address Encoding.

PLANid *n*
[MSS:RDCCH:CALLED:PLANid]
Specifies Called Address Numbering Plan Identification. Range of *n* is 0 to 15.

PLANid?
Returns current value of Called Address Numbering Plan Identification.

TYPE *n*
[MSS:RDCCH:CALLED:TYPE]
Specifies Called Address Type of Number. Range of *n* is 0 to 7.

TYPE?
Returns current value of Called Address Type of Number.

SUBaddress:

ODD_EVEN *n*
[MSS:RDCCH:CALLED:SUBaddress:ODD_EVEN]
Enables (*n* = 1) or disables (*n* = 0) Called Party Subaddress Odd/Even indicator.

ODD_EVEN?
Returns current state of Called Party Subaddress Odd/Even indicator.

TYPE *n*
[MSS:RDCCH:CALLED:SUBaddress:TYPE]
Specifies Called Party Subaddress Type of Subaddress. Range of *n* is 0 to 3.

TYPE?
Returns current value of Called Party Subaddress Type of Subaddress.

MSS:

RDCCH:

CALLED:

SUBaddress:

ADDRess *m,n*
[MSS:RDCCH:CALLED:SUBaddress:ADDRess]
Specifies Called Party Subaddress (*n*) selected by *m*. Range of *n* is 0 to 19; range of *m* is 0 to 255.

ADDRess? *n*
Returns current value of Called Party Subaddress selected by *n*. Range of *n* is 0 to 255.

CALLING:

ADDRess "*n*"
[MSS:RDCCH:CALLING:ADDRess]
Specifies Calling Address (ASCII String).

ADDRess?
Returns current ASCII string value of Calling Address.

ADDRess:

ENCoding *n*
[MSS:RDCCH:CALLING:ADDRess:ENCoding]
Enables (*n* = 1) or disables (*n* = 0) Calling Address Encoding.

ENCoding?
Returns current state of Calling Address Encoding.

PRESentation:

PI *n*
[MSS:RDCCH:CALLING:PRESentation:PI]
Specifies Calling Address Presentation Indicator. Range of *n* is 0 to 3.

PI?
Returns current value of Calling Address Presentation Indicator.

MSS:

RDCCH:

CALLING:

PRESentation:

SI *n*
[MSS:RDCCH:CALLING:PRESentation:SI]
Specifies Calling Address Screening Indicator. Range of *n* is 0 to 3.

SI?
Returns current value of Calling Address Screening Indicator.

PLANid *n*
[MSS:RDCCH:CALLING:PLANid]
Specifies Calling Address Numbering Plan Identification. Range of *n* is 0 to 15.

PLANid?
Returns current value of Calling Address Numbering Plan Identification.

TYPE *n*
[MSS:RDCCH:CALLING:TYPE]
Specifies Calling Address Type of Number. Range of *n* is 0 to 7.

TYPE?
Returns current value of Calling Address Type of Number.

SUBaddress:

LENGth *n*
[MSS:RDCCH:CALLING:SUBaddress:LENGth]
Specifies Calling Address Subaddress Length. Range of *n* is 1 to 21.

LENGth?
Returns current value of Calling Address Subaddress Length.

ODD_EVEN *n*
[MSS:RDCCH:CALLING:SUBaddress:ODD_EVEN]
Enables (*n* = 1) or disables (*n* = 0) Calling Address Subaddress Odd/Even indicator.

ODD_EVEN?
Returns current state of Calling Address Subaddress Odd/Even indicator.

TYPE *n*
[MSS:RDCCH:CALLING:SUBaddress:TYPE]
Specifies Calling Address Subaddress Type of Number. Range of *n* is 0 to 3.

TYPE?
Returns current value of Calling Address Subaddress Type of Number.

MSS:

RDCCH:

CALLING:

SUBaddress:

ADDRess *n,m*
[MSS:RDCCH:CALLING:SUBaddress:ADDRess]
Specifies Calling Address Subaddress (*m*) selected by *n*. Range of *n* is 0 to 19; range of *m* is 0 to 255.

ADDRess? *n*
Returns current value of Calling Address Subaddress selected by *n*. Range of *n* is 0 to 19.

SUBaddress:

LENGth *n*
[MSS:RDCCH:SUBaddress:LENGth]
Specifies Subaddress Length. Range of *n* is 1 to 21.

LENGth?
Returns current value of Subaddress Length.

ODD_EVEN *n*
[MSS:RDCCH:SUBaddress:ODD_EVEN]
Enables (*n* = 1) or disables (*n* = 0) Subaddress Odd/Even indicator.

ODD_EVEN?
Returns current state of Subaddress Odd/Even indicator.

TYPE *n*
[MSS:RDCCH:SUBaddress:TYPE]
Specifies Type of subaddress. Range of *n* is 0 to 3.

TYPE?
Returns current value of Type of subaddress.

ADDRess *n,m*
[MSS:RDCCH:SUBaddress:ADDRess]
Specifies Subaddress (*m*) selected by *n*. Range of *n* is 0 to 19; range of *m* is 0 to 255.

ADDRess? n
Returns current value of Subaddress selected by *n*. Range of n is 0 to 19.

MSS:

RDCCH:

RTRANSaction *n*
[MSS:RDCCH:RTRANSaction]
Specifies R-Transaction Identifier. Range of *n* is 0 to 255.

RTRANSaction?
Returns current value of R-Transaction Identifier.

RDATA_UNIT:

LENGth *n*
[MSS:RDCCH:RDATA_UNIT:LENGth]
Specifies R-Data Unit Length Indicator. Range of *n* is 0 to 255.

LENGth?
Returns current value of R-Data Unit Length Indicator.

HLP:

IDentifier *n*
[MSS:RDCCH:RDATA_UNIT:HLP:IDentifier]
Specifies R-Data Unit Higher Protocol Identifier. Range of *n* is 0 to 255.

IDentifier?
Returns current value of R-Data Unit Higher Protocol Identifier.

DATA *m,n*
[MSS:RDCCH:RDATA_UNIT:HLP:DATA]
Specifies R-Data Higher Layer Protocol Data Unit (*n*) selected by *m*. Range of *m* is 0 to 255; range of *n* is 0 to 255.

DATA? *n*
Returns current value of Higher Layer Protocol Data Unit selected by *n*. Range of *n* is 0 to 255.

RCAUSE *n*
[MSS:RDCCH:RCAUSE]
Specifies R-CAUSE. Range of *n* is 0 to 127.

RCAUSE?
Returns current value of R-CAUSE.

MSS:

RDCCH:

MESSage:

CENTer:

ADDRess "*n*"
[MSS:RDCCH:MESSage:CENTer:ADDRess]
Specifies Message Center Address (ASCII String).

ADDRess?
Returns current ASCII string value of Message Center Address.

ADDRess:

ENCoding *n*
[MSS:RDCCH:MESSage:CENTer:ADDRess:ENCoding]
Enables (*n* = 1) or disables (*n* = 0) Message Center Address Encoding.

ENCoding?
Returns current state of Message Center Address Encoding.

PLANid *n*
[MSS:RDCCH:MESSage:CENTer:PLANid]
Specifies Message Center Identification Plan. Range of *n* is 0 to 15.

PLANid?
Returns current value of Message Center Identification Plan.

TYPE *n*
[MSS:RDCCH:MESSage:CENTer:TYPE]
Specifies Message Center Type of Number. Range of *n* is 0 to 7.

TYPE?
Returns current value of Message Center Type of Number.

USER:

GROUP:

STATus *n*
[MSS:RDCCH:USER:GROUP:STATus]
Specifies User Group Status. Range of *n* is 0 to 3.

STATus?
Returns current value of User Group Status.

MSS:

RDCCH:

USER:

GROUP:

TYPE *n*
[MSS:RDCCH:USER:GROUP:TYPE]
Specifies User Group Type. Range of *n* is 0 to 3.

TYPE?
Returns current value of User Group Type.

UGID:

MS *n*
[MSS:RDCCH:USER:GROUP:UGID:MS]
Specifies the 18 most significant bits of User Group ID. Range of *n* is 0 to #h3FFFF.

MS?
Returns the 18 most significant bits of User Group ID.

LS *n*
[MSS:RDCCH:USER:GROUP:UGID:LS]
Specifies the 32 least significant bits of User Group ID. Range of *n* is 0 to #hFFFFFFFF.

LS?
Returns the 32 least significant bits of User Group ID.

MIN "*n*"
[MSS:RDCCH:USER:MIN]

If the User Group type is 2, signifying a 34-bit UGID, then this command sets up the User Group ID with a Mobile Identification Number.

Specifies MIN (ASCII String).

MIN?
Returns current string value of MIN.

DEST:

ADDRess "*n*"
[MSS:RDCCH:DEST:ADDRess]
Specifies User Destination Address (ASCII String).

ADDRess?
Returns current string value of User Destination Address.

MSS:

RDCCH:

DEST:

ADDRess:

ENCoding *n*
[MSS:RDCCH:DEST:ADDRess:ENCoding]
Enables (*n* = 1) or disables (*n* = 0) User Destination Address Encoding.

ENCoding?
Returns current state of User Destination Address Encoding.

PLANid *n*
[MSS:RDCCH:DEST:PLANid]
Specifies User Destination Identification Plan. Range of *n* is 0 to 15.

PLANid?
Returns current value of the User Destination Identification Plan.

TYPE *n*
[MSS:RDCCH:DEST:TYPE]
Specifies User Destination Type of Number. Range of *n* is 0 to 7.

TYPE?
Returns current value of User Destination Type of Number.

SUBaddress:

LENGth *n*
[MSS:RDCCH:DEST:SUBaddress:LENGth]
Specifies User Destination Subaddress Length. Range of *n* is 1 to 21.

LENGth?
Returns current value of User Destination Subaddress Length.

ODD_EVEN *n*
[MSS:RDCCH:DEST:SUBaddress:ODD_EVEN]
Enables (*n* = 1) or disables (*n* = 0) User Destination Subaddress Odd/Even indicator.

ODD_EVEN?
Returns current state of User Destination Subaddress Odd/Even indicator.

TYPE *n*
[MSS:RDCCH:DEST:SUBaddress:TYPE]
Specifies User Destination Subaddress Type of Number. Range of *n* is 0 to 3.

TYPE?
Returns current value of User Destination Type of Number.

MSS:

RDCCH:

DEST:

SUBaddress:

ADDRess *n,m*
[MSS:RDCCH:DEST:SUBaddress:ADDRess]
Specifies User Destination Subaddress (*m*) selected by *n*. Range of *n* is 0 to 19; range of *m* is 0 to 255.

ADDRess? *n*
Returns current value of User Destination Subaddress selected by *n*. Range of *n* is 0 to 19.

ORIG:

ADDRess "*n*"
[MSS:RDCCH:ORIG:ADDRess]
Specifies User Originating Address (ASCII String).

ADDRess?
Returns current string value of User Originating Address.

ADDRess:

ENCoding *n*
[MSS:RDCCH:ORIG:ADDRess:ENCoding]
Enables (*n* = 1) or disables (*n* = 0) User Originating Address Encoding.

ENCoding?
Returns current state of User Originating Address Encoding.

PLANid *n*
[MSS:RDCCH:ORIG:PLANid]
Specifies User Originating Identification Plan. Range of *n* is 0 to 15.

PLANid?
Returns current value of User Originating Identification Plan.

TYPE *n*
[MSS:RDCCH:ORIG:TYPE]
Specifies User Originating Type of Number. Range of *n* is 0 to 7.

TYPE?
Returns current value of User Originating Type of Number.

MSS:

RDCCH:

ORIG:

SUBaddress:

LENGth *n*
[MSS:RDCCH:ORIG:SUBaddress:LENGth]
Specifies User Originating Subaddress Length. Range of *n* is 1 to 21.

LENGth?
Returns current value of User Originating Subaddress Length.

ODD_EVEN *n*
[MSS:RDCCH:ORIG:SUBaddress:ODD_EVEN]
Enables (*n* = 1) or disables (*n* = 0) User Originating Subaddress Odd/Even indicator.

ODD_EVEN?
Returns current state of User Originating Subaddress Odd/Even indicator.

TYPE *n*
[MSS:RDCCH:ORIG:SUBaddress:TYPE]
Specifies User Originating Subaddress Type of Number. Range of *n* is 0 to 3.

TYPE?
Returns current value of User Originating Subaddress Type of Number.

ADDRess *n,m*
[MSS:RDCCH:ORIG:SUBaddress:ADDRess]
Specifies User Originating Subaddress (*m*) selected by *n*. Range of *n* is 0 to 19; range of *m* is 0 to 255.

ADDRess? n
Returns current value of User Originating Subaddress selected by *n*. Range of *n* is 0 to 19.

REG:

TYPE *n*
[MSS:RDCCH:REG:TYPE]
Specifies Registration Type. Range of *n* is 0 to 15.

TYPE?
Returns current value of Registration Type.

MSS:

RDCCH:

CNUMber:

ADDRess "*n*"
[MSS:RDCCH:CNUMber:ADDRess]
Specifies C-Number Address (ASCII String).

ADDRess?
Returns current string value of C-Number Address.

ADDRess:

ENCoding *n*
[MSS:RDCCH:CNUMber:ADDRess:ENCoding]
Enables (*n* = 1) or disables (*n* = 0) C-Number Address Encoding.

ENCoding?
Returns current state of the C-Number Address Encoding.

PLANid *n*
[MSS:RDCCH:CNUMber:PLANid]
Specifies C-Number Identification Plan. Range of *n* is 0 to 15.

PLANid?
Returns current value of the C-Number Identification Plan.

TYPE *n*
[MSS:RDCCH:CNUMber:TYPE]
Specifies C-Number Type of Number. Range of *n* is 0 to 7.

TYPE?
Returns current value of the C-Number Type of Number.

PFC:

REQuest *n*
[MSS:RDCCH:PFC:REQuest]
Specifies Paging Frame Class Request. Range of *n* is 0 to 7.

REQuest?
Returns current value of Paging Frame Class Request.

DCCH_MEM:

ALGORithm *n*
[MSS:RDCCH:DCCH_MEM:ALGORithm]
Specifies DCCH Message Encryption Algorithm. Range of *n* is 0 to 7.

ALGORithm?
Returns current value of the DCCH Message Encryption Algorithm

MSS:

RDCCH:

DCCH_MEM:

DOMAIN *n*
[MSS:RDCCH:DCCH_MEM:DOMAIN]
Specifies DCCH Message Encryption Domain. Range of *n* is 0 to 7.

DOMAIN?
Returns current value of DCCH Message Encryption Domain.

KEY *n*
[MSS:RDCCH:DCCH_MEM:KEY]
Specifies DCCH Message Encryption Key. Range of *n* is 0 to 7.

KEY?
Returns current value of DCCH Message Encryption Key.

ESN *n*
[MSS:RDCCH:ESN]
Specifies Electronic Serial Number. Range of *n* is 0 to #hFFFFFFFF.

ESN?
Returns current value of Electronic Serial Number.

SOC *n*
[MSS:RDCCH:SOC]
Specifies SOC. Range of *n* is 0 to 4095.

SOC?
Returns current value of SOC.

CUSTom:

LENGth *n*
[MSS:RDCCH:DCCH:CUSTom:LENGth]
Specifies Custom Control Length. Range of *n* is 1 to 253.

LENGth?
Returns current value of Custom Control Length.

CONTrol *n,x*
[MSS:RDCCH:DCCH:CUSTom:CONTrol]
Specifies Custom Control byte (*x*) selected by *n*. Range of *n* is 0 to 252; range of *x* is 0 to 255.

CONTrol? *n*
Returns current byte value of Custom Control selected by *n*. Range of *n* is 0 to 252.

MSS:

RDCCH:

CONFirmed:

MSGtype *n*
[MSS:RDCCH:CONFirmed:MSGtype]
Specifies Confirmed Message Type. Range of *n* is 0 to 63.

MSGtype?
Returns current value of Confirmed Message Type.

SSDUP:

STATus *n*
[MSS:RDCCH:SSDUP:STATus]
Specifies SSD Update Status. Range of *n* is 0 to 3.

STATus?
Returns current value of SSD Update Status.

AUTHU *n*
[MSS:RDCCH:AUTHU]
Specifies AUTHU. Range of *n* is 0 to #h3FFFF.

AUTHU?
Returns current value of AUTHU.

D. Optional Data Fields

Some of the elements of RACH messages are optional. Optional elements can be enabled or disabled by the following commands.

MSS:

RDCCH:

ENABle:

PSID_RSID:

SELect *n*
[MSS:RDCCH:ENABLe:PSID_RSID:SELect]
Enables (*n* = 1) or disables (*n* = 0) Selected PSID/RSID.

SELect?
Returns current state of Selected PSID/RSID.

DISPlay *n*
[MSS:RDCCH:ENABLe:DISPlay]
Enables (*n* = 1) or disables (*n* = 0) Display.

DISPlay?
Returns current state of Display.

MEASurement:

LTM *n*
[MSS:RDCCH:ENABLe:MEASurement:LTM]
Enables (*n* = 1) or disables (*n* = 0) LTM Measurement.

LTM?
Returns current state of LTM Measurement.

STM *n*
[MSS:RDCCH:ENABLe:MEASurement:STM]
Enables (*n* = 1) or disables (*n* = 0) STM Measurement.

STM?
Returns current state of STM Measurement.

MSS:

RDCCH:

ENABle:

MODE:

VOICe *n*
[MSS:RDCCH:ENABLe:MODE:VOICe]
Enables (*n* = 1) or disables (*n* = 0) Voice Mode.

VOICe?
Returns current state of Voice Mode.

DATA *n*
[MSS:RDCCH:ENABLe:MODE:DATA]
Enables (*n* = 1) or disables (*n* = 0) Data Mode.

DATA?
Returns current state of Data Mode.

MEM *n*
[MSS:RDCCH:ENABLe:MEM]
Enables (*n* = 1) or disables (*n* = 0) Message Encryption Mode.

MEM?
Returns current state of Message Encryption Mode.

BANDWidth *n*
[MSS:RDCCH:ENABLe:BANDWidth]
Enables (*n* = 1) or disables (*n* = 0) Bandwidth.

BANDWidth?
Returns current state of Bandwidth.

CALLING:

PRESentation *n*
[MSS:RDCCH:ENABLe:CALLING:PRESentation]
Enables (*n* = 1) or disables (*n* = 0) Calling Party Number Presentation Indicator.

PRESentation?
Returns current state of Calling Party Number Presentation Indicator.

ADDRess *n*
[MSS:RDCCH:ENABLe:CALLING:ADDRess]
Enables (*n* = 1) or disables (*n* = 0) Calling Party Number.

ADDRess?
Returns current state of Calling Party Number.

MSS:

RDCCH:

ENABle:

CALLING:

SUBaddress *n*
[MSS:RDCCH:ENABLe:CALLING:SUBaddress]
Enables (*n* = 1) or disables (*n* = 0) Calling Party Subaddress.

SUBaddress?
Returns current state of Calling Party Subaddress.

CALLED:

SUBaddress *n*
[MSS:RDCCH:ENABLe:CALLED:SUBaddress]
Enables (*n* = 1) or disables (*n* = 0) Called Party Subaddress.

SUBaddress?
Returns current state of Called Party Subaddress.

SUBaddress *n*
[MSS:RDCCH:ENABLe:SUBaddress]
Enables (*n* = 1) or disables (*n* = 0) Subaddress.

SUBaddress?
Returns current state of Subaddress.

MESSage:

CENTer:

ADDRess *n*
[MSS:RDCCH:ENABLe:MESSage:CENTer:ADDRess]
Enables (*n* = 1) or disables (*n* = 0) Message Center Address.

ADDRess?
Returns current state of Message Center Address.

USER:

GROUP *n*
[MSS:RDCCH:ENABLe:USER:GROUP]
Enables (*n* = 1) or disables (*n* = 0) User Group.

GROUP?
Returns current state of User Group.

MSS:

RDCCH:

ENABle:

USER:

DEST:

ADDRess *n*
[MSS:RDCCH:ENABLe:USER:DEST:ADDRess]
Enables (*n* = 1) or disables (*n* = 0) User Destination Address.

ADDRess?
Returns current state of User Destination Address.

SUBaddress *n*
[MSS:RDCCH:ENABLe:USER:DEST:SUBaddress]
Enables (*n* = 1) or disables (*n* = 0) User Destination Subaddress.

SUBaddress?
Returns current state of User Destination Subaddress.

ORIG:

ADDRess *n*
[MSS:RDCCH:ENABLe:USER:ORIG:ADDRess]
Enables (*n* = 1) or disables (*n* = 0) User Originating Address.

ADDRess?
Returns current state of User Originating Address.

SUBADDRess *n*
[MSS:RDCCH:ENABLe:USER:ORIG:SUBaddress]
Enables (*n* = 1) or disables (*n* = 0) User Originating Subaddress.

SUBADDRess?
Returns current state of User Originating Subaddress.

CNUMber *n*
[MSS:RDCCH:ENABLe:CNUMber]
Enables (*n* = 1) or disables (*n* = 0) C-Number.

CNUMber?
Returns current state of C-Number field.

MSS:

RDCCH:

ENABle:

PFC:

REQuest *n*
[MSS:RDCCH:ENABLe:PFC:REQuest]
Enables (*n* = 1) or disables (*n* = 0) PFC Request.

REQuest?
Returns current state of PFC Request.

DCCH:

MEM *n*
[MSS:RDCCH:ENABLe:DCCH:MEM]
Enables (*n* = 1) or disables (*n* = 0) DCCH MEM.

MEM?
Returns current state of DCCH MEM.

E. Building a RACH message

MSS:

RDCCH:

BUILD
[MSS:RDCCH:BUILD]
Builds the data that makes up a RACH message.

Before executing this command, the message types and data fields that make up the RACH should be programmed. This command then takes the message types and data fields and generates all the data that makes up the data field in each slot of the RACH message.

After executing this command, the data can returned by the MSS:RDCCH:DATA? command defined below. The RACH message generator can then be programmed with the MSS:RDCCH:PROGRAM command defined below.

LENGth?
[MSS:RDCCH:LENGth]
Returns Length of the RDCCH in number of slots.

Use this command after the build command (MSS:RDCCH:BUILD) has been executed.

DATA? *n,m*
[MSS:RDCCH:DATA]
Returns RDCCH data that has been built.

Returns current 16 bit value of selected word (*m*) in selected slot (*n*). Range of n is 0 to 319; range of m is 0 to 6.

The length of each slot is 101 bits (normal) or 79 bits (abbreviated) long. The data in each slot may returned, 16 bits at a time. The data bits in each word are left justified. The 16 most significant bits of the data are returned when *m* = 0. If the message is normal length, the 5 least significant bits of data are returned when *m* = 6. If the message is abbreviated length, then the 15 least significant bits of data are returned when *m* = 4.

This data format correlates with the data format used in the MSS:RDCCH:MESSAGE:DATA command. Therefore, the data in the message can be programmed by transferring the data, one word at a time, to the message generator. This enables the user to modify the data (i.e. scramble data, repeat a frame, change the CI bit) before programming the message generator. If a straight transfer of data is desired, then use the MSS:RDCCH:PROGRAM command. If that command is not used, then set the length of the message with the MSS:RDCCH:LENGTH? command.

MSS:

RDCCH:

PROGram
[MSS:RDCCH:PROGram]
Programs the RACH Message Generator with the data constructed by the **MSS:RDCCH:BUILD** command.

This command automatically determines the length of the RACH message built and overwrites the any length that may have been established by the MSS:RDCCH:MESSAGE:LENGTH command.

The RACH message is not sent with this command, but is sent with the MSS:RDCCH:MESSAGE:SEND command.

7-14-6 BASE STATION PERFORMANCE

The following are miscellaneous commands that are useful for testing the performance of a base station.

- Share Channel Feedback

  The Share Channel Feedback field is used to control access on the RACH. The following commands returns the state of the SCF flags when an access occurs.

  MSS:

  SCF?
  [MSS:SCF]
  Returns current value of the Shared Channel Feedback from the RACH subchannel (22 bit value).

  SCF:

  RN?
  [MSS:SCF:RN]
  Returns current value of Receive/Not Received in the Shared Channel Feedback from the RACH subchannel (5 bit value).

  CPE?
  [MSS:SCF:CPE]
  Returns current value of Coded Partial Echo in the Shared Channel Feedback from the RACH subchannel (11 bit value).

  BRI?
  [MSS:SCF:BRI]
  Returns current value of Busy/Reserved/Idle in the Shared Channel Feedback from the RACH subchannel (6 bit value).

- Coded Superframe Phase

  The Coded Superframe Phase is used to indicate which phase of the superframe is currently being transmitted. It may be useful to know the phase of the superframe associated with the RACH subchannel.

  MSS:

  CSFP?
  [MSS:CSFP]
  Returns current value of Coded Superframe Phase associated with the RACH subchannel (12 bit value).

- Response

  The Base station is required to respond to some of the RACHs. This command provides the time of the response.

  MSS:

  RESPONSE?
  [MSS:RESPONSE]
  Returns the Base station response time (in milliseconds) to a RACH requiring a response.

7-15 FDCCH DATA MONITOR

This section contains the TMAC commands to monitor the FDCCH. The FDCCH Data Monitor consists of the TMAC commands necessary to monitor the Forward Digital Control Channel.

7-15-1 SETUP COMMANDS

The FDCCH Setup commands configure the FM/AM-1600CSA to receive and decode data on the FDCCH.

FDCCH:

SETup
[FDCCH:SETup]
Configures the FM/AM-1600CSA to receive on the FDCCH.

This command also sets the FM/AM-1600S to receive through the Antenna Connector.

CHANnel *n*
[FDCCH:CHANnel]
Selects the Forward channel to monitor. Range of *n* is 1 to 1023.

SLOT *n*
[FDCCH:SLOT]
Selects the full or half rate SLOT on which to receive. Range of *n* is 1 to 3.

SLOT?
Returns current value of Slot.

CHANnel *n*
[FDCCH:CHANnel]
Selects the Forward channel to monitor. Range of *n* is 1 to 1023.

CHANnel?
Returns current value of Channel.

RATE *n*
[FDCCH:RATE]
Selects TDMA transmission rate: Full (*n* = 0) or Half (*n* = 1).

RATE?
Return current setting of Rate.

DVCC *n*
[FDCCH:DVCC]
Specifies Digital Verification Color Code. Range of *n* is 0 to 255.

DVCC?
Returns current value of DVCC.

7-15-2 CONTINUOUS REMOTE RAW TIMESLOT DATA

The following commands Start or Stop the transmission of data received in an IS-136 DCCH timeslot out the RS-232 Connector and specify whether or not a Sync word is included. The data is retransmitted exactly as received without being de-interleaved, corrected for errors or formatted.

Before entering this mode of operation, the baud rate should be set 57600 (for the FM/AM-1600CSA and remote terminal). Each Timeslot consists of 324 bits, which corresponds to 81 ASCII characters when displayed in hexadecimal. The 81 characters representing each Timeslot is separated by a newline character. If less than 80 characters/line is desirable, the sync portion of the data may be omitted (sync portion of data is redundant), reducing the bit count to 296 and the character count to 74. An extra character preceding the data indicates the Timeslot (75 total characters).

FDCCH:

REMote:

TIMEslot:

STARt
[FDCCH:REMote:TIMEslot:STARt]
Starts sending the received Timeslot data out the RS-232 Connector.

STOP
[FDCCH:REMote:TIMEslot:STOP]
Stops sending the received Timeslot data out the RS-232 Connector.

SYNC *n*
[FDCCH:REMote:TIMEslot:SYNC]
Enables (*n* = 1) or disables (*n* = 0) sync word.

Directs that Sync word be transmitted with the data out the RS-232 Connector.

7-15-3 CONTINUOUS REMOTE RAW DATA

This mode differs from Section 7-15-2. This mode de-interleaves the data and performs forward error correction before presenting the data. Only slots that contain a valid CRC are displayed. The Timeslot is broken up into the various data fields and continuously transmitted out the RS-232 Connector. Following the data is a millisecond time stamp which provides relative time between slots of data.

The data is presented in hexadecimal out the RS-232 Connector with each frame of data being separated by a newline character. The data is formatted as follows:

- The first 7 characters represent the 28 bits of the SYNC word.
- The next 3 characters are the hexadecimal value of the 12 bits of the Coded Superframe Phase (CSFP).
- The next 6 characters are the hexadecimal value (left justified) of the 22 bits of the Shared Channel Feedback (SCF).
- The next 32 characters are the hexadecimal value (left justified) of the 125 bits of data.
- The data is followed by a CRC check flag, 1 = good, 0 = bad.
- The flag is followed by a millisecond time stamp.

The following TMAC commands are used to start and stop this operation.

FDCCH:

REMote:

RAW:

STARt
[FDCCH:REMote:RAW:STARt]
Starts sending the received, de-interleaved and decoded data out RS-232 Connector.

STOP
[FDCCH:REMote:RAW:STOP]
Stops sending the received data out RS-232 Connector.

DVCC ***n***
[FDCCH:REMote:RAW:DVCC]
Specifies Digital Verification Color Code. Range of *n* is 1 to 255.

A DVCC, which is used when calculating the CRC, must be specified to decode the message.

An embedded macro named FRAW initiates the START when executed and sends a STOP when any key on the RS-232 terminal is pressed. To use this macro, type in the command FRAW at the RS-232 terminal.

7-15-4 BUFFERED RAW DATA

The FDCCH Raw Data commands consist of the TMAC commands used for Layer 1 raw data buffering. Up to 100 frames of raw data can be captured. (Similar to FOCC raw data TMAC commands.) The following commands are used for this operation.

FDCCH:

RAW:

STARt
[FDCCH:RAW:STARt]
Starts capturing raw data on FDCCH.

STOP
[FDCCH:RAW:STOP]
Stops capturing raw data on FDCCH.

STOP occurs automatically when the buffer is full.

FULL?
[FDCCH:RAW:FULL?]
Returns current state of raw buffer: 1 = full, 0 = not full.

SYNC? *n*
[FDCCH:RAW:SYNC?]
Returns current value of Sync word (22 bit value) in selected raw data frame (*n*). Range of *n* is 0 to 99.

SCF? *n*
[FDCCH:RAW:SCF?]
Returns current value of Shared Channel Feedback (22 bit value) in selected raw data frame. Range of *n* is 0 to 99.

CSFP? *n*
[FDCCH:RAW:CSFP?]
Returns the Coded Super Frame Phase (12 bit value) in the selected raw data frame (*n*). Range of *n* is 0 to 99.

DATA? *n,x*
[FDCCH:RAW:DATA?]
Returns the 16 bit value of the selected raw data byte (*x*) in the selected raw data frame (*n*). Range of *n* is 0 to 99; range of *x* is 0 to 15.

TS? *n*
[FDCCH:RAW:TS?]
Returns the Time Stamp (in milliseconds) (32 bit value) of selected raw data frame (*n*). Range of *n* is 0 to 99.

7-15-5 LAYER 2 DATA MONITOR

The FDCCH Layer 2 Data Monitor consists of the TMAC commands necessary to decode a Layer 2 message. Up to 100 frames can be captured into a buffer for non real-time decoding into Layer 2 data fields.

The process of decoding FDCCH data into Layer 2 consists of two steps:

1. Use the FDCCH Buffered Raw Data commands defined in Section 7-15-4 to capture 100 words into the raw data buffer.
2. Select one frame of raw data from the Raw Buffer and decode the frame into Layer 2 fields.

FDCCH:

LAYER2:

DECode *n*
[FDCCH:LAYER2:DECode]
Decodes the frame of data in selected raw buffer (*n*). Range of *n* is 0 to 99.

TYPE?
[FDCCH:LAYER2:TYPE]
Returns value of the type of data in the frame that had been previously decoded.

VALUE	TYPE
1	F-BCCH
2	E-BCCH
3	SPACH
4	S-BCCH
5	RESERVED

The specific commands used to access the Layer 2 data fields are determined by the TYPE of data.

The following commands are used to access the Layer 2 data fields when TYPE of data is FBCCH.

FDCCH:

LAYER2:

FBCCH:

BC?
[FDCCH:LAYER2:FBCCH:BC]
Returns current state of Begin/Continue (1 bit value). Returns -1 if not available.

BI? ***n***
[FDCCH:LAYER2:FBCCH:BI]
Returns current state of Begin Indicator (1 bit value) selected by *n*. Returns -1 if not available. The range of *n* is 0 to 3.

A Begin Indicator may follow any of the four Layer 3 data fields.

CLI?
[FDCCH:LAYER2:FBCCH:CLI]
Returns current value of Continuation Length Indicator (7 bit value). Returns -1 if not available.

CLI indicates length of first block of Layer 3 data in a continuation frame.

CRC?
[FDCCH:LAYER2:FBCCH:CRC]
Returns current value of Cyclic Redundancy Code (16 bit value). Returns -1 if not available.

EC?
[FDCCH:LAYER2:FBCCH:EC]
Returns current state of E-BCCH Change (1 bit value). Returns -1 if not available.

FC?
[FDCCH:LAYER2:FBCCH:FC]
Returns current state of F-BCCH Change (1 bit value). Returns -1 if not available.

L3DATA? ***n,x***
[FDCCH:LAYER2:FBCCH:L3DATA]
Returns the 8 bit value of the selected byte (*x*) of the selected Layer 3 data message (*n*). Returns -1 if not available. Range of *n* is 0 to 3; range of *x* is 0 to 15.

The number of Layer 3 data messages embedded within a Layer 2 frame can be anywhere from a portion of single data message to 4 full data messages. The maximum number of bytes in a Layer 2 frame is 16 bytes.

The L3LI and CLI may be used to determine the number of 8 bit "words" in a Layer 3 message.

FDCCH:

LAYER2:

FBCCH:

L3LI? *n*

[FDCCH:LAYER2:FBCCH:L3LI]

Returns one of four Layer 3 Length Indicators (8 bit value). Returns -1 if not available. Range of *n* is 0 to 3.

If returned from a continuation frame, the range of *n* is 1 to 3 because the length of the first block of Layer 3 data is specified by CLI.

The following commands are used to access the Layer 2 data fields when TYPE of data is E-BCCH.

FDCCH:

LAYER2:

EBCCH:

BC?
[FDCCH:LAYER2:EBCCH:BC]
Returns current state of Begin/Continue (1 bit value). Returns -1 if not available.

BI? *n*
[FDCCH:LAYER2:EBCCH:BI]
Returns current state of Begin Indicator (1 bit value) selected by *n*. Returns -1 if not available. The range of *n* is 0 to 3.

A Begin Indicator may follow any of the four Layer 3 data fields.

CLI?
[FDCCH:LAYER2:EBCCH:CLI]
Returns current value of Continuation Length Indicator (7 bit value). Returns -1 if not available.

CLI specifies length of first block of Layer 3 data in a continuation frame.

CRC?
[FDCCH:LAYER2:EBCCH:CRC]
Returns current value of Cyclic Redundancy Code (16 bit value). Returns -1 if not available.

ECL?
[FDCCH:LAYER2:EBCCH:ECL]
Returns current value of E-BCCH Cycle Length (8 bit value). Returns -1 if not available.

L3DATA? *n*, *x*
[FDCCH:LAYER2:EBCCH:L3DATA]
Returns the 8 bit value of the selected byte (*x*) of the selected Layer 3 data message (*n*). Returns -1 if not available. Range of *n* is 0 to 3; range of *x* is 0 to 15.

The number of Layer 3 data messages embedded within a Layer 2 frame can be anywhere from a portion of single data message to 4 full data messages. The maximum number of bytes in a Layer 2 frame is 16 bytes. The data returned is left justified.

The L3LI and CLI may be used to determine the number of 8 bit "words" in a Layer 3 message.

FDCCH:

LAYER2:

EBCCH:

L3LI? *n*
[FDCCH:LAYER2:EBCCH:L3LI]
Returns one of four Layer 3 Length Indicators (8 bit value). Returns -1 if not available. Range of *n* is 0 to 3.

If returned from a continuation frame, the range of *n* is 1 to 3 because the length of the first block of Layer 3 data is specified by CLI.

RSVD?
[FDCCH:LAYER2:EBCCH:RSVD]
Returns current state of E-BCCH Layer 2 Reserved (1 bit value). Returns -1 if not available.

The following commands are used to access the Layer 2 data fields when TYPE of data is SPACH:

FDCCH:

LAYER2:

SPACH:

ARM?
[FDCCH:LAYER2:SPACH:ARM]
Returns current state of ARQ Response Mode (1 bit value). Returns -1 if not available.

ARQ_RSVD?
[FDCCH:LAYER2:SPACH:ARQ_RSVD]
Returns current value of ARQ Layer 2 frame RSVD (2 bit value). Returns -1 if not available.

BCN?
[FDCCH:LAYER2:SPACH:BCN]
Returns current state of BCCH Change Notification (1 bit value). Returns -1 if not available.

BT?
[FDCCH:LAYER2:SPACH:BT]
Returns current value of Burst Type (3 bit value). Returns -1 if not available.

BU?
[FDCCH:LAYER2:SPACH:BU]
Returns current value of Burst Usage (3 bit value). Returns -1 if not available.

CRC?
[FDCCH:LAYER2:SPACH:CRC]
Returns current value of Cyclic Redundancy Code (16 bit value). Returns -1 if not available.

EH_RSVD?
[FDCCH:LAYER2:SPACH:EH_RSVD]
Returns current state of Extended Header RSVD (1 bit value). Returns -1 if not available.

FRNO?
[FDCCH:LAYER2:SPACH:FRNO]
Returns current value of Frame Number (5 bit value). Returns -1 if not available.

GA?
[FDCCH:LAYER2:SPACH:GA]
Returns current state of Go Away (1 bit value). Returns -1 if not available.

FDCCH:

LAYER2:

SPACH:

HA_RSVD?
[FDCCH:LAYER2:SPACH:HA_RSVD]
Returns SPACH Header A_RSVD field setting (1 bit value). Returns -1 if not available.

IDT?
[FDCCH:LAYER2:SPACH:IDT]
Returns current value of Identity Type (2 bit value). Returns -1 if not available.

L3DATA? *n,x*
[FDCCH:LAYER2:SPACH:L3DATA]
Returns the 8 bit value of the selected byte (*x*) of the selected Layer 3 Data message (*n*). Returns -1 if not available. Range of *n* is 0 to 3; range of *x* is 0 to 15.

The number of Layer 3 data messages embedded within a Layer 2 frame can be just a portion of a single data message on up to 4 full data messages. The maximum number of bytes in a Layer 2 frame is 16 bytes. The data returned is left justified.

The L3LENGTH command may be used to determine how many 8 bit "words" make up the Layer 3 data field in the frame currently being decoded.

L3LENGTH? *n*
[FDCCH:LAYER2:SPACH:L3LENGTH]
Returns current value of Layer 3 data field length (8 bit value) selected by *n*. Returns -1 if not available. Range of *n* is 0 to 3.

This is the length of the Layer 3 data field in the SPACH Layer 2 frame currently being decoded.

This is necessary because the L3LIs are all contained in the first frame of a multiframe SPACH message and subsequent frames do not have the information specifying the length of the Layer 3 data field.

L3LI? *n*
[FDCCH:LAYER2:SPACH:L3LI]
Returns current value of Layer 3 Length Indicator (8 bit value) selected by *n*. Returns -1 if not available. Range of *n* is 0 to 3.

A SPACH Layer 2 frame may contain up to four Layer 3 Length Indicators.

MEA?
[FDCCH:LAYER2:SPACH:MEA]
Returns current value of Message Encryption Mode (2 bit value). Returns -1 if not available.

FDCCH:

LAYER2:

SPACH:

MEK?
[FDCCH:LAYER2:SPACH:MEK]
Returns current value of Message Encryption Key (2 bit value). Returns -1 if not available.

MM?
[FDCCH:LAYER2:SPACH:MM]
Returns current state of Message Mapping (1 bit value). Returns -1 if not available.

MSID:

LS? *n*
[FDCCH:LAYER2:SPACH:MSID:LS]
Returns the 32 Least Significant bits of Mobile Station Identification selected by *n*. Range of *n* is 0 to 4.

MS? *n*
[FDCCH:LAYER2:SPACH:MSID:MS]
Returns the 18 Most Significant bits of Mobile Station Identification selected by *n*. Range of *n* is 0 to 4.

MSID? *n,x*
[FDCCH:LAYER2:SPACH:MSID]
Returns the 8 bit value of selected byte (*x*) of the selected Mobile Station Identity (*n*). Returns -1 if not available. Range of *n* is 0 to 4; range of *x* is 0 to 2, 4 or 6.

A Layer 2 frame may contain up to 5 mobile station identities. Each MSID is 20, 24, 34 or 50 bits long and is accessed 8 bits at a time with this command. The data returned is left justified.

PCON?
[FDCCH:LAYER2:SPACH:PCON]
Returns current state of PCH Continuation (1 bit value). Returns -1 if not available.

PEA?
[FDCCH:LAYER2:SPACH:PEA]
Returns current value of Partial Echo Assigned (7 bit value). Returns -1 if not available.

PFM?
[FDCCH:LAYER2:SPACH:PFM]
Returns current state of Paging Frame Modifier (1 bit value). Returns -1 if not available.

PI?
[FDCCH:LAYER2:SPACH:PI]
Returns current state of Polling Indicator (1 bit value). Returns -1 if not available.

FDCCH:

LAYER2:

SPACH:

SRM?
[FDCCH:LAYER2:SPACH:SRM]
Returns current state of SPACH Response Mode (1 bit value). Returns -1 if not available.

UGID:

LS?
[FDCCH:LAYER2:SPACH:UGID:LS]
Returns the 32 Least Significant bits of User Group Identification.

MS?
[FDCCH:LAYER2:SPACH:UGID:MS]
Returns the 18 Most Significant bits of User Group Identification.

UGID? ***n,x***
[FDCCH:LAYER2:SPACH:UGID]
Returns the 8 bit value of the selected byte (*x*) of the selected User Group Identity (*n*). Returns -1 if not available. Range of n is 0 to 4; range of x is 0 to 2, 4 or 6.

Each UGID is 20, 24, 34 or 50 bits long and is accessed 8 bits at a time with this command. The data returned is left justified.

7-15-6 L2 FILTERS

The Layer 2 filters enable the user to only buffer Layer 2 frames from a specific logical channel, logical SPACH channel, PCH subchannel, etc. One or more of the filters are selectable by the user.

7-15-7 FDCCH REAL TIME DATA MONITOR

In this mode of operation, the data return is the last data decoded for that data field. Once a data field item is returned, -1 is returned until a new value for that data field is decoded.

A. Layer 1 Data

With the exception of the first two, the following TMAC commands return the data fields defined on the physical layer.

FDCCH:

STARt
[FDCCH:STARt]
Starts decoding the FDCCH.

STOP
[FDCCH:STOP]
Stops decoding the FDCCH.

SYNC?
[FDCCH:SYNC]
Returns the value of the last decoded sync word (28 bit value). Returns -1 if already returned or not available.

CSFP?
[FDCCH:CSFP]
Returns the value of the last decoded Coded Super Frame Phase (12 bit value). Returns -1 if already returned or not available.

SCF?
[FDCCH:SCF]
Returns the value of the last decoded Share Channel Feedback (22 bit value). Returns -1 if already returned or not available.

BRI?
[FDCCH:BRI]
Returns the value of the last decoded Busy/Reserved/Idle (6 bit value). Returns -1 if already returned or not available.

R_N?
[FDCCH:R_N]
Returns the value of the last decoded Received/Not received (5 bit value). Returns -1 if already returned or not available.

FDCCH:

CPE?

[FDCCH:CPE]

Returns the value of the last decoded Coded Partial Echo (11 bit value). Returns -1 if already returned or not available.

TYPE?

[FDCCH:TYPE]

Returns value of the type of data of the last decoded frame. Returns -1 if already returned or not available. This command returns the following:

VALUE	TYPE
1	F-BCCH
2	E-BCCH
3	SPACH
4	S-BCCH
5	RESERVED

CRC?

[FDCCH:CRC]

Returns the value of the last decoded CRC (16 bit value). Returns -1 if already returned or not available.

B. F-BCCH Frames

The next set of commands return data from the F-BCCH frames.

FDCCH:

FBCCH:

BC?
[FDCCH:FBCCH:BC?]
Returns the last decoded value of Begin/Continue (1 bit value). Returns -1 if already returned or not available.

FC?
[FDCCH:FBCCH:FC?]
Returns the last decoded value of F-BCCH Change (1 bit value). Returns -1 if already returned or not available.

EC?
[FDCCH:FBCCH:EC?]
Returns the last decoded value of E-BCCH Change (1 bit value). Returns-1 if already returned or not available.

CLI?
[FDCCH:FBCCH:CLI?]
Returns the last decoded value of Continuation Length Indicator (7 bit value). Returns -1 if already returned or not available.

L3LI?
[FDCCH:FBCCH:L3LI?]
Returns the last decoded value of Layer 3 Length Indicator (8 bit value). Returns -1 if already returned or not available.

BI?
[FDCCH:FBCCH:BI?]
Returns the last decoded value of Begin Indicator (1 bit value). Returns -1 if already returned or not available.

PD?
[FDCCH:FBCCH:PD?]
Returns the last decoded value of Protocol Discriminator (1 bit value). Returns -1 if already returned or not available.

MSGtype?
[FDCCH:FBCCH:MSGtype?]
Returns the last decoded Message Type. Returns -1 if already returned or not available. This command returns the following valid message types:

DCCH STRUCT	ACCESS PARAM	SELECT PARAM	REG PARAM
SYSID	OLC	MACA	NEIGHBOR CELL
RCI	BSMC	EMERG INFO	NEIGHBOR INFO
SERVICE MENU	SOC BSMC ID	SOC	TIME DATE

FDCCH:

FBCCH:

NUMber:

FBCCH?
[FDCCH:FBCCH:NUMber:FBCCH?]
Returns the last decoded value of Number of F-BCCH (3 bit value). Returns -1 if already returned or not available.

This command returns the decoded value plus two in accordance with IS-136.

EBCCH?
[FDCCH:FBCCH:NUMber:EBCCH?]
Returns the last decoded value of Number of E-BCCH (3 bit value). Returns -1 if already returned or not available.

This command returns the decoded value plus one in accordance with IS-136.

SBCCH?
[FDCCH:FBCCH:NUMber:SBCCH?]
Returns the last decoded value of Number of S-BCCH (4 bit value). Returns -1 if already returned or not available.

REServed?
[FDCCH:FBCCH:NUMber:REServed?]
Returns the last decoded value of Number of Reserved Slots (3 bit value). Returns -1 if already returned or not available.

NON_PCH?
[FDCCH:FBCCH:NUMber:NON_PCH?]
Returns the last decoded value of Number of Non-PCH Subchannel Slots (2 bit value). Returns -1 if already returned or not available.

HYPERframe?
[FDCCH:FBCCH:HYPERframe?]
Returns the last decoded value of Hyperframe Counter (4 bit value). Returns -1 if already returned or not available.

EXTended:

PT?
[FDCCH:FBCCH:EXTended:PT?]
Returns the last decoded value of SMS Frame Counter Parameter Type or -1 if already returned or not available.

COUNt?
[FDCCH:FBCCH:EXTended:COUNt?]
Returns the last decoded value of SMS Frame Counter (4 bit value). Returns -1 if already returned or not available (3 bit value).

FDCCH:

FBCCH:

SUPERframe?
[FDCCH:FBCCH:SUPERframe?]
Returns the last decoded value of Primary Superframe indicator (1 bit value). Returns -1 if already returned or not available.

CONfiguration?
[FDCCH:FBCCH:CONfiguration?]
Returns the last decoded value of Slot Configuration (2 bit value). Returns -1 if already returned or not available.

DVCC?
[FDCCH:FBCCH:DVCC?]
Returns the last decoded value of Digital Verification Color Code (8 bit value). Returns -1 if already returned or not available.

PFC?
[FDCCH:FBCCH:PFC?]
Returns the last decoded value of Maximum Supported PFC (Paging Frame Class) (3 bit value). Returns -1 if already returned or not available.

PCH?
[FDCCH:FBCCH:PCH?]
Returns the last decoded value of PCH (Paging Channel) Displacement (3 bit value). Returns -1 if already returned or not available.

PFM?
[FDCCH:FBCCH:PFM?]
Returns the last decoded value of PFM (Paging Frame Modifier) Direction (1 bit value). Returns -1 if already returned or not available.

CBN:

PT?
[FDCCH:FBCCH:CBN:PT?]
Returns the last decoded value of CBN_High Parameter Type (4 bit value). Returns -1 if already returned or not available.

HIGH?
[FDCCH:FBCCH:CBN:HIGH?]
Returns the last decoded value of CBN_High (16 bit value). Returns -1 if already returned or not available.

FDCCH:

FBCCH:

NONPublic:

PROBability:

PT?
[FDCCH:FBCCH:NONPublic:PROBability:PT?]
Returns the last decoded value of Non-Public Probability Block Parameter Type (4 bit value). Returns -1 if already returned or not available.

LENGth?
[FDCCH:FBCCH:NONPublic:PROBability:LENGth?]
Returns the last decoded value of Non-Public Map Length (4 bit value). Returns -1 if already returned or not available.

BLOCk?
[FDCCH:FBCCH:NONPublic:PROBability:BLOCk?]
Returns the last decoded value of Non-Public Block Map (1 to 16 bit value). Returns -1 if already returned or not available.

REGistration:

PT?
[FDCCH:FBCCH:NONPublic:REGistration:PT?]
Returns the last decoded value of Non-Public Registration Control Parameter Type (4 bit value). Returns -1 if already returned or not available.

CONTrol?
[FDCCH:FBCCH:NONPublic:PROBability:CONTrol?]
Returns the last decoded value of Non-Public Registration Control (2 bit value). Returns -1 if already returned or not available.

AUTH?
[FDCCH:FBCCH:AUTH?]
Returns the last decoded value of AUTH (1 bit value). Returns -1 if already returned or not available.

S?
[FDCCH:FBCCH:S?]
Returns the last decoded value of S (1 bit value). Returns -1 if already returned or not available.

RAND?
[FDCCH:FBCCH:RAND?]
Returns the last decoded value of RAND (32 bit value). Returns -1 if already returned or not available.

FDCCH:

FBCCH:

ACCess:

MS_PWR?
[FDCCH:FBCCH:ACCess:MS_PWR?]
Returns the last decoded value of MS_ACC_PWR (Mobile Station Access Power) (4 bit value). Returns -1 if already returned or not available.

RSS_MIN?
[FDCCH:FBCCH:CBN:RSS_MIN?]
Returns the last decoded value of RSS_ACC_MIN (Minimum Access Received Signal Strength) (5 bit value). Returns -1 if already returned or not available.

BURSTsize?
[FDCCH:FBCCH:CBN:BURSTsize?]
Returns the last decoded value of Access Burst Size (1 bit value). Returns -1 if already returned or not available.

MAX:

RETries?
[FDCCH:FBCCH:MAX:RETries?]
Returns the last decoded value of Max Retries (3 bit value). Returns -1 if already returned or not available.

BUSY?
[FDCCH:FBCCH:MAX:BUSY?]
Returns the last decoded value of Max Busy/Reserved (1 bit value). Returns -1 if already returned or not available.

REPetitions?
[FDCCH:FBCCH:MAX:REPetitions?]
Returns the last decoded value of Max Repetitions (2 bit value). Returns -1 if already returned or not available.

STOP?
[FDCCH:FBCCH:MAX:STOP?]
Returns the last decoded value of Max Stop Counter (1 bit value). Returns -1 if already returned or not available.

RDATA:

LENGth?
[FDCCH:FBCCH:RDATA:LENGth?]
Returns the last decoded value of R-DATA Message Length (3 bit value). Returns -1 if already returned or not available.

FDCCH:

FBCCH:

BARred?
[FDCCH:FBCCH:BARred?]
Returns the last decoded value of Cell Barred (5 bit value). Returns -1 if already returned or not available.

SUBaddressing?
[FDCCH:FBCCH:SUBaddressing?]
Returns the last decoded value of Subaddressing Support (1 bit value). Returns -1 if already returned or not available.

DIC?
[FDCCH:FBCCH:DIC?]
Returns the last decoded value of Delay Interval Compensation Mode (1 bit value). Returns -1 if already returned or not available.

SS_SUFF?
[FDCCH:FBCCH:SS_SUFF?]
Returns the last decoded value of SS_SUFF (Signal Strength Sufficient) (5 bit value). Returns -1 if already returned or not available.

SCAN:

INTerval?
[FDCCH:FBCCH:SCAN:INTerval?]
Returns the last decoded value of SCANINTERVAL (4 bit value). Returns -1 if already returned or not available.

OPTion?
[FDCCH:FBCCH:RDATA:OPTion?]
Returns the last decoded value of Scanning Option Indicator (1 bit value). Returns -1 if already returned or not available.

INITial?
[FDCCH:FBCCH:INITial?]
Returns the last decoded value of Initial Selection Control (1 bit value). Returns -1 if already returned or not available.

DELay?
[FDCCH:FBCCH:DELay?]
Returns the last decoded value of DELAY (4 bit value). Returns -1 if already returned or not available.

FDCCH:

FBCCH:

ADDitional:

PT?
[FDCCH:FBCCH:ADDitional:PT?]
Returns the last decoded value of Additional DCCH Information Parameter Type (4 bit value). Returns -1 if already returned or not available.

NUMber?
[FDCCH:FBCCH:ADDitional:NUMber?]
Returns the last decoded value of Number of Additional DCCH Channels (3 bit value). Returns -1 if already returned or not available.

CHANnel? *n*
[FDCCH:FBCCH:ADDitional:CHANnel]
Returns the last decoded value of Additional DCCH Channel Information (11 bit value) selected by *n*. Up to 8 instances can be returned. Range of *n* is 0 to 7 Returns -1 if already returned or not available.

SLOT? *n*
[FDCCH:FBCCH:ADDitional:SLOT]
Returns the last decoded value of Additional Slot Information (2 bit value) selected by *n*. Up to 8 instances can be returned. Range of *n* is 0 to 7. Returns -1 if already returned or not available.

REGH?
[FDCCH:FBCCH:REGH?]
Returns the last decoded value of REGH (1 bit value). Returns -1 if already returned or not available.

REGR?
[FDCCH:FBCCH:REGR?]
Returns the last decoded value of REGR (1 bit value). Returns -1 if already returned or not available.

PUREG?
[FDCCH:FBCCH:PUREG?]
Returns the last decoded value of PUREG (1 bit value). Returns -1 if already returned or not available.

PDREG?
[FDCCH:FBCCH:PDREG?]
Returns the last decoded value of PDREG (1 bit value). Returns -1 if already returned or not available.

FDCCH:

FBCCH:

SYREG?
[FDCCH:FBCCH:SYREG?]
Returns the last decoded value of SYREG (1 bit value). Returns -1 if already returned or not available.

LAREG?
[FDCCH:FBCCH:LAREG?]
Returns the last decoded value of LAREG (1 bit value). Returns -1 if already returned or not available.

DEREG?
[FDCCH:FBCCH:DEREG?]
Returns the last decoded value of DEREG (1 bit value). Returns -1 if already returned or not available.

FOREG?
[FDCCH:FBCCH:FOREG?]
Returns the last decoded value of FOREG (1 bit value). Returns -1 if already returned or not available.

CAPability?
[FDCCH:FBCCH:CAPability?]
Returns the last decoded value of Capability Request (1 bit value). Returns -1 if already returned or not available.

RNUM:

PT?
[FDCCH:FBCCH:RNUM:PT?]
Returns the last decoded value of Present RNUM Parameter Type (4 bit value). Returns -1 if already returned or not available.

NUMber?
[FDCCH:FBCCH:RNUM:NUMber?]
Returns the last decoded value of Present RNUM (10 bit value). Returns -1 if already returned or not available.

REGistration:

PT?
[FDCCH:FBCCH:REGistration:PT?]
Returns the last decoded value of Registration Period Parameter Type (4 bit value). Returns -1 if already returned or not available.

PERiod?
[FDCCH:FBCCH:REGistration:PERiod?]
Returns the last decoded value of REGPER (Registration Period) (9 bit value). Returns -1 if already returned or not available.

FDCCH:

FBCCH:

REGID:

PT?
[FDCCH:FBCCH:REGid:PT?]
Returns the last decoded value of REGID Parameter Type (4 bit value). Returns -1 if already returned or not available.

ID?
[FDCCH:FBCCH:REGistration:ID?]
Returns the last decoded value of REGID (20 bit value). Returns -1 if already returned or not available.

PER?
[FDCCH:FBCCH:REGistration:PER?]
Returns the last decoded value of REGID_PER (Registration ID Period) (4 bit value). Returns -1 if already returned or not available.

SID?
[FDCCH:FBCCH:SID?]
Returns the last decoded value of SID (System Identification) (15 bit value). Returns -1 if already returned or not available.

NETwork?
[FDCCH:FBCCH:NETwork?]
Returns the last decoded value of Network Type (3 bit value). Returns -1 if already returned or not available.

PROTocol?
[FDCCH:FBCCH:PROTocol?]
Returns the last decoded value of Protocol Version (4 bit value). Returns -1 if already returned or not available.

FDCCH:

FBCCH:

PSID_RSID:

PT?
[FDCCH:FBCCH:PSID_RSID:PT?]
Returns the last decoded value of PSID/RSID Set Parameter Type (4 bit value). Returns -1 if already returned or not available.

SOC?
[FDCCH:FBCCH:PSID_RSID:SOC?]
Returns the last decoded value of PSID/RSID SOC (12 bit value). Returns -1 if already returned or not available.

NUMber?
[FDCCH:FBCCH:PSID_RSID:NUMber?]
Returns the last decoded value of Number of PSID/RSID (4 bit value). Returns -1 if already returned or not available.

TYPE? *n*
[FDCCH:FBCCH:PSID_RSID:TYPE?]
Returns the last decoded state of TYPE (1 bit value) of PSID/RSID. Range of *n* is 0 to 1.

VALUE? *n*
[FDCCH:FBCCH:PSID_RSID:VALUE?]
Returns the last decoded value of VALUE (16 bit value) of PSID/RSID. Range of *n* is 0 to 65535.

COUNTRY:

PT?
[FDCCH:FBCCH:COUNTRY:PT?]
Returns the last decoded value of Country Code Parameter Type (4 bit value). Returns -1 if already returned or not available.

CODE?
[FDCCH:FBCCH:COUNTRY:CODE?]
Returns the last decoded value of Country Code (10 bit value). Returns -1 if already returned or not available.

FDCCH:

FBCCH:

ALPHA:

SID:

PT?
[FDCCH:FBCCH:ALPHA:SID:PT?]
Returns the last decoded value of Alphanumeric SID Parameter Type (4 bit value). Returns -1 if already returned or not available.

LENGth?
[FDCCH:FBCCH:ALPHA:SID:LENGth?]
Returns the last decoded value of Length of Alphanumeric System ID (8 bit value). Returns -1 if already returned or not available.

CHARacters?
[FDCCH:FBCCH:ALPHA:SID:CHARacters?]
Returns the selected last decoded value of Alphanumeric SID (ASCII String). Returns -1 if already returned or not available.

BSMC?
[FDCCH:FBCCH:BSMC?]
Returns the last decoded value of BSMC (Base Station Manufacture Code) (8 bit value). Returns -1 if already returned or not available.

CUSTOM:

LENGth?
[FDCCH:FBCCH:CUSTOM:LENGth?]
Returns the last decoded value of the Length of the Custom Control (8 bit value) in octets. Returns or -1 if already returned or not available.

CONTrol? ***n***
[FDCCH:FBCCH:CUSTOM:CONTrol]
Returns the last decoded value of Custom Control (8 bit value) selected by *n*. Up to 256 instances can be returned. Range of *n* is 0 to 255,

Validity is determined by **FDCCH:FBCCH:CUSTOM:LENGth?.**

MACA:

STATus?
[FDCCH:FBCCH:MACA:STATus?]
Returns the last decoded value of MACA_STATUS (2 bit value). Returns -1 if already returned or not available.

TYPE?
[FDCCH:FBCCH:MACA:TYPE?]
Returns the last decoded value of MACA_TYPE (4 bit value). Returns -1 if already returned or not available.

FDCCH:

FBCCH:

MACA:

EIGHT:

PT?
[FDCCH:FBCCH:MACA:EIGHT:PT?]
Returns the last decoded value of MACA_8_CONTROL Parameter Type (4 bit value). Returns -1 if already returned or not available.

CONTrol?
[FDCCH:FBCCH:MACA:EIGHT:CONTrol?]
Returns the last decoded value of MACA_8_CONTROL (1 bit value). Returns -1 if already returned or not available.

LIST:

PT?
[FDCCH:FBCCH:MACA:LIST:PT?]
Returns the last decoded value of MACA_LIST Parameter Type (4 bit value). Returns -1 if already returned or not available.

NUMber?
[FDCCH:FBCCH:MACA:LIST:NUMber?]
Returns the last decoded value of Number of MACA Channels (4 bit value). Returns -1 if already returned or not available.

CHAN? ***n***
[FDCCH:FBCCH:MACA:LIST:CHAN]
Returns the last decoded value of MACA_LIST CHAN (11 bit value) selected by *n*. Up to 16 instances can be returned. Range of *n* is 0 to 15.

Validity is determined by **FDCCH:FBCCH:MACA:LIST:NUMber?**.

FDCCH:

FBCCH:

MACA:

LIST:

OTHER:

PT?
[FDCCH:FBCCH:MACA:LIST:OTHER:PT?]
Returns the last decoded value of MACA_LIST (Other Hyperband) Parameter Type (4 bit value). Returns -1 if already returned or not available.

HYPERband?
[FDCCH:FBCCH:MACA:LIST:OTHER:HYPERband?]
Returns the last decoded value of Hyperband (Other Hyperband) (2 bit value). Returns -1 if already returned or not available.

NUMber?
[FDCCH:FBCCH:MACA:LIST:OTHER:NUMber?]
Returns the last decoded value of Number of MACA Channels (Other Hyperband) (4 bit value). Returns -1 if already returned or not available.

CHAN? *n*
[FDCCH:FBCCH:MACA:LIST:OTHER:CHAN]
Returns the selected last decoded value of MACA_LIST (Other Hyperband) CHAN (11 bit value) selected by *n*. Up to 16 instances can be returned. Range of *n* is 0 to 15.

Validity is determined by **FDCCH:FBCCH:MACA:LIST:OTHER:NUMber?**.

OLC?
[FDCCH:FBCCH:OLC?]
Returns the last decoded value of Overload Control (16 bit value). Returns -1 if already returned or not available.

MAP:

VPM?
[FDCCH:FBCCH:MAP:VPM?]
Returns the last decoded value of Voice Privacy Mode Map (4 bit value). Returns -1 if already returned or not available.

DPM?
[FDCCH:FBCCH:MAP:DPM?]
Returns the last decoded value of Data Privacy Mode Map (4 bit value). Returns -1 if already returned or not available.

FDCCH:

FBCCH:

MAP:

CODER?
[FDCCH:FBCCH:MAP:CODER?]
Returns the last decoded value of Voice Coder Map (6 bit value). Returns -1 if already returned or not available.

MEA:

DOMAIN?
[FDCCH:FBCCH:MAP:MEA:DOMAIN?]
Returns the last decoded value of Message Encryption Algorithm Domain Map (8 bit value). Returns -1 if already returned or not available.

ALGORithms? *n*
[FDCCH:FBCCH:MAP:MEA:ALGORithms]
Returns the selected last decoded value of Message Encryption Algorithm (4 bit value) selected by *n*. Up to 8 instances can be returned. Range of *n* is 0 to 7.

Validity is determined by **FDCCH:FBCCH:MAP:MEA:DOMAIN?**.

MEK?
[FDCCH:FBCCH:MAP:MEK?]
Returns the last decoded value of Message Encryption Key Map (4 bit value). Returns -1 if already returned or not available.

MENU?
[FDCCH:FBCCH:MAP:MENU?]
Returns the last decoded value of Menu Map (10 bit value). Returns -1 if already returned or not available.

ARQ?
[FDCCH:FBCCH:MAP:ARQ?]
Returns the last decoded value of FACCH/SACCH ARQ Map (1 bit value). Returns -1 if already returned or not available.

USER?
[FDCCH:FBCCH:MAP:USER?]
Returns the last decoded value of User Group Map (1 bit value). Returns -1 if already returned or not available.

SMS?
[FDCCH:FBCCH:MAP:SMS?]
Returns the last decoded value of SMS Map (2 bit value). Returns -1 if already returned or not available.

FDCCH:

FBCCH:

IRA? *n*
[FDCCH:FBCCH:IRA?]
Returns the last decoded state of IRA Support (1 bit value) selected by *n*. Range of *n* is 0 to 1.

SOC?
[FDCCH:FBCCH:SOC?]
Returns the last decoded value of SOC (System Operator Code) (12 bit value). Returns -1 if already returned or not available.

C. E-BCCH Frames

The following set of commands return the data fields in E-BCCH Messages.

FDCCH:

EBCCH:

BC?
[FDCCH:EBCCH:BC?]
Returns the last decoded value of Begin/Continue (1 bit value). Returns -1 if already returned or not available.

CLI?
[FDCCH:EBCCH:CLI?]
Returns the last decoded value of Continuation Length Indicator (7 bit value). Returns -1 if already returned or not available.

L3LI?
[FDCCH:EBCCH:L3LI?]
Returns the last decoded value of Layer 3 Length Indicator (8 bit value). Returns -1 if already returned or not available.

BI?
[FDCCH:EBCCH:BI?]
Returns the last decoded value of Begin Indicator (1 bit value). Returns -1 if already returned or not available.

ECL?
[FDCCH:EBCCH:ECL?]
Returns the last decoded value of E-BCCH Cycle Length (8 bit value). Returns -1 if already returned or not available.

PD?
[FDCCH:EBCCH:PD?]
Returns the last decoded value of Protocol Discriminator (2 bit value). Returns -1 if already returned or not available.

MSGtype?
[FDCCH:EBCCH:MSGtype?]
Returns the last decoded value of Message Type. Returns -1 if already returned or not available. This command returns the following valid message types:

DCCH STRUCT	ACCESS PARAM	SELECT PARAM	REG PARAM
SYSID	OLC	MACA	NEIGHBOR CELL
RCI	BSMC	EMERG INFO	NEIGHBOR INFO
SERVICE MENU	SOC BSMC ID	SOC	TIME DATE
ALT RCI			

FDCCH:

EBCCH:

SERV_SS?
[FDCCH:EBCCH:SERV_SS?]
Returns the last decoded value of SERV_SS (4 bit value). Returns -1 if already returned or not available.

NONPublic:

PT?
[FDCCH:EBCCH:NONPublic:PT?]
Returns the last decoded value of Non-Public Probability Blocks Parameter Type (4 bit value). Returns -1 if already returned or not available.

LENGth?
[FDCCH:EBCCH:NONPublic:LENGth?]
Returns the last decoded value of Non-Public Map Length (4 bit value). Returns -1 if already returned or not available.

BLOCk?
[FDCCH:EBCCH:NONPublic:BLOCk?]
Returns the last decoded value of Non-Public Block Map (16 bit value). Returns -1 if already returned or not available.

NEIGHbor:

TDMA:

PT?
[FDCCH:EBCCH:NEIGHbor:TDMA:PT?]
Returns the last decoded value of Neighbor Cell List (TDMA) Parameter Type (4 bit value). Returns -1 if already returned or not available.

NUMber?
[FDCCH:EBCCH:NEIGHbor:TDMA:NUMber?]
Returns the last decoded value of Number of TDMA Neighbor Cells (5 bit value). Returns -1 if already returned or not available.

CELL:

CHAN? *n*
[FDCCH:EBCCH:NEIGHbor:TDMA:CELL:CHAN?]
Returns the last decoded value of TDMA Neighbor Cell CHAN (11 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

PROTocol? *n*
[FDCCH:EBCCH:NEIGHbor:TDMA:CELL:PROTocol?]
Returns the last decoded value of TDMA Neighbor Cell Protocol Version (4 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

FDCCH:

EBCCH:

NEIGHbor:

TDMA:

CELL:

DVCC? *n*
[FDCCH:EBCCH:NEIGHbor:TDMA:CELL:DVCC?]
Returns the last decoded value of TDMA Neighbor Cell DVCC (8 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

OFFset? *n*
[FDCCH:EBCCH:NEIGHbor:TDMA:CELL:OFFset?]
Returns the last decoded value of TDMA Neighbor Cell RESEL_OFFSET (7 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

SS_SUFF? *n*
[FDCCH:EBCCH:NEIGHbor:TDMA:CELL:SS_SUFF?]
Returns the last decoded value of TDMA Neighbor Cell SS_SUFF (5 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

DELay? *n*
[FDCCH:EBCCH:NEIGHbor:TDMA:CELL:DELay?]
Returns the last decoded value of TDMA Neighbor Cell DELAY (4 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

HL_FREQ? *n*
[FDCCH:EBCCH:NEIGHbor:TDMA:CELL:HL_FREQ?]
Returns the last decoded value of TDMA Neighbor Cell HL_FREQ (1 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

SYNC? *n*
[FDCCH:EBCCH:NEIGHbor:TDMA:CELL:SYNC?]
Returns the last decoded value of TDMA Neighbor Cell SYNC (1 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

FDCCH:

EBCCH:

NEIGHbor:

TDMA:

CELL:

TYPE:

CELL? *n*
[FDCCH:EBCCH:NEIGHbor:TDMA:CELL:TYPE:CELL?]
Returns the last decoded value of TDMA Neighbor Cell CELLTYPE (2 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

NETwork? *n*
[FDCCH:EBCCH:NEIGHbor:TDMA:CELL:TYPE:NETwork?]
Returns the last decoded value of TDMA Neighbor Cell Network Type (3 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

RETRY? *n*
[FDCCH:EBCCH:NEIGHbor:TDMA:CELL:RETRY?]
Returns the last decoded value of TDMA Neighbor Cell Directed Retry Channel (1 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

ACCess:

MS_PWR? *n*
[FDCCH:EBCCH:NEIGHbor:TDMA:CELL:ACCess:MS_PWR?]
Returns the last decoded value of TDMA Neighbor Cell MS_ACC_PWR (4 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

RSS_MIN? *n*
[FDCCH:EBCCH:NEIGHbor:TDMA:CELL:ACCess:RSS_MIN?]
Returns the last decoded value of TDMA Neighbor Cell RSS_ACC_MIN (5 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

FDCCH:

EBCCH:

NEIGHbor:

TDMA:

CELL:

PSID_RSID:

INDicator? *n*
[FDCCH:EBCCH:NEIGHbor:TDMA:CELL:PSID_RSID:INDicator?]
Returns the last decoded value of TDMA Neighbor Cell PSID/RSID Indicator (1 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

LENGth? *n*
[FDCCH:EBCCH:NEIGHbor:TDMA:CELL:PSID_RSID:LENGth?]
Returns the last decoded value of TDMA Neighbor Cell PSID/RSID Support Length (4 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

SUPport? *n*
[FDCCH:EBCCH:NEIGHbor:TDMA:CELL:PSID_RSID:SUPport?]
Returns the last decoded value of TDMA Neighbor Cell PSID/RSID Support (16 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

INFO:

PT?
[FDCCH:EBCCH:NEIGHbor:TDMA:INFO:PT?]
Returns the last decoded value of TDMA Service Info Parameter Type (4 bit value). Returns -1 if already returned or not available.

COUNt?
[FDCCH:EBCCH:NEIGHbor:TDMA:INFO:COUNt?]
Returns the last decoded value of TDMA Neighbor Count (5 bit value). Returns -1 if already returned or not available.

FDCCH:

EBCCH:

NEIGHbor:

TDMA:

INFO:

SERVice:

INDicator? *n*
[FDCCH:EBCCH:NEIGHbor:TDMA:INFO:SERVice:INDicator?]
Returns the last decoded value of TDMA Service Map Indicator (1 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

MAP? *n*
[FDCCH:EBCCH:NEIGHbor:TDMA:INFO:SERVice:MAP?]
Returns the last decoded value of TDMA Service Map (10 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

ANAlog:

PT?
[FDCCH:EBCCH:NEIGHbor:ANAlog:PT?]
Returns the last decoded value of Neighbor Cell List (Analog) Parameter Type (4 bit value). Returns -1 if already returned or not available.

NUMber?
[FDCCH:EBCCH:NEIGHbor:ANAlog:NUMber?]
Returns the last decoded value of Number of Analog Neighbor Cells (5 bit value). Returns -1 if already returned or not available.

CELL:

CHAN? *n*
[FDCCH:EBCCH:NEIGHbor:ANAlog:CELL:CHAN?]
Returns the last decoded value of Neighbor Cell List (Analog) CHAN (11 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

PROTocol? *n*
[FDCCH:EBCCH:NEIGHbor:ANAlog:CELL:PROTocol?]
Returns the last decoded value of Neighbor Cell List (Analog) Protocol Version (4 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

FDCCH:

EBCCH:

NEIGHbor:

ANAlog:

CELL:

DCC? *n*
[FDCCH:EBCCH:NEIGHbor:ANAlog:CELL:DCC?]
Returns the last decoded value of Neighbor Cell List (Analog) DCC (2 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

OFFset? *n*
[FDCCH:EBCCH:NEIGHbor:ANAlog:CELL:OFFset?]
Returns the last decoded value of Neighbor Cell List (Analog) RESEL_OFFSET (7 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

SS_SUFF? *n*
[FDCCH:EBCCH:NEIGHbor:ANAlog:CELL:SS_SUFF?]
Returns the last decoded value of Neighbor Cell List (Analog) SS_SUFF (5 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

DELay? *n*
[FDCCH:EBCCH:NEIGHbor:ANAlog:CELL:DELay?]
Returns the last decoded value of Neighbor Cell List (Analog) DELAY (4 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

HL_FREQ? *n*
[FDCCH:EBCCH:NEIGHbor:ANAlog:CELL:HL_FREQ?]
Returns the last decoded value of Neighbor Cell List (Analog) HL_FREQ (1 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

TYPE:

CELL? *n*
[FDCCH:EBCCH:NEIGHbor:ANAlog:CELL:TYPE:CELL?]
Returns the last decoded value of Neighbor Cell List (Analog) CELLTYPE (2 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

FDCCH:

EBCCH:

NEIGHbor:

ANAlog:

CELL:

TYPE:

NETwork? *n*
[FDCCH:EBCCH:NEIGHbor:ANAlog:CELL:TYPE:NETwork?]
Returns the last decoded value of Neighbor Cell List (Analog) Network Type (3 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

RETRY? *n*
[FDCCH:EBCCH:NEIGHbor:ANAlog:CELL:RETRY?]
Returns the last decoded value of Neighbor Cell List (Analog) Directed Retry Channel (1 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

ACCess:

MS_PWR? *n*
[FDCCH:EBCCH:NEIGHbor:ANAlog:CELL:ACCess:MS_PWR?]
Returns the last decoded value of Neighbor Cell List (Analog) MS_ACC_PWR (4 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

RSS_MIN? *n*
[FDCCH:EBCCH:NEIGHbor:ANAlog:CELL:ACCess:RSS_MIN?]
Returns the last decoded value of Neighbor Cell List (Analog) RS_ACC_MIN (5 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

OTHER:

PT?
[FDCCH:EBCCH:NEIGHbor:OTHER:PT?]
Returns the last decoded value of Neighbor Cell List (Other Hyperband) Parameter Type (4 bit value). Returns -1 if already returned or not available.

NUMber?
[FDCCH:EBCCH:NEIGHbor:OTHER:NUMber?]
Returns the last decoded value of Number of Neighbor Cells (Other Hyperband) (5 bit value). Returns -1 if already returned or not available.

FDCCH:

EBCCH:

NEIGHbor:

OTHER:

CELL:

HYPERband?
[FDCCH:EBCCH:NEIGHbor:OTHER:CELL:HYPERband?]
Returns the last decoded value of Neighbor Cell List (Other Hyperband) (2 bit value). Returns -1 if already returned or not available.

CHAN? *n*
[FDCCH:EBCCH:NEIGHbor:OTHER:CELL:CHAN?]
Returns the last decoded value of Neighbor Cell List (Other Hyperband) CHAN (11 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

PROTocol? *n*
[FDCCH:EBCCH:NEIGHbor:OTHER:CELL:PROTocol?]
Returns the last decoded value of Neighbor Cell List (Other Hyperband) Protocol Version (4 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

DVCC? *n*
[FDCCH:EBCCH:NEIGHbor:OTHER:CELL:DVCC?]
Returns the last decoded value of Neighbor Cell List (Other Hyperband) DVCC (8 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

OFFset? *n*
[FDCCH:EBCCH:NEIGHbor:OTHER:CELL:OFFset?]
Returns the last decoded value of Neighbor Cell List (Other Hyperband) RESEL_OFFSET (7 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

SS_SUFF? *n*
[FDCCH:EBCCH:NEIGHbor:OTHER:CELL:SS_SUFF?]
Returns the last decoded value of Neighbor Cell List (Other Hyperband) SS_SUFF (5 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

DELay? *n*
[FDCCH:EBCCH:NEIGHbor:OTHER:CELL:DELay?]
Returns the last decoded value of Neighbor Cell List (Other Hyperband) DELAY (4 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

FDCCH:

EBCCH:

NEIGHbor:

OTHER:

CELL:

HL_FREQ? *n*
[FDCCH:EBCCH:NEIGHbor:OTHER:CELL:HL_FREQ?]
Returns the last decoded value of Neighbor Cell List (Other Hyperband) HL_FREQ (1 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

SYNC? *n*
[FDCCH:EBCCH:NEIGHbor:OTHER:CELL:SYNC?]
Returns the last decoded value of Neighbor Cell List (Other Hyperband) SYNC (1 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

TYPE:

CELL? *n*
[FDCCH:EBCCH:NEIGHbor:OTHER:CELL:TYPE:CELL?]
Returns the last decoded value of Neighbor Cell List (Other Hyperband) CELLTYPE (2 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

NETwork? *n*
[FDCCH:EBCCH:NEIGHbor:OTHER:CELL:TYPE:NETwork?]
Returns the last decoded value of Neighbor Cell List (Other Hyperband) Network Type (3 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

RETRY? *n*
[FDCCH:EBCCH:NEIGHbor:OTHER:CELL:RETRY?]
Returns the last decoded value of Neighbor Cell List (Other Hyperband) Directed Retry Channel (1 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

FDCCH:

EBCCH:

NEIGHbor:

OTHER:

CELL:

ACCess:

MS_PWR? *n*
[FDCCH:EBCCH:NEIGHbor:OTHER:CELL:ACCess:MS_PWR?]
Returns the last decoded value of Neighbor Cell List (Other Hyperband) MS_ACC_PWR (4 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

RSS_MIN? *n*
[FDCCH:EBCCH:NEIGHbor:OTHER:CELL:ACCess:RSS_MIN?]
Returns the last decoded value of Neighbor Cell List (Other Hyperband) RSS_ACC_MIN (5 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

PSID_RSID:

INDicator? *n*
[FDCCH:EBCCH:NEIGHbor:OTHER:CELL:PSID_RSID:INDicator?]
Returns the last decoded value of Neighbor Cell List (Other Hyperband) PSID/RSID Indicator (1 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

LENGth? *n*
[FDCCH:EBCCH:NEIGHbor:OTHER:CELL:PSID_RSID:LENGth?]
Returns the last decoded value of Neighbor Cell List (Other Hyperband) PSID/RSID Support Length (4 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

SUPport?
[FDCCH:EBCCH:NEIGHbor:OTHER:CELL:PSID_RSID:SUPport?]
Returns the last decoded value of Neighbor Cell List (Other Hyperband) PSID/RSID Support (16 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

Validity is determined by **FDCCH:EBCCH:NEIGHbor:OTHER:CELL:PSID_RSID:LENGth?**.

FDCCH:

EBCCH:

NEIGHbor:

OTHER:

INFO:

PT?
[FDCCH:EBCCH:NEIGHbor:OTHER:INFO:PT?]
Returns the last decoded value of TDMA Service Info (Other Hyperband) Parameter Type (4 bit value). Returns -1 if already returned or not available.

HYPERband?
[FDCCH:EBCCH:NEIGHbor:OTHER:INFO:HYPERband?]
Returns the last decoded value of TDMA Service Info (Other Hyperband) (2 bit value). Returns -1 if already returned or not available.

COUNt?
[FDCCH:EBCCH:NEIGHbor:OTHER:INFO:COUNt?]
Returns the last decoded value of TDMA Service Info (Other Hyperband) Neighbor Count (5 bit value). Returns -1 if already returned or not available.

SERVice:

INDicator? ***n***
[FDCCH:EBCCH:NEIGHbor:OTHER:INFO:SERVice:INDicator?]
Returns the last decoded value of TDMA Service Info (Other Hyperband) Service Map Indicator (1 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

MAP? ***n***
[FDCCH:EBCCH:NEIGHbor:OTHER:INFO:SERVice:MAP?]
Returns the last decoded value of TDMA Service Info (Other Hyperband) Service Map (10 bit value) selected by *n*. Up to 32 instances can be returned. Range of *n* is 0 to 31. Returns -1 if already returned or not available.

RCI?
[FDCCH:EBCCH:RCI?]
Returns the last decoded value of RCI (2 bit value). Returns -1 if already returned or not available.

FDCCH:

EBCCH:

CHANnel:

PT?
[FDCCH:EBCCH:CHANnel:PT?]
Returns the last decoded value of RF Channel Allocation Parameter Type (4 bit value). Returns -1 if already returned or not available.

NUMber?
[FDCCH:EBCCH:CHANnel:NUMber?]
Returns the last decoded value of RF Channel Allocation Number of Channel Groups (6 bit value). Returns -1 if already returned or not available.

GROUP:

FIRST? *n*
[FDCCH:EBCCH:CHANnel:GROUP:FIRST?]
Returns the last decoded value of RF Channel Allocation Channel Group First Channel (11 bit value) selected by *n*. Up to 64 instances can be returned. Range of *n* is 0 to 63. Returns -1 if already returned or not available.

LAST? *n*
[FDCCH:EBCCH:CHANnel:GROUP:LAST?]
Returns the last decoded value of RF Channel Allocation Channel Group Last Channel (11 bit value) selected by *n*. Up to 64 instances can be returned. Range of *n* is 0 to 63. Returns -1 if already returned or not available.

BSMC?
[FDCCH:EBCCH:BSMC?]
Returns the last decoded value of BSMC (Base Station Manufacture Code) (8 bit value). Returns -1 if already returned or not available.

CUSTOM:

LENGth?
[FDCCH:EBCCH:CUSTOM:LENGth?]
Returns the last decoded value of Length of Custom Control in octets (8 bit value). Returns -1 if already returned or not available.

CONTrol? *n*
[FDCCH:EBCCH:CUSTOM:CONTrol?]
Returns the last decoded value of Custom Control (8 bit value) selected by *n*. Up to 256 instances can be returned. Range of *n* is 0 to 255. Returns -1 if already returned or not available.

Validity is determined by **FDCCH:EBCCH:CUSTOM:LENGth?**.

FDCCH:

EBCCH:

TEXT:

LENGth?
[FDCCH:EBCCH:TEXT:LENGth?]
Returns the last decoded value of Length of Text Message Data Unit in octets (8 bit value). Returns -1 if already returned or not available.

ENCoding?
[FDCCH:EBCCH:TEXT:ENCoding?]
Returns the last decoded value of Text Message Data Unit Encoding Identifier (5 bit value). Returns -1 if already returned or not available.

REServed?
[FDCCH:EBCCH:TEXT:REServed?]
Returns the last decoded value of Text Message Data Unit Reserved (3 bit value). Returns -1 if already returned or not available.

CHARacter? ***n***
[FDCCH:EBCCH:TEXT:CHARacter?]
Returns the last decoded value of Text Message Data Unit Short Message Character (8 bit value) selected by *n*. Up to 245 instances can be returned. Range of *n* is 0 to 244. Returns -1 if already returned or not available.

Validity is determined by FDCCH:EBCCH:TEXT:LENGth?.

SIGnal:

PT?
[FDCCH:EBCCH:SIGnal:PT?]
Returns the last decoded value of Signal Parameter Type (4 bit value). Returns -1 if already returned or not available.

PITCH?
[FDCCH:EBCCH:SIGnal:PITCH?]
Returns the last decoded value of Signal Pitch (2 bit value). Returns -1 if already returned or not available.

CADence?
[FDCCH:EBCCH:SIGnal:CADence?]
Returns the last decoded value of Signal Cadence (6 bit value). Returns -1 if already returned or not available.

DURation?
[FDCCH:EBCCH:SIGnal:DURation?]
Returns the last decoded value of Signal Duration (4 bit value). Returns -1 if already returned or not available.

FDCCH:

EBCCH:

MACA:

STATus?
[FDCCH:EBCCH:MACA:STATus?]
Returns the last decoded value of MACA_STATUS (2 bit value). Returns -1 if already returned or not available.

TYPE?
[FDCCH:EBCCH:MACA:TYPE?]
Returns the last decoded value of MACA_TYPE (4 bit value). Returns -1 if already returned or not available.

EIGHT:

PT?
[FDCCH:EBCCH:MACA:EIGHT:PT?]
Returns the last decoded value of MACA_8_CONTROL Parameter Type (4 bit value). Returns -1 if already returned or not available.

CONTrol?
[FDCCH:EBCCH:MACA:EIGHT:CONTrol?]
Returns the last decoded value of MACA_8_CONTROL (1 bit value). Returns -1 if already returned or not available.

LIST:

PT?
[FDCCH:EBCCH:MACA:LIST:PT?]
Returns the last decoded value of MACA_LIST Parameter Type (4 bit value). Returns -1 if already returned or not available.

NUMber?
[FDCCH:EBCCH:MACA:LIST:NUMber?]
Returns the last decoded value of Number of MACA Channels (4 bit value). Returns -1 if already returned or not available.

CHAN? *n*
[FDCCH:EBCCH:MACA:LIST:CHAN?]
Returns the last decoded value of MACA_LIST CHAN (11 bit value) selected by *n*. Up to 16 instances can be returned. Range of *n* is 0 to 15. Returns -1 if already returned or not available.

Validity is determined by **FDCCH:EBCCH:MACA:LIST:NUMber?**.

FDCCH:

EBCCH:

MACA:

OTHER:

PT?
[FDCCH:EBCCH:MACA:OTHER:PT?]
Returns the last decoded value of MACA_LIST (Other Hyperband) Parameter Type (4 bit value). Returns -1 if already returned or not available.

HYPERband?
[FDCCH:EBCCH:MACA:OTHER:HYPERband?]
Returns the last decoded value of Hyperband (2 bit value). Returns -1 if already returned or not available.

NUMber?
[FDCCH:EBCCH:MACA:OTHER:NUMber?]
Returns the last decoded value of Number of MACA Channels (4 bit value). Returns -1 if already returned or not available.

CHAN? *n*
[FDCCH:EBCCH:MACA:OTHER:CHAN?]
Returns the last decoded value of MACA_LIST (Other Hyperband) CHAN (11 bit value) selected by *n*. Up to 16 instances can be returned. Range of *n* is 0 to 15. Returns -1 if already returned or not available.

Validity is determined by **FDCCH:EBCCH:MACA: OTHER:NUMber?**.

MAP:

VPM?
[FDCCH:EBCCH:MAP:VPM?]
Returns the last decoded value of Voice Privacy Mode Map (4 bit value). Returns -1 if already returned or not available.

DPM?
[FDCCH:EBCCH:MAP:DPM?]
Returns the last decoded value of Data Privacy Mode Map (4 bit value). Returns -1 if already returned or not available.

CODER?
[FDCCH:EBCCH:MAP:CODER?]
Returns the last decoded value of Voice Coder Map (6 bit value). Returns -1 if already returned or not available.

FDCCH:

EBCCH:

MAP:

MEA:

DOMAIN?
[FDCCH:EBCCH:MAP:MEA:DOMAIN?]
Returns the last decoded value of Message Encryption Algorithm Domain Map (8 bit value). Returns -1 if already returned or not available.

ALGORithms? *n*
[FDCCH:EBCCH:MAP:MEA:ALGORithms?]
Returns the last decoded value of Message Encryption Algorithm (4 bit value) selected by *n*. Up to 8 instances can be returned. Range of *n* is 0 to 7. Returns -1 if already returned or not available.

Validity is determined by **FDCCH:EBCCH:MAP:MEA:DOMAIN?**.

MEK?
[FDCCH:EBCCH:MAP:MEK?]
Returns the last decoded value of Message Encryption Key Map (4 bit value). Returns -1 if already returned or not available.

MENU?
[FDCCH:EBCCH:MAP:MENU?]
Returns the last decoded value of Menu Map (10 bit value). Returns -1 if already returned or not available.

ARQ?
[FDCCH:EBCCH:MAP:ARQ?]
Returns the last decoded value of FACCH/SACCH ARQ Map (1 bit value). Returns -1 if already returned or not available.

USER?
[FDCCH:EBCCH:MAP:USER?]
Returns the last decoded value of User Group Map (1 bit value). Returns -1 if already returned or not available.

SMS?
[FDCCH:EBCCH:MAP:SMS?]
Returns the last decoded value of SMS Map (2 bit value). Returns -1 if already returned or not available.

SOC?
[FDCCH:EBCCH:SOC?]
Returns the last decoded value of SOC (System Operator Code) (12 bit value). Returns -1 if already returned or not available.

FDCCH:

EBCCH:

TIME?
[FDCCH:EBCCH:TIME?]
Returns the last decoded value of Time from Jan 1, 1980 (32 bit value). Returns -1 if already returned or not available.

SID?
[FDCCH:EBCCH:SID?]
Returns the last decoded value of System Identification (14 bit value). Returns -1 if already returned or not available.

CHAN?
[FDCCH:EBCCH:CHAN?]
Returns the last decoded value of CHAN (10 bit value). Returns -1 if already returned or not available.

MCC:

CODE?
[FDCCH:EBCCH:MCC:CODE?]
Returns the last decoded value of Mobile Country Code (9 bit value). Returns -1 if already returned or not available.

PT?
[FDCCH:EBCCH:MCC:PT?]
Returns the last decoded value of Mobile Country Code Parameter Type (4 bit value). Returns -1 if already returned or not available.

HYPERband:

INFO?
[FDCCH:EBCCH:HYPERband:INFO?]

Returns the last decoded value of Hyperband Info (2 bit value). Returns -1 if already returned or not available.

PT?
[FDCCH:EBCCH:HYPERband:PT?]
Returns the last decoded value of Hyperband Info Parameter Type (4 bit value). Returns -1 if already returned or not available.

FDCCH:

EBCCH:

ZONE:

DIRection?
[FDCCH:EBCCH:ZONE:DIRection?]
Returns the last decoded value of Time Zone Offset Direction (1 bit value). Returns -1 if already returned or not available.

MINutes?
[FDCCH:EBCCH:ZONE:MINutes?]
Returns the last decoded value of Time Zone Offset Minutes (10 bit value). Returns -1 if already returned or not available.

DST?
[FDCCH:EBCCH:ZONE:DST?]
Returns the last decoded value of Time Zone Offset Daylight Savings Indicator (1 bit value). Returns -1 if already returned or not available.

D. SPACH Frames

The next set of commands return data from SPACH frames.

FDCCH:

SPACH:

BU?
[FDCCH:SPACH:BU?]
Returns the last decoded value of Burst Usage (3 bit value). Returns -1 if already returned or not available.

PCON?
[FDCCH:SPACH:PCON?]
Returns the last decoded value of PCH Continuation (1 bit value). Returns -1 if already returned or not available.

BCN?
[FDCCH:SPACH:BCN?]
Returns the last decoded value of BCCH Change Notification (1 bit value). Returns -1 if already returned or not available.

PFM?
[FDCCH:SPACH:PFM?]
Returns the last decoded value of Paging Frame Modifier (1 bit value). Returns -1 if already returned or not available.

BT?
[FDCCH:SPACH:BT?]
Returns the last decoded value of Burst Type (3 bit value). Returns -1 if already returned or not available.

IDT?
[FDCCH:SPACH:IDT?]
Returns the last decoded value of Identity Type (2 bit value). Returns -1 if already returned or not available.

FDCCH:

SPACH:

MSID:

MS? *n*
[FDCCH:SPACH:MSID:MS?]
Returns the 18 Most Significant bits of MSID (Mobile Station Identification) selected by *n*. Range of *n* is 0 to 4.

LS? *n*
[FDCCH:SPACH:MSID:LS?]
Returns the 32 Least Significant bits of MSID (Mobile Station Identification) selected by *n*. Range of *n* is 0 to 4.

MIN? *n*
[FDCCH:SPACH:MSID:MIN?]
Returns the last decoded value of MIN (ASCII String) selected by *n*. Up to four instances can be returned. Range of *n* is 0 to 3. Returns -1 if already returned or not available.

If IDT indicates a 34 bit MSID then the value is also stored as a Mobile Identification Number.

MM?
[FDCCH:SPACH:MM?]
Returns the last decoded value of Message Mapping (1 bit value). Returns -1 if already returned or not available.

PEA?
[FDCCH:SPACH:PEA?]
Returns the last decoded value of Partial Echo Assigned (7 bit value). Returns -1 if already returned or not available.

PI?
[FDCCH:SPACH:PI?]
Returns the last decoded value of Polling Indicator (1 bit value). Returns -1 if already returned or not available.

SRM?
[FDCCH:SPACH:SRM?]
Returns the last decoded value of SPACH Response Mode (1 bit value). Returns -1 if already returned or not available.

FDCCH:

SPACH:

UGID:

MS? *n*
[FDCCH:SPACH:UGID:MS?]
Returns the 18 Most Significant bits of UGID (Mobile Station Identification) selected by *n*. Range of *n* is 0 to 4.

LS? *n*
[FDCCH:SPACH:UGID:LS?]
Returns 32 Least Significant bits of UGID (Mobile Station Identification) selected by *n*. Range of *n* is 0 to 4.

MIN? *n*
[FDCCH:SPACH:UGID:MIN?]
Returns the last decoded value of MIN (ASCII String) selected by *n*. Up to four instances can be returned. Range of *n* is 0 to 3. Returns -1 if already returned or not available.

If IDT indicates a 34 bit MSID then the value is also stored as a Mobile Identification Number.

EHI?
[FDCCH:SPACH:EHI?]
Returns the last decoded value of Extended Header Information (1 bit value). Returns -1 if already returned or not available.

MEA?
[FDCCH:SPACH:MEA?]
Returns the last decoded value of Message Encryption Algorithm (2 bit value). Returns -1 if already returned or not available.

MEK?
[FDCCH:SPACH:MEK?]
Returns the last decoded value of Message Encryption Key (2 bit value). Returns -1 if already returned or not available.

ARM?
[FDCCH:SPACH:ARM?]
Returns the last decoded value of ARQ Response Mode (1 bit value). Returns -1 if already returned or not available.

FRNO?
[FDCCH:SPACH:FRNO?]
Returns the last decoded value of Frame Number (5 bit value). Returns -1 if already returned or not available.

FDCCH:

SPACH:

GA?
[FDCCH:SPACH:GA?]
Returns the last decoded value of Go Away (1 bit value). Returns -1 if already returned or not available.

L3DATA:

SELect *n*
[FDCCH:SPACH:L3DATA:SELect]
Returns the last decoded value of MIN (ASCII String) selected by *n*. (There may be up to four L3DATA messages in a SPACH message.) Range of *n* is 0 to 3. Returns -1 if already returned or not available.

SELect?
Returns the number of the selected L3DATA Message (2 bit value).

MSGtype?
[FDCCH:SPACH:MSGtype?]
Returns the last decoded value of Message Type from the selected L3DATA Message. Returns -1 if already returned or not available. This command returns the following valid message types:

ANALOG	AUDIT	BSCHALCON
BSMC	CAPABILITY	DIGITAL
DRETRY	MSGWTG	PAGE
PU	R-DATA	R-DATA ACCEPT
R-DATA REJECT	REG ACCEPT	REG REJECT
RELEASE	REORDER/INTER	SOC
SPACH NOTIF	SSDUP	TEST REG
UCHAL	USER ALERT	INVALID

INVALID is returned if an illegal message code was decoded.

PD?
[FDCCH:SPACH:PD?]
Returns the last decoded value of Protocol Discriminator (2 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

MEM?
[FDCCH:SPACH:MEM?]
Returns the last decoded value of Message Encryption Mode (1 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

FDCCH:

SPACH:

SCC?
[FDCCH:SPACH:SCC?]
Returns the last decoded value of SAT Color Code (2 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

VMAC?
[FDCCH:SPACH:VMAC?]
Returns the last decoded value of Voice Mobile Attenuation Code (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

CHAN?
[FDCCH:SPACH:CHAN?]
Returns the last decoded value of CHAN (11 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

PROTocol?
[FDCCH:SPACH:PROTocol?]
Returns the last decoded value of Protocol Version (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

SUBaddress:

PT?
[FDCCH:SPACH:SUBaddress:PT?]
Returns the last decoded value of Subaddress Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

LENGth?
[FDCCH:SPACH:SUBaddress:LENGth?]
Returns the last decoded value of Length of Subaddress Info content (8 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

ODD_EVEN?
[FDCCH:SPACH:SUBaddress:ODD_EVEN?]
Returns the last decoded value of Subaddress Odd/Even indicator (1 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

TYPE?
[FDCCH:SPACH:SUBaddress:TYPE?]
Returns the last decoded value of Type of Subaddress (3 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

FDCCH:

SPACH:

SUBaddress:

REServed?
[FDCCH:SPACH:SUBaddress:REServed?]
Returns the last decoded value of the combination of the two Subaddress Reserved fields (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

ADDRess?
[FDCCH:SPACH:SUBaddress:ADDress?]
Returns the last decoded value of Subaddressing (ASCII String) from the selected L3DATA Message. Returns -1 if already returned or not available.

DTX:

PT?
[FDCCH:SPACH:DTX:PT?]
Returns the last decoded value of DTX Support Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

SUPport?
[FDCCH:SPACH:DTX:SUPport?]
Returns the last decoded value of DTX Support (2 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

DISPlay:

PT?
[FDCCH:SPACH:DISPlay:PT?]
Returns the last decoded value of Display Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

LENGth?
[FDCCH:SPACH:DISPlay:LENGth?]
Returns the last decoded value of Length of Display Info (8 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

CHARacter? ***n***
[FDCCH:SPACH:DISPlay:CHARacter?]
Returns the last decoded value Display Character (8 bit value) selected by *n* from the selected L3DATA Message. Up to 82 instances can be returned. Range of *n* is 0 to 81. Returns -1 if already returned or not available.

Validity is determined by **FDCCH:SPACH:DISPlay:LENGth?**.

FDCCH:

SPACH:

REREG?
[FDCCH:SPACH:REREG?]
Returns the last decoded value of Forced Re-registration (1 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

DEBUG?
[FDCCH:SPACH:DEBUG?]
Returns the last decoded value of Debug Display Allowed (1 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

AUTHBS?
[FDCCH:SPACH:AUTHBS?]
Returns the last decoded value of AUTHBS (18 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

BSMC?
[FDCCH:SPACH:BSMC?]
Returns the last decoded value of BSMC (Base Station Manufacture Code) (8 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

CUSTOM:

LENGth?
[FDCCH:SPACH:CUSTOM:LENGth?]
Returns the last decoded value of Length of Custom Control in octets (8 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

CONTrol? *n*
[FDCCH:SPACH:CUSTOM:CONTrol?]
Returns the last decoded value of Custom Control (8 bit value) selected by *n* from the selected L3DATA Message. Up to 256 instances can be returned. Range of *n* is 0 to 255. Returns -1 if already returned or not available.

Validity is determined by **FDCCH:SPACH:CUSTOM:LENGth?**.

DVCC?
[FDCCH:SPACH:DVCC?]
Returns the last decoded value of DVCC (8 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

DMAC?
[FDCCH:SPACH:DMAC?]
Returns the last decoded value of DMAC (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

FDCCH:

SPACH:

ATS?
[FDCCH:SPACH:ATS?]
Returns the last decoded value of ATS (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

SB?
[FDCCH:SPACH:SB?]
Returns the last decoded value of SB (1 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

TA?
[FDCCH:SPACH:TA?]
Returns the last decoded value of Time Alignment (5 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

MODE:

DIC?
[FDCCH:SPACH:MODE:DIC?]
Returns the last decoded value of Delay Interval Compensation Mode (1 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

VOICE:

PT?
[FDCCH:SPACH:MODE:VOICE:PT?]
Returns the last decoded value of Voice Mode Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

VC?
[FDCCH:SPACH:MODE:VOICE:VC?]
Returns the last decoded value of Voice Coder (3 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

PM_V?
[FDCCH:SPACH:MODE:VOICE:PM_V?]
Returns the last decoded value of Voice Privacy Mode (3 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

MEM:

PT?
[FDCCH:SPACH:MODE:MEM:PT?]
Returns the last decoded value of Message Encryption Mode Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

FDCCH:

SPACH:

MODE:

MEM:

MEA?
[FDCCH:SPACH:MODE:MEM:MEA?]
Returns the last decoded value of Message Encryption Algorithm (3 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

MED?
[FDCCH:SPACH:MODE:MEM:MED?]
Returns the last decoded value of Message Encryption Domain (3 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

MEK?
[FDCCH:SPACH:MODE:MEM:MEK?]
Returns the last decoded value of Message Encryption Key (3 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

HYPERband:

PT?
[FDCCH:SPACH:HYPERband:PT?]
Returns the last decoded value of Hyperband Info Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

INFO?
[FDCCH:SPACH:HYPERband:INFO?]
Returns the last decoded value of Hyperband Info (2 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

LT?
[FDCCH:SPACH:LT?]
Returns the last decoded value of Last Try (1 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

FDCCH:

SPACH:

FLAG:

PT?
[FDCCH:SPACH:FLAG:PT?]
Returns the last decoded value of the RCF and AUTH flags Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

RCF?
[FDCCH:SPACH:FLAG:RCF?]
Returns the last decoded value of RCF flag (1 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

AUTH?
[FDCCH:SPACH:FLAG:AUTH?]
Returns the last decoded value of AUTH flag (1 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

MSGWTG:

NV?
[FDCCH:SPACH:MSGWTG:NV?]
Returns the last decoded value of Message Waiting Info Number of Values (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

TYPE? *n*
[FDCCH:SPACH:MSGWTG:TYPE?]
Returns the last decoded value of Type of Message Waiting (4 bit value) selected by *n* from the selected L3DATA Message. Up to 16 instances can be returned. Range of *n* is 0 to 15. Returns -1 if already returned or not available.

NUMber? *n*
[FDCCH:SPACH:MSGWTG:NUMber?]
Returns the last decoded value of Number of Messages Waiting (6 bit value) selected by *n* from the selected L3DATA Message. Up to 16 instances can be returned. Range of *n* is 0 to 15. Returns -1 if already returned or not available.

SERVice?
[[FDCCH:SPACH:SERVice?]]
Returns the last decoded value of Service Code (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

FDCCH:

SPACH:

SIGnal:

PT?
[FDCCH:SPACH:SIGnal:PT?]
Returns the last decoded value of Signal Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

PITCH?
[FDCCH:SPACH:SIGnal:PITCH?]
Returns the last decoded value of Signal Pitch (2 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

CADence?
[FDCCH:SPACH:SIGnal:CADence]
Returns the last decoded value of Signal Cadence (6 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

DURation?
[FDCCH:SPACH:SIGnal:DURation?]
Returns the last decoded value of Signal Duration (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

CALLED:

PT?
[FDCCH:SPACH:CALLED:PT?]
Returns the last decoded value of Called Party Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

LENGth?
[FDCCH:SPACH:CALLED:LENGth?]
Returns the last decoded value of Called Party Length of Address Info (8 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

TYPE?
[FDCCH:SPACH:CALLED:TYPE?]
Returns the last decoded value of Called Party Type of Number (3 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

PLANid?
[FDCCH:SPACH:CALLED:PLANid?]
Returns the last decoded value of Called Party Numbering Plan Identification (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

FDCCH:

SPACH:

CALLED:

ENCoding?
[FDCCH:SPACH:CALLED:ENCoding?]
Returns the last decoded value of Called Party Address Encoding (1 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

ADDRess?
[FDCCH:SPACH:CALLED:ADDRess?]
Returns the last decoded value of Called Party Address (ASCII String) from the selected L3DATA Message. Returns -1 if already returned or not available.

SUBaddress:

PT?
[FDCCH:SPACH:CALLED:SUBaddress:PT?]
Returns the last decoded value of Called Party Subaddress Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

LENGth?
[FDCCH:SPACH:CALLED:SUBaddress:LENGth?]
Returns the last decoded value of Length of Called Party Subaddress Info (8 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

ODD_EVEN?
[FDCCH:SPACH:CALLED:SUBaddress:ODD_EVEN?]
Returns the last decoded value of Called Party Subaddress Odd/Even Indicator (1 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

TYPE?
[FDCCH:SPACH:CALLED:SUBaddress:TYPE?]
Returns the last decoded value of Type of Called Party Subaddress (3 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

REServed?
[FDCCH:SPACH:CALLED:SUBaddress:REServed?]
Returns the last decoded value of the combination of the two Called Party Subaddress Reserved fields (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

FDCCH:

SPACH:

CALLED:

SUBaddress:

ADDRess?
[FDCCH:SPACH:CALLED:SUBaddress:ADDRess?]
Returns the last decoded value of Called Party Subaddress (ASCII String) from the selected L3DATA Message. Returns -1 if already returned or not available.

CALLING:

PT?
[FDCCH:SPACH:CALLING:PT?]
Returns the last decoded value of Calling Party Number Parameter Type available (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not.

LENGth?
[FDCCH:SPACH:CALLING:LENGth?]
Returns the last decoded value of Calling Party Length of Address Info (8 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

TYPE?
[FDCCH:SPACH:CALLING:TYPE?]
Returns the last decoded value of Type of Calling Party Number (3 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

PLANid?
[FDCCH:SPACH:CALLING:PLANid?]
Returns the last decoded value of Calling Party Number Plan Identification (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

ENCoding?
[FDCCH:SPACH:CALLING:ENCoding?]
Returns the last decoded value of Calling Party Address Encoding (1 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

ADDRess?
[FDCCH:SPACH:CALLING:ADDRess?]
Returns the last decoded value of Calling Party Number Address (ASCII String) from the selected L3DATA Message. Returns -1 if already returned or not available.

FDCCH:

SPACH:

CALLING:

SUBaddress:

PT?
[FDCCH:SPACH:CALLING:SUBaddress:PT?]
Returns the last decoded value of Calling Party Subaddress Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

LENGth?
[FDCCH:SPACH:CALLING:SUBaddress:LENGth?]
Returns the last decoded value of Calling Party Length of Subaddress Info (8 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

ODD_EVEN?
[FDCCH:SPACH:CALLING:SUBaddress:ODD_EVEN?]
Returns the last decoded value of Calling Party Subaddress Odd/Even Indicator (1 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

TYPE?
[FDCCH:SPACH:CALLING:SUBaddress:TYPE?]
Returns the last decoded value of Calling Party Type of Subaddress (3 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

REServed?
[FDCCH:SPACH:CALLING:SUBaddress:REServed?]
Returns the last decoded value of the combination of the two Calling Party Subaddress Reserved fields (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

ADDRess? ***n***
[FDCCH:SPACH:CALLING:SUBaddress:ADDRess?]
Returns the last decoded value of Calling Party Subaddress (8 bit value) selected by *n* from the selected L3DATA Message. Up to 20 instances can be returned. Range of *n* is 0 to 19. Returns -1 if already returned or not available.

FDCCH:

SPACH:

CALLING:

PRESentation:

PT?
[FDCCH:SPACH:CALLING:PRESentation:PT?]
Returns the last decoded value of Calling Party Presentation Indicator Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

PI?
[FDCCH:SPACH:CALLING:PRESentation:PI?]
Returns the last decoded value of Presentation Indicator (2 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

SI?
[FDCCH:SPACH:CALLING:PRESentation:SI?]
Returns the last decoded value of Screening Indicator (2 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

RN?
[FDCCH:SPACH:RN?]
Returns the last decoded value of Request Number (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

RTRANSaction?
[FDCCH:SPACH:RTRANSaction?]
Returns the last decoded value of R-Transaction Identifier (8 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

RDATA_UNIT:

LENGth?
[FDCCH:SPACH:RDATA_UNIT:LENGth?]
Returns the last decoded value of R-Data Unit Length (8 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

FDCCH:

SPACH:

RDATA_UNIT:

HLP:

IDentifier?
[FDCCH:SPACH:RDATA_UNIT:HLP:IDentifier?]
Returns the last decoded value of R-Data Unit Higher Layer Protocol Identifier (8 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

DATA? *n*
[FDCCH:SPACH:RDATA_UNIT:HLP:DATA?]
Returns the last decoded value of Higher Layer Protocol Data Unit (8 bit value) selected by *n* from the selected L3DATA Message. Up to 255 instances can be returned. Range of *n* is 0 to 254.

Validity is determined by **FDCCH:SPACH:RDATA_UNIT:LENGth?**.

MESSage:

CENTer:

PT?
[FDCCH:SPACH:MESSage:CENTer:PT?]
Returns the last decoded value of Message Center Address Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

LENGth?
[FDCCH:SPACH:MESSage:CENTer:LENGth?]
Returns the last decoded value of Message Center Length of Address Info (8 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

TYPE?
[FDCCH:SPACH:MESSage:CENTer:TYPE?]
Returns the last decoded value of Message Center Address Type of Number (3 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

PLANid?
[FDCCH:SPACH:MESSage:CENTer:PLANid?]
Returns the last decoded value of Message Center Numbering Plan Identification (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

FDCCH:

SPACH:

MESSage:

CENTer:

ENCoding?
[FDCCH:SPACH:MESSage:CENTer:ENCoding?]
Returns the last decoded value of Message Center Address Encoding (1 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

ADDRess?
[FDCCH:SPACH:MESSage:CENTer:ADDRess?]
Returns the last decoded value of Message Center Address (ASCII String) from the selected L3DATA Message. Returns -1 if already returned or not available.

USER:

DEST:

PT?
[FDCCH:SPACH:USER:DEST:PT?]
Returns the last decoded value of User Destination Address Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

LENGth?
[FDCCH:SPACH:USER:DEST:LENGth?]
Returns the last decoded value of User Destination Length of Address Info (8 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

TYPE?
[FDCCH:SPACH:USER:DEST:TYPE?]
Returns the last decoded value of User Destination Type of Number (3 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

PLANid?
[FDCCH:SPACH:USER:DEST:PLANid?]
Returns the last decoded value of User Destination Address Identification Plan (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

FDCCH:

SPACH:

USER:

DEST:

ENCoding?
[FDCCH:SPACH:USER:DEST:ENCoding?]
Returns the last decoded value of User Destination Address Encoding (1 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

ADDRess?
[FDCCH:SPACH:USER:DEST:ADDRess?]
Returns the last decoded value of User Destination Address (ASCII String) from the selected L3DATA Message. Returns -1 if already returned or not available.

SUBaddress:

PT?
[FDCCH:SPACH:USER:DEST:SUBaddress:PT?]
Returns the last decoded value of User Destination Subaddress Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

LENGth?
[FDCCH:SPACH:USER:DEST:SUBaddress:LENGth?]
Returns the last decoded value of User Destination Length of Subaddress Info (8 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

ODD_EVEN?
[FDCCH:SPACH:USER:DEST:SUBaddress:ODD_EVEN?]
Returns the last decoded value of User Destination Subaddress Odd/Even Indicator (1 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

TYPE?
[FDCCH:SPACH:USER:DEST:SUBaddress:TYPE?]
Returns the last decoded value of User Destination Type of Subaddress (3 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

REServed?
[FDCCH:SPACH:USER:DEST:SUBaddress:REServed?]
Returns the last decoded value of the combination of the two User Destination Subaddress Reserved fields (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

FDCCH:

SPACH:

USER:

DEST:

SUBaddress:

ADDRess?
[FDCCH:SPACH:USER:DEST:SUBaddress:ADDRess?]
Returns the last decoded value of User Destination Subaddressing (ASCII String) from the selected L3DATA Message. Returns -1 if already returned or not available.

GROUP:

PT?
[[FDCCH:SPACH:USER:GROUP:PT?]]
Returns the last decoded value of User Group Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

STATus?
[FDCCH:SPACH:USER:GROUP:STATus?]
Returns the last decoded value of User Group Status (2 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

TYPE?
[FDCCH:SPACH:USER:GROUP:TYPE?]
Returns the last decoded value of User Group Type (2 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

ID:

MS?
[FDCCH:SPACH:USER:GROUP:ID:MS?]
Returns the 18 Most Significant bits of User Group Identification. Returns -1 if already returned or not available.

LS?
[FDCCH:SPACH:USER:GROUP:ID:LS?]
Returns the 32 Least Significant bits of User Group Identification. Returns -1 if already returned or not available.

FDCCH:

SPACH:

USER:

ORIG:

PT?
[FDCCH:SPACH:USER:ORIG:PT?]
Returns the last decoded value of User Originating Address Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

LENGth?
[FDCCH:SPACH:USER:ORIG:LENGth?]
Returns the last decoded value of User Originating Length of Address Info (8 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

TYPE?
[FDCCH:SPACH:USER:ORIG:TYPE?]
Returns the last decoded value of User Originating Type of Address (3 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

PLANid?
[FDCCH:SPACH:USER:ORIG:PLANid?]
Returns the last decoded value of User Originating Address Identification Plan (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

ENCoding?
[FDCCH:SPACH:USER:ORIG:ENCoding?]
Returns the last decoded value of User Originating Address Encoding (1 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

ADDRess?
[FDCCH:SPACH:USER:ORIG:ADDRess?]
Returns the last decoded value of User Originating Address (ASCII String) from the selected L3DATA Message. Returns -1 if already returned or not available.

FDCCH:

SPACH:

USER:

ORIG:

SUBaddress:

PT?
[FDCCH:SPACH:USER:ORIG:SUBaddress:PT?]
Returns the last decoded value of User Originating Subaddress Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

LENGth?
[FDCCH:SPACH:USER:ORIG:SUBaddress:LENGth?]
Returns the last decoded value of User Originating Length of Subaddress Info (8 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

ODD_EVEN?
[FDCCH:SPACH:USER:ORIG:SUBaddress:ODD_EVEN?]
Returns the last decoded value of User Originating Subaddress Odd/Even Indicator (1 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

TYPE?
[FDCCH:SPACH:USER:ORIG:SUBaddress:TYPE?]
Returns the last decoded value of User Originating Type of Subaddress (3 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

REServed?
[FDCCH:SPACH:USER:ORIG:SUBaddress:REServed?]
Returns the last decoded value of the combination of the two User Originating Subaddress Reserved fields (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

ADDRess? *n*
[FDCCH:SPACH:USER:ORIG:SUBaddress:ADDRess?]
Returns the last decoded value of User Originating Subaddress (8 bit value) selected by *n* from the selected L3DATA Message. Up to 20 instances can be returned. Range of *n* is 0 to 19. Returns -1 if already returned or not available.

FDCCH:

SPACH:

PFC:

PT?
[FDCCH:SPACH:PFC:PT?]
Returns the last decoded value of PFC Assignment Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

ASSIGNment?
[FDCCH:SPACH:PFC:ASSIGNment?]
Returns the last decoded value of PFC Assignment (3 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

RNUM:

PT?
[FDCCH:SPACH:RNUM:PT?]
Returns the last decoded value of RNUM List Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

NUMber?
[FDCCH:SPACH:RNUM:NUMber?]
Returns the last decoded value of Number of RNUMs (6 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

LIST? *n*
[FDCCH:SPACH:RNUM:LIST?]
Returns the last decoded value of RNUM (10 bit value) selected by *n* from the selected L3DATA Message. Up to 50 instances can be returned. Range of *n* is 0 to 49. Returns -1 if already returned or not available.

Validity is determined by **FDCCH:SPACH:RNUM:NUMber?**

MSID:

PT?
[FDCCH:SPACH:MSID:PT?]
Returns the last decoded value of MSID Assignment Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

IDT?
[FDCCH:SPACH:MSID:IDT?]
Returns the last decoded value of MSID Assignment IDT (2 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

FDCCH:

SPACH:

MSID:

ASSIGNment?
[FDCCH:SPACH:MSID:ASSIGNment?]
Returns the last decoded value of MSID Assignment (24 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

PSID_RSID:

AVAILable:

PT?
[FDCCH:SPACH:PSID_RSID:AVAILable:PT?]
Returns the last decoded value of PSID/RSID Available Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

NUMber?
[FDCCH:SPACH:PSID_RSID:AVAILable:NUMber?]
Returns the last decoded value of Number of PSID/RSID (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

TYPE? *n*
[FDCCH:SPACH:PSID_RSID:AVAILable:TYPE?]
Returns the last decoded value of PSID/RSID Type Indicator (1 bit value) selected by *n* from the selected L3DATA Message. Up to 16 instances can be returned. Range of *n* is 0 to 15. Returns -1 if already returned or not available.

Validity is determined by **FDCCH:SPACH:PSID_RSID:AVAILable:NUMber?**.

VALUE? *n*
[FDCCH:SPACH:PSID_RSID:AVAILable:VALUE?]
Returns the last decoded value of PSID/RSID Value (1 bit value) selected by *n* from the selected L3DATA Message. Up to 16 instances can be returned. Range of *n* is 0 to 15. Returns -1 if already returned or not available.

Validity is determined by **FDCCH:SPACH:PSID_RSID:AVAILable:NUMber?**.

MAP?
[FDCCH:SPACH:PSID_RSID:MAP?]
Returns the last decoded value of PSID/RSID Map (16 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

FDCCH:

SPACH:

DIRectory:

PT?
[FDCCH:SPACH:DIRectory:PT?]
Returns the last decoded value of Directory Address Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

LENGth?
[FDCCH:SPACH:DIRectory:LENGth?]
Returns the last decoded value of Director Length of Address Info (8 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

TYPE?
[FDCCH:SPACH:DIRectory:TYPE?]
Returns the last decoded value of Directory Address Type of Number (3 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

PLANid?
[FDCCH:SPACH:DIRectory:PLANid?]
Returns the last decoded value of Directory Address Identification Plan (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

ENCoding?
[FDCCH:SPACH:DIRectory:ENCoding?]
Returns the last decoded value of Directory Address Encoding (1 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

ADDRess?
[FDCCH:SPACH:DIRectory:ADDRess?]
Returns the last decoded value of Directory Address (ASCII String) from the selected L3DATA Message. Returns -1 if already returned or not available.

SUBaddress:

PT?
[FDCCH:SPACH:DIRectory:SUBaddress:PT?]
Returns the last decoded value of Directory Subaddress Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

FDCCH:

SPACH:

DIRectory:

SUBaddress:

LENGth?
[FDCCH:SPACH:DIRectory:SUBaddress:LENGth?]
Returns the last decoded value of Directory Length of Subaddress Info (8 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

ODD_EVEN?
[FDCCH:SPACH:DIRectory:SUBaddress:ODD_EVEN?]
Returns the last decoded value of Directory Subaddress Odd/Even Indicator (1 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

TYPE?
[FDCCH:SPACH:DIRectory:SUBaddress:TYPE?]
Returns the last decoded value of Directory Type of Subaddress (3 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

REServed?
[FDCCH:SPACH:DIRectory:SUBaddress:REServed?]
Returns the last decoded value of the combination of the two Directory Subaddress Reserved fields (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

ADDRess? *n*
[FDCCH:SPACH:DIRectory:SUBaddress:ADDRess?]
Returns the last decoded value of Directory Subaddress (8 bit value) selected by *n* from the selected L3DATA Message. Up to 20 instances can be returned. Range of *n* is 0 to 19. Returns -1 if already returned or not available.

REJect:

REGistration:

CAUSE?
[FDCCH:SPACH:REJect:REGistration:CAUSE?]
Returns the last decoded value of Registration Reject Cause (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

FDCCH:

SPACH:

REJect:

REGistration:

TIME:

PT?
[FDCCH:SPACH:REJect:REGistration:TIME:PT?]
Returns the last decoded value of Reject Time Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

LOWer?
[FDCCH:SPACH:REJect:REGistration:TIME:LOWer?]
Returns the last decoded value of Reject Lower Time Boundary in 100 SF (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

UPPer?
[FDCCH:SPACH:REJect:REGistration:TIME:UPPer?]
Returns the last decoded value of Reject Upper Time Boundary in 100 SF (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

RDATA:

CAUSE?
[FDCCH:SPACH:REJect:RDATA:CAUSE?]
Returns the last decoded value of R-CAUSE (7 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

SPARE?
[FDCCH:SPACH:REJect:RDATA:SPARE?]
Returns the last decoded value of R-CAUSE Spare (1 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

RELease:

CAUSE?
[FDCCH:SPACH:RELease:CAUSE?]
Returns the last decoded value of Release Cause (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

FDCCH:

SPACH:

REorder:

CAUSE?
[FDCCH:SPACH:REorder:CAUSE?]
Returns the last decoded value of Reorder/Intercept Cause (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

TONE?
[FDCCH:SPACH:REorder:TONE?]
Returns the last decoded value of Tone Indicator (2 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

SOC?
[FDCCH:SPACH:SOC?]
Returns the last decoded value of SOC (System Operator Code) (12 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

NOTification?
[FDCCH:SPACH:NOTification?]
Returns the last decoded value of SPACH Notification (6 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

RANDSSD1?
[FDCCH:SPACH:RANDSSD1?]
Returns the last decoded value of the 24 most significant bits of RANDSSD from the selected L3DATA Message. Returns -1 if already returned or not available.

RANDSSD2?
[FDCCH:SPACH:RANDSSD2?]
Returns the last decoded value of the 32 least significant bits of RANDSSD from the selected L3DATA Message. Returns -1 if already returned or not available.

ALPHA:

SID:

PT?
[FDCCH:SPACH:ALPHA:SID:PT?]
Returns the last decoded value of Alphanumeric System ID Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

LENGth?
[FDCCH:SPACH:ALPHA:SID:LENGth?]
Returns the last decoded value of Length of Alphanumeric System ID (8 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

FDCCH:

SPACH:

ALPHA:

SID:

CHARacters?
[FDCCH:SPACH:ALPHA:SID:CHARacters?]
Returns the last decoded value of Alphanumeric System ID (ASCII String) from the selected L3DATA Message. Returns -1 if already returned or not available.

PSID_RSID:

PT?
[FDCCH:SPACH:ALPHA:PSID_RSID:PT?]
Returns the last decoded value of Alphanumeric PSID/RSID List Parameter Type (4 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

LENGth?
[FDCCH:SPACH:ALPHA:PSID_RSID:LENGth?]
Returns the last decoded value of Length of Alphanumeric PSID/RSID List (8 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

NAME:

LENGth? *n*
[FDCCH:SPACH:ALPHA:PSID_RSID:NAME:LENGth?]
Returns the last decoded value of Length of PSID/RSID Alphanumeric Name (4 bit value) selected by *n* from the selected L3DATA Message. Up to 16 instances can be returned. Range of *n* is 0 to 15. Returns -1 if already returned or not available.

CHARacters? *n*
[FDCCH:SPACH:ALPHA:PSID_RSID:NAME:CHARacters?]
Returns the last decoded value of Alphanumeric PSID/RSID Display Characters (ASCII String) selected by *n* from the selected L3DATA Message. Up to 16 instances can be returned. Range of *n* is 0 to 15. Returns -1 if already returned or not available.

RANDU?
[FDCCH:SPACH:RANDU?]
Returns the last decoded value of RANDU (24 bit value) from the selected L3DATA Message. Returns -1 if already returned or not available.

7-16 DCCH CELL SITE SIMULATION

This section contains the TMAC commands necessary to simulate the transmit portion of a Base Station operating on the DCCH. The RDCCH monitor commands, also considered a part of Cell Site Simulation, are described in Section 7-17.

7-16-1 SETUP COMMANDS

To configure the FM/AM-1600CSA for Cell Site Simulation, use the following TMAC commands:

- **CSS:SETup** (See 7-9-1)
- **CSS:CHANnel** (See 7-9-1)
- **CSS:RFLVL** (See 7-9-1)
- **CSS:SLOT** (See 7-9-1)
- **CSS:RATE** (See 7-9-1)

7-16-2 SUPERFRAME SETUP

The FDCCH Generator has all the TMAC commands and operations needed to build and maintain a Superframe. Each phase of the Superframe can be changed as the Superframe is being transmitted. A TMAC command returns the Superframe phase currently being transmitted. This enables a TMAC program, that changes the data, to become synchronized with the Superframe cycle.

CSS:

FDCCH:

SUPERframe:

SFP *n,m*
[CSS:FDCCH:SUPERframe:SFP]
Selects a Super Frame Phase (*m*) within a Superframe slot (*n*) being programmed. Range of *n* is 0 to 31; range of *m* is 0 to 255.

The CRC is performed by the FM/AM-1600CSA.

SFP? *n*
Returns current value of SFP indexed by *n*. Range of *n* is 0 to 31.

BRI *n,m*
[CSS:FDCCH:SUPERframe:BRI]
Specifies Busy/Idle/Reserved (*m*) within a selected Superframe slot (*n*) being programmed. Range of *n* is 0 to 31; range of *m* is 0 to 63.

BRI? *n*
Returns current value of BRI indexed by *n*. Range of *n* is 0 to 31.

PE *n,m*
[CSS:FDCCH:SUPERframe:PE]
Specifies Partial Echo (*m*) within a Superframe slot (*n*) being programmed. Range of *n* is 0 to 31; range of *m* is 0 to 127.

The CRC is performed by the FM/AM-1600CSA.

PE? *n*
Returns current value of PE indexed by *n*. Range of *n* is 0 to 31.

RN *n,m*
[CSS:FDCCH:SUPERframe:RN]
Specifies Received/Not Received (*m*) within a Superframe slot (*n*) being programmed. Range of *n* is 0 to 31; range of *m* is 0 to 31.

RN? *n*
Returns current value of RN indexed by *n*. Range of *n* is 0 to 31.

CSS:

FDCCH:

SUPERframe:

DATA *n,x,word*
[CSS:FDCCH:SUPERframe:DATA]
Specifies one of the 7 words (indexed by *x*) that comprise the data transmitted per selected Superframe slot (*n*). Range of *n* is 0 to 31; range of *x* is 0 to 6; range of *word* is 0 to #hFFFF.

The data transmitted in a Superframe slot consists of 109 bits. The data is divided into seven 16 bit words. *x* = 0 selects the most significant word. *x* = 6 selects the least significant word.

The CRC, convolutional encoding and interleaving processes are performed by the FM/AM-1600CSA.

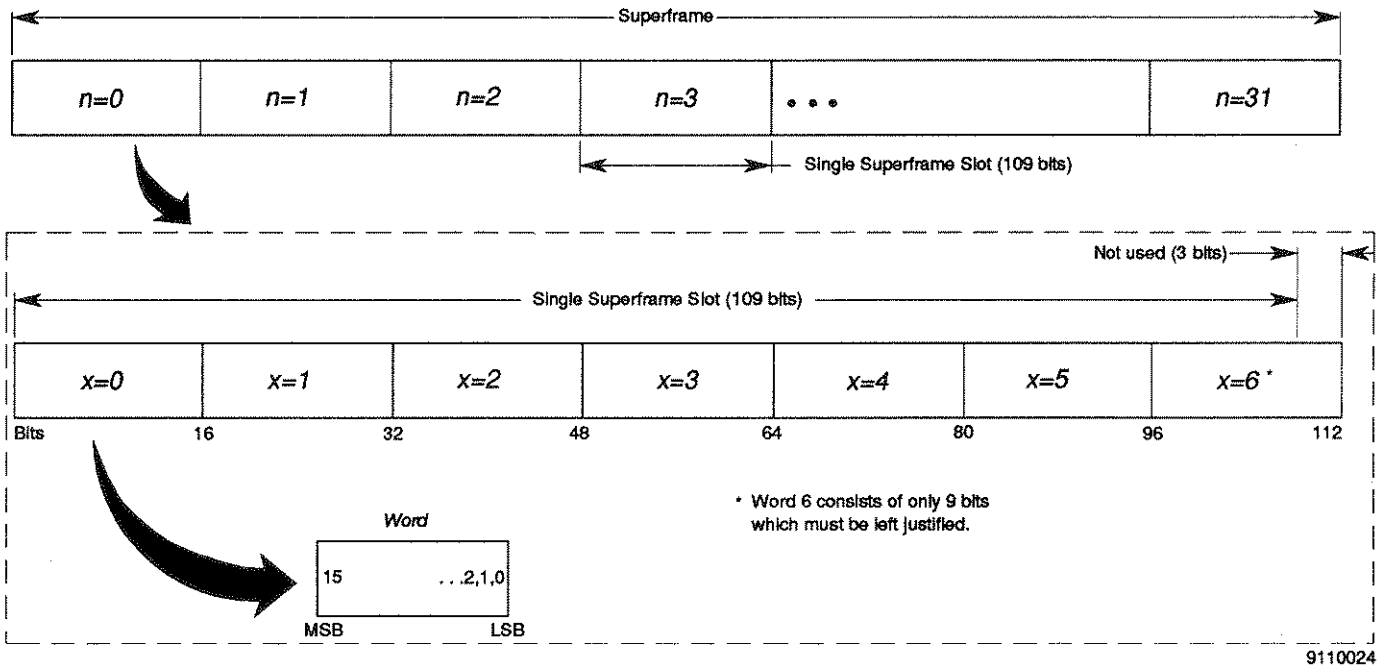


Figure 7-11 Superframe Data Message

DATA? *n,m*
Returns current value of DATA indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 6.

CSS:

FDCCH:

SUPERframe:

TYPE *n,m*
[CSS:FDCCH:SUPERframe:TYPE]
Specifies Type (*m*) of data in Superframe slot (*n*). Range of *n* is 0 to 31, range of *m* is 0 to 4.

m	TYPE
0	F-BCCH
1	E-BCCH
2	S-BCCH
3	SPACH
4	RESERVED

TYPE? *n*
Returns current value of TYPE indexed by *n*. Range of *n* is 0 to 31.

DVCC *n*
[CSS:FDCCH:SUPERframe:DVCC]
Specifies Digital Verification Color Code. Range of *n* is 0 to 255.

DVCC must be specified in order to calculate the CRC for each type except F-BCCH.

DVCC?
Returns current value of DVCC. Range of *n* is 0 to 31.

STARt
[CSS:FDCCH:SUPERframe:STARt]
Starts the superframe generating task.

STOP
[CSS:FDCCH:SUPERframe:STOP]
Stops the superframe generating task.

CSS:

FDCCH:

SUPERframe:

ACCess:

TYPE:

RANDom
[CSS:FDCCH:SUPERframe:ACCess:TYPE:RANDom]
Programs the FM/AM-1600CSA to allow a mobile station to make a Random access.

The values that make up the SCF (Shared Channel Feedback) are overwritten when an access occurs.

REServed
[CSS:FDCCH:SUPERframe:ACCess:TYPE:REServed]
Programs the FM/AM-1600CSA to allow a mobile station to make a Random access.

The values that make up the SCF (Shared Channel Feedback) are overwritten when an access occurs.

PROGram
[CSS:FDCCH:SUPERframe:ACCess:TYPE:PROGram]
The values of SCF will be setup as pre-programmed by the following command.

NONE
[CSS:FDCCH:SUPERframe:ACCess:TYPE:NONE]
The values that make up the SCF do not change when an access occurs.

TYPE?
[CSS:FDCCH:SUPERframe:ACCess:TYPE?]
Returns current value of TYPE.

PE *n*
[CSS:FDCCH:SUPERframe:ACCess:PE]
Programs the Partial Echo used during a Random or Reserved access. Range of *n* is 0 to 127.

The CRC is performed by the FM/AM-1600CSA.

PE?
Returns current value of PE.

CSS:

FDCCH:

SUPERframe:

ACCess:

SCF *n,m*
[CSS:FDCCH:SUPERframe:ACCess:SCF]
Pre-program the Shared Channel Feedback response (*m*) in the selected frame (*n*) of a RACH. Range of *n* is 0 to 79; range of *m* is 0 to 11.

m	PE MATCH	R/N	BRI
0	Match	R	B
1	Mismatch	R	B
2	Match	R	R
3	Mismatch	R	R
4	Match	R	I
5	Mismatch	R	I
6	Match	N	B
7	Mismatch	N	B
8	Match	N	R
9	Mismatch	N	R
10	Match	N	I
11	Mismatch	N	I

LEGEND	
R	Received
N	Not Received
B	Busy
R	Reserved
I	Idle

Table 7-1 Shared Channel Feedback Response

SCF? *n*
Returns current value of SCF indexed by *n*. Range of *n* is 0 to 31.

NUMber?
[CSS:FDCCH:SUPERframe:NUMber?]
Returns the current number of the selected slot in the superframe being transmitted.

The returned value of this command indicates which slot of the superframe is currently being transmitted. Using this information, a program or macro can change the data in the superframe without disturbing the slot currently being transmitted.

7-16-3 F-BCCH COMMANDS

The F-BCCH commands are used to build data into the F-BCCH slots of the superframe. These commands enable the user to construct the F-BCCH by specifying the Layer 3 BCCH message types and data fields. The F-BCCH slots can be built and then, with the aid of the **CSS:FDCCH:SUPERframe:NUMber?** command, be placed at the start of the superframe while the E-BCCH slots or SPACH slots are being transmitted. This enables the F-BCCH to be constantly updated, even as the superframe is being transmitted. In addition, any slot of the F-BCCH can be manually modified or distorted by the **CSS:FDCCH:SUPERframe:DATA** command.

CSS:

FBCCH:

BUILD
[CSS:FBCCH:BUILD]
This command builds the data that makes up the F-BCCH. This data can then be returned by the **CSS:FBCCH:DATA?** command defined below. Before executing this command, the message types and data fields that make up the F-BCCH should be programmed. This command then, takes that information and generates all the data that makes up the data field in each slot of the F-BCCH section of the superframe. The superframe can then be programmed with the **CSS:FBCCH:PROGram** command defined below. Perform the following steps to build an F-BCCH:

1. Enable the desired Message Types.
2. Enable the desired optional fields associated with the selected message types.
3. Program the data fields associated with the enabled message types.
4. Build the F-BCCH.
5. Program the superframe.

LENGth?
[CSS:FBCCH:LENGth?]
After the **CSS:FBCCH:BUILD** command has been executed, the Length of the F-BCCH in slots, can be returned. Knowing the length of the F-BCCH aids the user in setting up the number of F-BCCH data fields in the DCCH Structure.

DATA? *n,m*
[CSS:FBCCH:DATA?]
This command returns the F-BCCH data that has been built. Returns the 16 bit word indexed by *m* from slot (*n*). Range of *n* is 0 to 10; range of *m* is 0 to 6.

Each slots consist of 109 bits. The 16 most significant bits of the data are returned when *m* = 0; the 13 least significant bits of data are returned when *m* = 6. The data returned is left justified.

The above data format correlates with the data format used in the **CSS:FDCCH:SUPER:DATA *n,x,word*** command.

CSS:

FBCCH:

PROGram
[CSS:FBCCH:PROGram]
Programs the F-BCCH slots in the superframe with the data constructed by the **CSS:FBCCH:BUILD** command.

FC *n*
[CSS:FBCCH:FC]
Enables (*n* = 1) or disables (*n* = 0) Fast Broadcast Control Channel Change Flag.

FC?
Returns current state of FC.

EC *n*
[CSS:FBCCH:EC]
Enables (*n* = 1) or disables (*n* = 0) Extended Broadcast Control Channel Change Flag.

EC?
Returns current state of EC.

MSGtype:

STRUCTure *n*
[CSS:FBCCH:MSGtype:STRUCTure]
Enables (*n* = 1) or disables (*n* = 0) DCCH Structure.

This message shall always be sent first.

ACCess *n*
[CSS:FBCCH:MSGtype:ACCess]
Enables (*n* = 1) or disables (*n* = 0) Access Parameters.

SELection *n*
[CSS:FBCCH:MSGtype:SELection]
Enables (*n* = 1) or disables (*n* = 0) Control Channel Selection Parameters.

REGistration *n*
[CSS:FBCCH:MSGtype:REGistration]
Enables (*n* = 1) or disables (*n* = 0) Registration parameters.

SYSID *n*
[CSS:FBCCH:MSGtype:SYSID]
Enables (*n* = 1) or disables (*n* = 0) System Identification.

OLC *n*
[CSS:FBCCH:MSGtype:OLC]
Enables (*n* = 1) or disables (*n* = 0) Overload Class.

Used to regulate originations and registrations on the RACH.

CSS:

FBCCH:

MSGtype:

MACA *n*
[CSS:FBCCH:MSGtype:MACA]
Enables (*n* = 1) or disables (*n* = 0) Mobile Assisted Channel Allocation.

When enabled, orders the mobile station to report radio measurements on certain channels. Order consists of instructions regarding the channels the mobile station shall measure and when to report the measurements for the Mobile Assisted Channel Allocation.

BSMC *n*
[CSS:FBCCH:MSGtype:BSMC]
Enables (*n* = 1) or disables (*n* = 0) Base Station Manufacture Code.

SERVice *n*
[CSS:FBCCH:MSGtype:SERVice]
Enables (*n* = 1) or disables (*n* = 0) Service Menu.

Provides a list of services supported by the BMI.

SOC_BSMC *n*
[CSS:FBCCH:MSGtype:SOC_BSMC]
Enables (*n* = 1) or disables (*n* = 0) System Operator Code/Base Station Manufacture Code.

SOC and BSMC value associated with the BMI.

SOC *n*
[CSS:FBCCH:MSGtype:SOC]
Enables (*n* = 1) or disables (*n* = 0) Message Delivery.

NUMber:

FBCCH *n*
[CSS:FBCCH:NUMber:FBCCH]
Specifies Number of FBCCH. Range of *n* is 0 to 7.

FBCCH?
Returns current value of FBCCH.

EBCCH *n*
[CSS:FBCCH:NUMber:EBCCH]
Specifies Number of EBCCH. Range of *n* is 0 to 7.

EBCCH?
Returns current value of FBCCH.

CSS:

FBCCH:

NUMber:

SBCCH *n*
[CSS:FBCCH:NUMber:SBCCH]
Specifies Number of SBCCH. Range of *n* is 0 to 15.

SBCCH?
Returns current value of SBCCH.

REServed *n*
[CSS:FBCCH:NUMber:REServed]
Specifies Number of Reserved Slots. Range of *n* is 0 to 7.

REServed?
Returns current value of REServed.

NON_PCH *n*
[CSS:FBCCH:NUMber:NON_PCH]
Specifies Number of Non-Paging Channel Subchannel Slots. Range of *n* is 0 to 3.

NON_PCH?
Returns current value of NON_PCH.

HYPERframe *n*
[CSS:FBCCH:HYPERframe]
Specifies Hyperframe Counter. Range of *n* is 0 to 15.

HYPERframe?
Returns current value of HYPERframe.

EXTended *n*
[CSS:FBCCH:EXTended]
Specifies Extended Hyperframe Counter. Range of *n* is 0 to 7.

EXTended?
Returns current value of EXTended.

SUPERframe *n*
[CSS:FBCCH:SUPERframe]
Enables (*n* = 1) or disables (*n* = 0) Primary Superframe Indicator.

SUPERframe?
Returns current state of SUPERframe.

CSS:

FBCCH:

CONfiguration *n*
[CSS:FBCCH:CONfiguration]
Specifies Slot Configuration. Range of *n* is 0 to 3.

CONfiguration?
Returns current value of CONFiguration.

DVCC *n*
[CSS:FBCCH:DVCC]
Specifies Digital Verification Color Code. Range of *n* is 0 to 255.

DVCC?
Returns current value of DVCC.

PFC *n*
[CSS:FBCCH:PFC]
Specifies MAX_SUPPORTED_PFC. Range of *n* is 0 to 7.

Maximum paging frame class supported by a DCCH or a mobile station

PFC?
Returns current value of PFC.

PCH *n*
[CSS:FBCCH:PCH]
Specifies PCH_DISPLACEMENT (Paging Channel Displacement). Range of *n* is 0 to 7.

Number of additional SPACH Slots the mobile station reads when PCON (page continuation) is enabled.

PCH?
Returns the value of PCH.

PFM *n*
[CSS:FBCCH:PFM]
Enables (*n* = 1) or disables (*n* = 0) PFM_DIRECTION.

Paging Frame Modifier defines whether the Layer 2 PFM flag is a pull-in (reduce the Paging Frame Class by one) or a push-out (increment by one) flag.

PFM?
Returns current state of PFM.

CSS:

FBCCH:

CBN:

HIGH *n*
[CSS:FBCCH:CBN:HIGH]
Specifies CBN_High. Range of *n* is 0 to #hFFFF.

Contains information to support message encryption on the forward and reverse DCCH and DTC.

HIGH?
Returns the value of HIGH.

NONPublic:

PROBability:

Can be used to determine if each channel probability block for a given system configuration contains a DCCH for a non-public system in the current service area.

LENGth *n*
[CSS:FBCCH:NONPublic:PROBability:LENGth]
Specifies Non-Public Map Length. Range of *n* is 0 to 15.

LENGth?
Returns current value of LENGth.

BLOCK *n*
[CSS:FBCCH:NONPublic:PROBability:BLOCK]
Specifies Non-Public Block Map. Range of *n* is 0 to #hFFFF.

BLOCk?
Returns current value of BLOCk.

REGistration:

CONTrol *n*
[CSS:FBCCH:NONPublic:REGistration:CONTrol]
Specifies Non-Public Registration Control. Range of *n* is 0 to 3.

CONTrol?
Returns current value of CONTrol.

CSS:

FBCCH:

AUTH *n*
[CSS:FBCCH:AUTH]
Enables (*n* = 1) or disables (*n* = 0) AUTH.

When enabled, mobile station sends the Authentication message along with a Registration, Origination, Page Response or SPACH Confirmation message due to SPACH Notification indicating R-DATA.

AUTH?
Returns current state of AUTH.

S *n*
[CSS:FBCCH:S]
Enables (*n* = 1) or disables (*n* = 0) Serial number.

When enabled, the mobile station sends the Serial Number message along with a Registration, Origination, Page Response or SPACH Confirmation message due to SPACH Notification indicating R-DATA, Base Station Challenge Order or Unique Challenge Order Confirmation.

S?
Returns current state of S.

RAND *n*
[CSS:FBCCH:RAND]
Specifies RAND. Range of *n* is 0 to #hFFFFFFFF.

Random number stored by a mobile station is used for selected authentication processes.

RAND?
Returns current value of RAND.

ACCess:

MS_PWR *n*
[CSS:FBCCH:ACCess:MS_PWR]
Specifies MS_ACC_PWR (Mobile Station Analog Control Channel Power). Range of *n* is 0 to 15.

Maximum nominal output power that the mobile station shall use when accessing the BMI (Base Station, MSC and Interworking Function). MS_ACC_PWR is also used when determining criteria for control channel selection and reselection.

MS_PWR?
Returns current value of MS_PWR.

CSS:

FBCCH:

ACCess:

RSS_MIN *n*
[CSS:FBCCH:ACCess:RSS_MIN]
RSS_ACC_MIN (Received Signal Strength Analog Control Channel Minimum). Range of *n* is 0 to 31.

Used for the cell (re)selection process. RSS_ACC_MIN is the minimum received signal strength required to access the cell.

RSS_MIN?
Returns current value of RSS_MIN.

BURSTsize *n*
[CSS:FBCCH:ACCess:BURSTsize]
Enables (*n* = 1) or disables (*n* = 0) Access Burst Size.

Informs the mobile station of which burst size to use on the RACH (Random Access Control Channel).

BURSTsize?
Returns current state of BURSTsize.

MAX:

RETries *n*
[CSS:FBCCH:MAX:RETries]
Specifies Max Retries. Range of *n* is 0 to 7.

Maximum number of access attempts that Layer 2 can make before declaring the access to have failed.

RETries?
Returns current value of RETries.

BUSY *n*
[CSS:FBCCH:MAX:BUSY]
Enables (*n* = 1) or disables (*n* = 0) Max Busy/Reserved.

Maximum number of times that BRI (Busy Reserved Idle) ≠ Idle can be detected during any given access attempt before Layer 2 declares an access attempt failure.

BUSY?
Returns current state of BUSY.

CSS:

FBCCH:

MAX:

REPetitions *n*
[CSS:FBCCH:MAX:REPititions]
Specifies Max Repetitions. Range of *n* is 0 to 3.

Maximum number of times a specific burst within any given access attempt may be sent to the RACH before Layer 2 declares an access attempt failure.

REPetitions?
Returns current value of REPetitions.

STOP *n*
[CSS:FBCCH:MAX:STOP]
Enables (*n* = 1) or disables (*n* = 0) Max Stop Counter.

Max Stop Counter identifies the maximum number of times either of the following conditions can be detected for any given access attempt before Layer 2 declares an access attempt failure:

- **BRI set to Reserved or Idle after sending an intermediate burst of an access attempt.**
- **R/N set to Not Received along with BRI set to Reserved or Idle after sending the last burst of an access attempt.**

STOP?
Returns current value of STOP.

RDATA:

LENGth *n*
[CSS:FBCCH:RDATA:LENGth]
Specifies R-DATA Message Length. Range of *n* is 0 to 7.

LENGth?
Returns current value of LENGth.

BARred *n*
[CSS:FBCCH:BARred]
Specifies Cell Barred. Range of *n* is (0-31).

BARred?
Returns current value of BARred.

CSS:

FBCCH:

SUBaddressing *n*
[CSS:FBCCH:SUBaddressing]
Enables (*n* = 1) or disables (*n* = 0) Subaddressing Support.

SUBaddressing?
Returns current state of SUBaddressing.

DIC *n*
[CSS:FBCCH:DIC]
Enables (*n* = 1) or disables (*n* = 0) Delay Interval Compensation Mode.

Controls the DIC mode application in the mobile station. When received in the access parameters message, the domain of DIC application shall be the DCCH. When received in the Digital Traffic Channel Designation message, the domain of DIC application shall be the DTC.

DIC?
Returns current value of DIC.

SS_SUFF *n*
[CSS:FBCCH:SS_SUFF]
Specifies Signal Strength Sufficient. Range of *n* is 0 to 31.

Minimum Signal Strength Sufficient for a candidate control channel to be considered for control channel reselection. SS_SUFF is used, in some instances, to control cell reselection using an absolute threshold.

SS_SUFF?
Returns current value of SS_SUFF.

SCAN:

INTerval *n*
[CSS:FBCCH:SCAN:INTerval]
Specifies SCANINTERVAL. Range of *n* is 0 to 15.

Basic interval, in Hyperframes, between consecutive signal strength measurements. The basic interval is set to 1 plus the value of the field.

INTerval?
Returns current value of INTerval.

CSS:

FBCCH:

SCAN:

OPTION *n*
[CSS:FBCCH:SCAN:OPTION]
Enables (*n* = 1) or disables (*n* = 0) Scanning Option Indicator.

When enabled, mobile station can apply the optional enhancements to the signal strength measurement interval applicable to NL entries.

OPTION?
Returns current state of OPTION.

INITial *n*
[CSS:FBCCH:INITial]
Enables (*n* = 1) or disables (*n* = 0) Initial Selection Control.

Used to discourage a mobile station executing the Control Channel Selection procedure (initial selection) from selecting a DCCH for camping purposes.

INITial?
Returns current state of INITial.

DELay *n*
[CSS:FBCCH:DELay]
Specifies Delay. Range of *n* is 0 to 15.

Used for Control Channel reselection purposes.

DELay?
Returns current value of DELay.

ADDitional:

NUMber *n*
[CSS:FBCCH:ADDitional:NUMber]
Specifies Number of additional DCCH Channels. Range of *n* is 0 to 7.

NUMber?
Returns current value of NUMber.

DCCH:

CHANnel *n,m*
[CSS:FBCCH:ADDitional:DCCH:CHANnel]
Specifies DCCH Channel (*m*) indexed by *n*. Range of *n* is 0 to 7; range of *m* is 0 to 2047.

CHANnel? *n*
Returns current value of CHANnel indexed by *n*. Range of *n* is 0 to 7.

CSS:

FBCCH:

ADDitional:

DCCH:

SLOT *n,m*
[CSS:FBCCH:ADDitional:DCCH:SLOT]
Specifies Slot Configuration (*m*) indexed by *n*. Range of *n* is 0 to 7; range of *m* is 0 to 3.

Indicates the number of slots assigned to DCCHs on the channel in the DCCH Channel field.

SLOT? *n*
Returns current value of SLOT indexed by *n*. Range of *n* is 0 to 7.

REGH *n*
[CSS:FBCCH:REGH]
Enables (*n* = 1) or disables (*n* = 0) REGH.

When enabled, a home mobile station is allowed to register.

REGH?
Returns current state of REGH.

REGR *n*
[CSS:FBCCH:REGR]
Enables (*n* = 1) or disables (*n* = 0) REGR.

When enabled, a roaming mobile station is allowed to register.

REGR?
Returns current state of REGR.

PUREG *n*
[CSS:FBCCH:PUREG]
Enables (*n* = 1) or disables (*n* = 0) Power Up Registration.

PUREG?
Returns current state of PUREG.

PDREG *n*
[CSS:FBCCH:PDREG]
Enables (*n* = 1) or disables (*n* = 0) Power Down Registration.

PDREG?
Returns current state of PDREG.

CSS:

FBCCH:

SYREG *n*
[CSS:FBCCH:SYREG]
Enables (*n* = 1) or disables (*n* = 0) SYREG.
When enabled, a mobile station is to register when it enters a new system identification area.

SYREG?
Returns current state of SYREG.

LAREG *n*
[CSS:FBCCH:LAREG]
Enables (*n* = 1) or disables (*n* = 0) LAREG.
When enabled, a mobile station is to register when the registration number of the current DCCH is not a part of its registration number list used to define its location area.

LAREG?
Returns current state of LAREG.

DEREG *n*
[CSS:FBCCH:DEREG]
Enables (*n* = 1) or disables (*n* = 0) De-Registration.

DEREG?
Returns current state of DEREG.

FOREG *n*
[CSS:FBCCH:FOREG]
Enables (*n* = 1) or disables (*n* = 0) forced registration.

FOREG?
Returns current state of FOREG.

CAPability *n*
[CSS:FBCCH:CAPability]
Enables (*n* = 1) or disables (*n* = 0) Capability Request.
When enabled, a mobile station shall include a Capability Report message when sending a New System registration, Forced registration or Power-Up registration.

CAPability?
Returns current state of CAPability.

CSS:

FBCCH:

RNUM *n*
[CSS:FBCCH:RNUM]
Specifies Present RNUM. Range of *n* is 0 to 1023.

Registration number that is used to define a particular mobile station's VMLA (Virtual Mobile Location Area).

RNUM?
Returns current value of RNUM.

REGPER *n*
[CSS:FBCCH:REGPER]
Specifies REG Period. Range of *n* is 0 to 511.

Identifies the registration periodically in number of 94 superframes. REGPER values are coded from 0 to 511 indicating 94 to 48128 superframes (approximately 1 minute to 8.5 hours).

REGPER?
Returns current value of REGPER.

REGID:

ID *n*
[CSS:FBCCH:REGID:ID]
Specifies System clock. Range of *n* is 0 to #hFFFFF.

ID?
Returns current value of ID.

PER *n*
[CSS:FBCCH:REGID:PER]
Specifies how often ID is incremented. Range of *n* is 0 to 15.

PER?
Returns current value of PER.

SID *n*
[CSS:FBCCH:SID]
Specifies System Identification. Range of *n* is 0 to 32767.

Digital identification associated with a cellular system where each system is assigned a unique number.

SID?
Returns current value of SID.

CSS:

FBCCH:

NETwork *n*
[CSS:FBCCH:NETwork]
Specifies which Network Types are supported on a control channel. Range of *n* is 0 to 7.

NETwork?
Returns current value of NETwork.

PROTocol *n*
[CSS:FBCCH:PROTocol]
Specifies Protocol Version supported. Range of *n* is 0 to 15.

PROTocol?
Returns current value of PROTocol

PSID_RSID:

Private/Residential System Identification.

SOC *n*
[CSS:FBCCH:PSID_RSID:SOC]
Specifies System Operator Code. Range of *n* is 0 to 4095.

Identifies which operator is providing service. If the mobile station receives a reserved SOC value, the value shall be considered an unknown system operator code.

SOC?
Returns current value of SOC.

NUMber *n*
[CSS:FBCCH:PSID_RSID:NUMber]
Specifies Number of PSID/RSID. Range of *n* is 0 to 15.

NUMber?
Returns current value of NUMber.

TYPE *n,m*
[CSS:FBCCH:PSID_RSID:TYPE]
Enables (*m* = 1) or disables (*m* = 0) PSID/RSID Type Indicator indexed by *n*. Range of *n* is 0 to 15.

TYPE? *n*
Returns current state of TYPE indexed by *n*. Range of *n* is 0 to 15.

CSS:

FBCCH:

PSID_RSID:

VALUE *n,m*
[CSS:FBCCH:PSID_RSID:VALUE]
Specifies PSID/RSID Value (*m*) indexed by *n*. Range of *n* is 0 to 15; range of *m* is 0 to #hFFFF.

VALUE? *n*
Returns current value of VALUE indexed by *n*. Range of *n* is 0 to 15.

COUNTRY:

CODE *n*
[CSS:FBCCH:COUNTRY:CODE]
Specifies Mobile Country Code of the current DCCH. Range of *n* is 0 to 1023.

CODE?
Returns current value of CODE.

ALPHA:

SID "*n*"
[CSS:FBCCH:ALPHA:SID]
Specifies Alphanumeric System Identification. *n* is ASCII string.

SID?
Returns current value of SID.

BSMC *n*
[CSS:FBCCH:BSMC]
Specifies assigned manufacturers code (Base Station Manufacture Code). Range of *n* is 0 to 255.

The value of 0 is reserved and shall be considered an unknown base station manufacturer code by the receiving mobile station.

BSMC?
Returns current value of BSMC.

CUSTOM:

LENGth *n*
[CSS:FBCCH:CUSTOM:LENGth]
Specifies Length. Range of *n* is 1 to 64.

LENGth?
Returns current value of LENGth.

CSS:

FBCCH:

CUSTOM:

CONTrol *n,m*
[CSS:FBCCH:CUSTOM:CONTrol]
Specifies Custom Control (*m*) indexed by *n*. Range of *n* is 0 to 63; range of *m* is 0 to 255.

CONTrol?
Returns current value of CONTrol.

MACA:

Mobile Assisted Channel Allocation.

STATus *n*
[CSS:FBCCH:MACA:STATus]
Specifies which MACA function combinations are enabled (MACA_STATUS). Range of *n* is 0-3.

STATus?
Returns current value of STATus.

TYPE *n*
[CSS:FBCCH:MACA:TYPE]
Specifies when MACA reporting is to take place (MACA_TYPE). Range of *n* is 0 to 15.

TYPE?
Returns current value of TYPE.

EIGHT:

CONTrol *n*
[CSS:FBCCH:MACA:EIGHT:CONTrol]
Enables (*n* = 1) or disables (*n* = 0) MACA_8_CONTROL.

This information element, together with MACA_TYPE and MACA_STATUS, determines the number of channels reported.

CONTrol?
Returns current value of CONTrol.

CSS:

FBCCH:

MACA:

LIST:

Contains information regarding the channels, other than the current DCCH, the mobile station shall measure for mobile assisted channel allocation.

NUMber *n*
[CSS:FBCCH:MACA:LIST:NUMber]
Specifies Number of MACA Channels. Range of *n* is 0 to 15.

NUMber?
Returns current value of NUMber.

CHAN *n,m*
[CSS:FBCCH:MACA:LIST:CHAN]
Specifies Channel (*m*) indexed by *n*. Range of *n* is 0 to 15; range of *m* is 0 to 2047.

CHAN? *n*
Returns current value of CHAN indexed by *n*. Range of *n* is 0 to 15.

OTHER:

Contains information regarding the channels other than the current DCCH. A mobile station that is not capable of performing measurements on a channel specified in this list shall report a RSS value of 00000 for that channel.

HYPERband *n*
[CSS:FBCCH:MACA:LIST:OTHER:HYPERband]
Specifies Frequency band information. Range of *n* is 0 to 3.

HYPERband?
Returns current value of HYPERband.

NUMber *n*
[CSS:FBCCH:MACA:LIST:OTHER:NUMber]
Specifies Number of MACA Channels. Range of *n* is 0 to 15.

NUMber?
Returns current value of NUMber.

CHAN *n,m*
[CSS:FBCCH:MACA:LIST:OTHER:CHAN]
Specifies Channel (*m*) indexed by *n*. Range of *n* is 0 to 15; range of *m* is 0 to 2047.

CHAN? *n*
Returns current value of CHAN indexed by *n*. Range of *n* is 0 to 15.

CSS:

FBCCH:

OLC *n*
[CSS:FBCCH:OLC]
Specifies Overload Class. Range of *n* is 0 to #hFFFF.

Determines whether or not a mobile station can make an Origination, Registration or Originated Point-to-Point Teleservice. The mobile station must examine the value of the OLC bit map corresponding to internally stored access overload class assignment. If the bit in the OLC bit map is enabled, the mobile station shall continue with its access attempt. Otherwise, it shall not make an access attempt.

OLC?
Returns current value of OLC.

MAP:

VPM *n*
[CSS:FBCCH:MAP:VPM]
Specifies the forms of voice privacy supported by the BMI (Voice Privacy Mode Map). Range of *n* is 0 to 15.

VPM?
Returns current value of VPM.

DPM *n*
[CSS:FBCCH:MAP:DPM]
Specifies the forms of data privacy supported by the BMI (Data Privacy Mode Map). Range of *n* is 0 to 15.

DPM?
Returns current value of DPM.

CODER *n*
[CSS:FBCCH:MAP:CODER]
Specifies the types of voice coders supported by the BMI (Voice Coder Map). Range of *n* is 0 to 63.

CODER?
Returns current value of CODER.

CSS:

FBCCH:

MAP:

MEA:

Message Encryption Algorithm Map - Identifies the message encryption algorithms, domain and keys supported by a DCCH or a DTC.

DOMAIN ***n***
[CSS:FBCCH:MAP:MEA:DOMAIN]
Specifies the number of instances and ordering of the Encryption Algorithms field (Domain Map). Range of *n* is 0 to 255.

DOMAIN?
Returns current value of DOMAIN.

ALGORithms ***n,m***
[CSS:FBCCH:MAP:MEA:ALGORithms]
Specifies Encryption Algorithms (*m*) indexed by *n*. Range of *n* is 0 to 7; range of *m* is 0 to 15.

ALGORithms? ***n***
Returns current value of ALGORithms indexed by *n*. Range of *n* is 0 to 7.

MEK ***n***
[CSS:FBCCH:MAP:MEK]
Specifies the message encryption keys supported by the BMI (Message Encryption Key Map). Range of *n* is 0 to 15.

MEK?
Returns current value of MEK.

MENU ***n***
[CSS:FBCCH:MAP:MENU]
Specifies the services supported by the BMI (Menu Map). Range of *n* is 0 to #h3FF.

MENU?
Returns current value of MENU.

ARQ ***n***
[CSS:FBCCH:MAP:ARQ]
Enables (*n* = 1) or disables (*n* = 0) FACCH/SACCH ARQ (Automatic Retransmission Request) Map.

When enabled, the BMI supports FACCH/SACCH ARQ mode on the IS-136 digital traffic channels.

ARQ?
Returns current state of ARQ.

CSS:

FBCCH:

MAP:

USER *n*
[CSS:FBCCH:MAP:USER]
Enables (*n* = 1) or disables (*n* = 0) User Group Map.

When enabled, the BMI supports User Group operation.

USER?
Returns current state of USER.

SMS *n*
[CSS:FBCCH:MAP:SMS]
Specifies the extent to which the BMI supports the CMT teleservice (Short Message Service Map). Range of *n* is 0 to 3.

SMS?
Returns current value of SMS.

IRA *n*
[CSS:FBCCH:IRA]
Enables (*n* = 1) or disables (*n* = 0) International Reference Alphabet.

When enabled, mobile station or BMI supports IRA address encoding in the Address Info information element.

IRA?
Returns current state of IRA.

SOC *n*
[CSS:FBCCH:SOC]
Specifies which operator is providing service (System Operator Code). Range of *n* is 0 to 4095.

A received reserved SOC value shall be considered an unknown system operator code by the receiving mobile station.

SOC?
Returns current value of SOC.

ENABLE:

EXTENDED *n*
[CSS:FBCCH:ENABLE:EXTENDED]
Enables (*n* = 1) or disables (*n* = 0) Extended Hyperframe Counter.

EXTENDED?
Returns current state of EXTENDED.

CSS:

FBCCH:

ENABLE:

CBN:

HIGH *n*
[CSS:FBCCH:ENABLE:CBN:HIGH]
Enables (*n* = 1) or disables (*n* = 0) CBN_High.

HIGH?
Returns current state of HIGH.

NONPublic:

PROBability *n*
[CSS:FBCCH:ENABLE:NONPublic:PROBability]
Enables (*n* = 1) or disables (*n* = 0) Non-Public Probability Blocks.

PROBability?
Returns current state of PROBability.

REGistration *n*
[CSS:FBCCH:ENABLE:NONPublic:REGistration]
Enables (*n* = 1) or disables (*n* = 0) Non-Public Registration Control.

REGistration?
Returns current state of REGistration.

ADDitional:

DCCH *n*
[CSS:FBCCH:ENABLE:ADDitional:DCCH]
Enables (*n* = 1) or disables (*n* = 0) Additional DCCH information.

DCCH?
Returns current state of DCCH.

RNUM *n*
[CSS:FBCCH:ENABLE:RNUM]
Enables (*n* = 1) or disables (*n* = 0) Present Registration Number.

RNUM?
Returns current state of RNUM.

REGPER *n*
[CSS:FBCCH:ENABLE:REGPER]
Enables (*n* = 1) or disables (*n* = 0) REG Period.

REGPER?
Returns current state of REGPER.

CSS:

FBCCH:

ENABLE:

REGID *n*
[CSS:FBCCH:ENABLE:REGID]
Enables (*n* = 1) or disables (*n* = 0) REGID Parameters.

REGID?
Returns current state of REGID.

PSID_RSID *n*
[CSS:FBCCH:ENABLE:PSID_RSID]
Enables (*n* = 1) or disables (*n* = 0) Private/Residential System Identification.

PSID_RSID?
Returns current state of PSID_RSID.

COUNTRY:

CODE *n*
[CSS:FBCCH:ENABLE:COUNTRY:CODE]
Enables (*n* = 1) or disables (*n* = 0) Mobile Country Code.

CODE?
Returns current state of CODE.

ALPHA:

SID *n*
[CSS:FBCCH:ENABLE:ALPHA:SID]
Enables (*n* = 1) or disables (*n* = 0) Alphanumeric System Identification.

SID?
Returns current state of SID.

MACA:

Mobile Assisted Channel Allocation.

EIGHT:

CONTrol *n*
[CSS:FBCCH:ENABLE:MACA:EIGHT:CONTrol]
Enables (*n* = 1) or disables (*n* = 0) MACA_8_CONTROL.

CONTrol?
Returns current state of CONTrol.

CSS:

FBCCH:

ENABLE:

MACA:

LIST *n*
[CSS:FBCCH:ENABLE:MACA:LIST]
Enables (*n* = 1) or disables (*n* = 0) MACA_LIST.

LIST?
Returns current state of LIST.

LIST:

OTHER *n*
[CSS:FBCCH:ENABLE:MACA:LIST:OTHER]
Enables (*n* = 1) or disables (*n* = 0) MACA_LIST (Other Hyperband).

OTHER?
Returns current state of OTHER.

7-16-4 E-BCCH COMMANDS

CSS:

EBCCH:

BUILD
[CSS:EBCCH:BUILD]
This command builds the data that makes up the E-BCCH.

Before executing this command, the message types and data fields that make up the E-BCCH should be programmed. This command takes that information and generates all of the slots in the E-BCCH cycle. Since an E-BCCH cycle may continue through several superframes, the length of the E-BCCH built may be longer than Number of E-BCCH allocated to a superframe. Therefore, each time after the E-BCCH slots have been sent, the **CSS:EBCCH:PROGram *n*** command, defined below, can be used to transfer the appropriate number of slots of data into the superframe. The following, outlines the process involved in building a E-BCCH:

1. Enable the desired message types.
2. Enable the desired optional fields associated with the selected message types.
3. Program the data fields associated with the enabled message types.
4. Build the E-BCCH.
5. Program the superframe with the next section of the E-BCCH cycle.

The data built to make up the E-BCCH can be returned by the CSS:EBCCH:DATA? *n,m* command defined below.

LENGth?
[CSS:EBCCH:LENGth?]
Returns current length of the E-BCCH in slots after the **CSS:EBCCH:BUILD** command has been executed.

This value is useful in programming the ECL Layer 2 field.

DATA? *n,m*
[CSS:EBCCH:DATA?]
Returns the E-BCCH data that has been built.

Returns current 16 bit word (indexed by *m*) in slot (*n*). Range of *n* is 0 to 255; range of *m* is 0 to 6.

Each slot consists of 109 bits of data. The 16 most significant bits of the data are returned when *m* = 0, and the 13 least significant bits of data are returned when *m* = 6. All returned data is left justified.

This data format correlates with the data format used in the CSS:FDCCH:SUPER:DATA n,x,word command.

CSS:

EBCCH:

PROGram *dest,source,length*
[CSS:EBCCH:PROGram]
Programs the slots in the superframe with the data constructed by **CSS:EBCCH:BUILD** command. *dest* is the location in the superframe; *source* is the start location in the EBCCH buffer; *length* is the number of frames of data moved from the EBCCH buffer to the superframe. Range of *dest* is 0 to 31; range of *source* is 0 to 255; range of *length* is 0 to 8.

ECL *n*
[CSS:EBCCH:ECL]
Specifies the total number of Layer 2 frames required for the current E-BCCH Cycle (Extended Broadcast Control Channel Cycle Length). Range of *n* is 0 to 255.

ECL?
Returns current value of ECL.

MSGtype:

MACA *n*
[CSS:EBCCH:MSGtype:MACA]
Enables (*n* = 1) or disables (*n* = 0) Mobile Assisted Channel Allocation.

Used to order the mobile station to report radio measurements on certain channels. It contains information regarding the channels the mobile station shall measure and when to report the measurements for the mobile assisted channel allocation.

NEIGHbor:

CELL *n*
[CSS:EBCCH:MSGtype:NEIGHbor:CELL]
Enables (*n* = 1) or disables (*n* = 0) Neighbor Cell.

SERVice *n*
[CSS:EBCCH:MSGtype:NEIGHbor:SERVice]
Enables (*n* = 1) or disables (*n* = 0) Neighbor Service Info.

Provides information regarding services support by a TDMA Neighbor.

RCI *n*
CSS:EBCCH:MSGtype:RCI
Enables (*n* = 1) or disables (*n* = 0) Regulatory Configuration.

BSMC *n*
[CSS:EBCCH:MSGtype:BSMC]
Enables (*n* = 1) or disables (*n* = 0) Base Station Manufacture Code Message Delivery.

Used to carry BSMC specific signaling information, the content of which is beyond the scope of this specification.

CSS:

EBCCH:

MSGtype:

EMERGency *n*
[CSS:EBCCH:MSGtype:EMERGency]
Enables (*n* = 1) or disables (*n* = 0) Emergency Information Broadcast.

Provides emergency information to all mobile stations.

SERVice *n*
[CSS:EBCCH:MSGtype:SERVice]
Enables (*n* = 1) or disables (*n* = 0) Service Menu.

Provides a list of services supported by the BMI.

SOC_BSMC *n*
[CSS:EBCCH:MSGtype:SOC_BSMC]
Enables (*n* = 1) or disables (*n* = 0) System Operator Code/Base Station Manufacture Code Message Delivery.

Used to identify the SOC and BSMC value associated with the BMI.

SOC *n*
[CSS:EBCCH:MSGtype:SOC]
Enables (*n* = 1) or disables (*n* = 0) System Operator Code Message Delivery.

Used to carry SOC specific signaling information, the content of which is beyond the scope of this specification.

TIME *n*
[CSS:EBCCH:MSGtype:TIME]
Enables (*n* = 1) or disables (*n* = 0) Time and Date.

ALTrci *n*
[CSS:EBCCH:MSGtype:ALTrci]
Enables (*n* = 1) or disables (*n* = 0) Alternate Regulatory Configuration Information.

Provides information regarding a DCCH associated with a regulatory configuration different from that of the current DCCH.

SERV_SS n
[CSS:EBCCH:SERV_SS]
Specifies SERV_SS. Range of *n* is 0 to 15.

Used in the control channel reselection process.

SERV_SS?
Returns current value of SERV_SS.

CSS:

EBCCH:

NONPublic:

LENGth *n*
[CSS:EBCCH:NONPublic:LENGth]
Specifies Non-Public Map Length. Range of *n* is 0 to 15.

LENGth?
Returns current value of LENGth.

BLOCK *n*
[CSS:EBCCH:NONPublic:BLOCK]
Specifies Non-Public Block Map. Range of *n* is 0 to #hFFFF.

BLOCK?
Returns current value of BLOCK.

NEIGHbor:

TDMA:

Time Division Multiplex Access. This optional information element specifies the digital channels the mobile station shall measure with regard to the Control Channel Selection and Reselection procedures.

NUMber *n*
[CSS:EBCCH:NEIGHbor:TDMA:NUMber]
Specifies Number of TDMA Neighbor Cells. Range of *n* is 0 to 31.

NUMber?
Returns current value of NUMber.

CELL:

Provides neighbor DCCH specific information.

CHAN *n,m*
[CSS:EBCCH:NEIGHbor:TDMA:CELL:CHAN]
Specifies CHAN (m) select by index (*n*). Range of *n* is 0 to 31; range of *m* is 0 to 2047.

CHAN? *n*
Returns current value of CHAN indexed by n, Range of *n* is 0 to 31.

CSS:

EBCCH:

NEIGHbor:

TDMA:

CELL:

PROTocol *n,m*
[CSS:EBCCH:NEIGHbor:TDMA:CELL:PROTocol]
Specifies Version (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 15.

Identifies the protocol version supported.

PROTocol?
Returns current value of PROTocol indexed by *n*. Range of *n* is 0 to 31.

DVCC *n,m*
[CSS:EBCCH:NEIGHbor:TDMA:CELL:DVCC]
Specifies Digital Verification Color Code (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 255.

DVCC?
Returns current value of DVCC indexed by *n*. Range of *n* is 0 to 31.

OFFset *n,m*
[CSS:EBCCH:NEIGHbor:TDMA:CELL:OFFset]
Specifies RESEL_OFFSET (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 127.

Used to increase/decrease the preference of a new candidate cell being considered for control channel reselection.

OFFset? *n*
Returns current value of RESEL_OFFSET indexed by *n*. Range of *n* is 0 to 31.

SS_SUFF *n,m*
[CSS:EBCCH:NEIGHbor:TDMA:CELL:SS_SUFF]
Specifies SS_SUFF (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 31.

Identifies the minimum signal strength sufficient for a candidate control channel to be considered for control channel reselection. SS_SUFF is used, in some instances, to control cell reselection using an absolute threshold.

SS_SUFF? *n*
Returns current value of SS_SUFF indexed by *n*. Range of *n* is 0 to 31.

CSS:

EBCCH:

NEIGHbor:

TDMA:

CELL:

DELAY *n,m*
[CSS:EBCCH:NEIGHbor:TDMA:CELL:DELAY]
Specifies DELAY (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 15.

Used for Control Channel Reselection purposes.

DELAY? *n*
Returns current value of DELAY indexed by *n*. Range of *n* is 0 to 31.

HL_FREQ *n,m*
[CSS:EBCCH:NEIGHbor:TDMA:CELL:HL_FREQ]
Selects HL_FREQ (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 or 1.

Used to determine the frequency of channel measurements.

HL_FREQ? *n*
Returns current state of HL_FREQ indexed by *n*. Range of *n* is 0 to 31.

SYNC *n,m*
[CSS:EBCCH:NEIGHbor:TDMA:CELL:SYNC]
Selects SYNC (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 or 1.

When enabled, a candidate DCCH is superframe-synchronized with the current DCCH. If this flag is enabled, the maximum time offset between superframes sent on the candidate DCCH and current DCCH shall be no more than 7.5 symbols, i.e., a mobile station shall expect to find synchronization on a candidate DCCH within +/- 7.5 symbols relative to its current DCCH superframe.

SYNC?
Returns current state of SYNC indexed by *n*. Range of *n* is 0 to 31.

CSS:

EBCCH:

NEIGHbor:

TDMA:

CELL:

TYPE:

CELL *n,m*
[CSS:EBCCH:NEIGHbor:TDMA:CELL:TYPE:CELL]
Specifies CELLTYPE (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 3.

Provides a relative distinction by an operator to bias mobile station control channel reselection decisions in order to insure traffic flows according to an operator's desires.

CELL? *n*
Returns current value of CELL indexed by *n*. Range of *n* is 0 to 31.

NETwork *n,m*
[CSS:EBCCH:NEIGHbor:TDMA:CELL:TYPE:NETwork]
Specifies Network Type (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 7.

Identifies which Network Types are supported on a control channel.

NETwork? *n*
Returns current value of NETwork indexed by *n*. Range of *n* is 0 to 31.

RETRY *n,m*
[CSS:EBCCH:NEIGHbor:TDMA:CELL:RETRY]
Enables (*m* = 1) or disables (*m* = 0) Directed Retry Channel indexed by *n*. Range of *n* is 0 to 31.

Used to determine if the neighbor list is to be considered for Directed Retry purposes.

RETRY? *n*
Returns current state of RETRY indexed by *n*. Range of *n* is 0 to 31.

CSS:

EBCCH:

NEIGHbor:

TDMA:

CELL:

ACCess:

MS_PWR *n,m*
[CSS:EBCCH:NEIGHbor:TDMA:CELL:ACCess:MS_PWR]
Specifies MS_ACC_PWR (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 15.

Mobile Station/Analog Control Channel Power. Identifies the maximum nominal output power that the mobile station shall use when accessing the BMI (Base Station , MSC and Interworking Function). MS_ACC_PWR is also used when determining criteria for control channel selection and reselection.

MS_PWR? *n*
Returns current value of MS_PWR indexed by *n*. Range of *n* is 0 to 31.

RSS_MIN *n,m*
[CSS:EBCCH:NEIGHbor:TDMA:CELL:ACCess:RSS_MIN]
Specifies RSS_ACC_MIN (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 31.

Received Signal Strength/Analog Control Channel Minimum. Used for the cell (re)selection process. RSS_ACC_MIN is the minimum received signal strength required to access the cell.

RSS_MIN? *n*
Returns current value of RSS_MIN indexed by *n*. Range of *n* is 0 to 31.

PSID_RSID:

Private/Residential System Identification

INDicator *n,m*
[CSS:EBCCH:NEIGHbor:TDMA:CELL:PSID_RSID:INDicator]
Enables (*m* = 1) or disables (*m* = 0) PSID/RSID Indicator indexed by *n*. Range of *n* is 0 to 31.

Indicates whether or not the PSID/RSID related mapping fields are present for the associated DCCH neighbor. Specifically, if the PSID/RSID Indicator value is set to 1, the PSID/RSID Support Length and PSID/RSID Support fields are present, otherwise, not present.

INDicator? *n*
Returns current state of INDicator indexed by *n*. Range of *n* is 0 to 31.

CSS:

EBCCH:

NEIGHbor:

TDMA:

CELL:

PSID_RSID:

LENGth *n,m*

[CSS:EBCCH:NEIGHbor:TDMA:CELL:PSID_RSID:LENGth]

Specifies PSID/RSID Support Length (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 15.

The PSID/RSID Support Length field is only present when the PSID/RSID Indicator field is set to 1. When present, this field is always 4 bits long and is used to determine the length of the PSID/RSID Support field. The value of PSID/RSID Support Length is 0 if not present.

LENGth? *n*

Returns current value of LENGth indexed by *n*. Range of *n* is 0 to 31.

CSS:

EBCCH:

NEIGHbor:

TDMA:

CELL:

PSID_RSID:

SUPport *n,m*
[CSS:EBCCH:NEIGHbor:TDMA:CELL:PSID_RSID:SUPport]
Specifies PSID/RSID Support (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 1 to #hFFFF.

The PSID/RSID Support field is only present when the PSID/RSID Indicator field is set to 1. When present, the length of this is determined by adding 1 to the value of the PSID/RSID Support Length field. The value of PSID/RSID Support is 0 if not present.

The PSID/RSID Support field indicates which PSID/RSID values identified in the PSID/RSID Set information element of the System Identity message of the current DCCH are supported by the DCCH neighbor under consideration. The ordering of the bits in this field reflects the ordering of the PSID/RSID Set sent in the System Identity message in that the least significant bit is corresponds to the first PSID/RSID listed in the PSID/RSID Set. If a bit in this field is set to 1, then the corresponding PSID/RSID entry in the PSID/RSID Set is supported by the neighbor cell under consideration. On the other hand, if a bit in this field is set to 0, then the corresponding PSID/RSID entry in the PSID/RSID Set is ***not*** supported by the neighbor cell under consideration.

For example, if 16 PSID/RSIDs are listed in the System Identity message and the neighbor cell supports the 1st, 2nd,3rd, and 5th PSID/RSIDs in the PSID/RSID Set, the values of PSID/RSID Support Length and PSID/RSID Support shall be:

PSID/RSID Support length = 0100
PSID/RSID Support = 10111

If 16 PSID/RSIDs are listed in the System Identity message and the neighbor cell supports 3rd and 16th PSID/RSID Set, the values of PSID/RSID Support Length and PSID/RSID Support shall be:

PSID/RSID Support length = 1111
PSID/RSID Support = 1000000000000100

SUPport? *n*
Returns current value of SUPport indexed by *n*. Range of *n* is 0 to 31.

CSS:

EBCCH:

NEIGHbor:

TDMA:

INFO:

Service Info - Provides service attribute information for TDMA neighbors.

COUNt *n*
[CSS:EBCCH:NEIGHbor:TDMA:INFO:COUNt]
Specifies number of TDMA Service Map instances present in the Neighbor Service Info message (TDMA Neighbor Count). Range of *n* is 0 to 31.

Set to 1 plus the value in this field.

COUNt?
Returns current value of COUNt.

SERVice:

Provides service information for one or more TDMA neighbors.

INDicator *n,m*
[CSS:EBCCH:NEIGHbor:TDMA:INFO:SERVice:INDicator]
Enables (*m* = 1) or disables (*m* = 0) Service Map Indicator indexed by *n*. Range of *n* is 0 to 31.

Indicates, when enabled, that Service Map field is present.

INDicator? *n*
Returns current state of INDicator indexed by *n*. Range of *n* is 0 to 31.

MAP *n,m*
[CSS:EBCCH:NEIGHbor:TDMA:INFO:SERVice:MAP]
Specifies Service Map (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 1023.

This field, when present, provides service information.

MAP? *n*
Returns the value of MAP indexed by *n*. Range of *n* is 0 to 31.

CSS:

EBCCH:

NEIGHbor:

ANAlog:

This optional information element contains information regarding the analog channels the mobile station shall measure with regard to the Control Channel Selection and Reselection procedures. This information element always specifies analog neighbors in the 800 MHz hyperband.

NUMber *n*
[CSS:EBCCH:NEIGHbor:ANAlog:NUMber]
Specifies Number of Analog Neighbor Cells. Range of *n* is 0 to 31.

NUMber?
Returns current value of NUMber.

CELL:

CHAN *n,m*
[CSS:EBCCH:NEIGHbor:ANAlog:CELL:CHAN]
Specifies CHAN (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 2047.

CHAN? *n*
Returns current value of CHAN indexed by *n*. Range of *n* is 0 to 31.

PROTocol *n,m*
[CSS:EBCCH:NEIGHbor:ANAlog:CELL:PROTocol]
Specifies Protocol Version (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* in 0 to 15.

Identifies the protocol version supported.

PROTocol? *n*
Returns current value of PROTocol indexed by *n*. Range of *n* is 0 to 31.

DCC *n,m*
[CSS:EBCCH:NEIGHbor:ANAlog:CELL:DCC]
Specifies Digital Color Code (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 3.

DCC? *n*
Returns current value of DCC indexed by *n*. Range of *n* is 0 to 31.

CSS:

EBCCH:

NEIGHbor:

ANAlog:

CELL:

OFFset *n,m*
[CSS:EBCCH:NEIGHbor:ANAlog:CELL:OFFset]
Specifies RESEL_OFFSET (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 127.

Used to increase or decrease the preference of a new candidate cell being considered for control channel reselection.

OFFset? *n*
Returns current value of OFFset indexed by *n*. Range of *n* is 0 to 31.

SS_SUFF *n,m*
[CSS:EBCCH:NEIGHbor:ANAlog:CELL:SS_SUFF]
Specifies SS_SUFF (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 31.

Identifies the minimum signal strength sufficient for a candidate control channel to be considered for control channel reselection. SS_SUFF is used, in some instances, to control cell reselection using an absolute threshold.

SS_SUFF? *n*
Returns current value of SS_SUFF indexed by *n*. Range of *n* is 0 to 31.

DELAY *n,m*
[CSS:EBCCH:NEIGHbor:ANAlog:CELL:DELAY]
Specifies DELAY (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 15.

Used for Control Channel Reselection purposes.

DELAY? *n*
Returns current value of DELAY indexed by *n*. Range of *n* is 0 to 31.

HL_FREQ *n,m*
[CSS:EBCCH:NEIGHbor:ANAlog:CELL:HL_FREQ]
Enables (*m* = 1) or disables (*m* = 0) HL_FREQ indexed by *n*. Range of *n* is 0 to 31.

Used to determine the frequency of channel measurements.

HL_FREQ? *n*
Returns current state of HL_FREQ indexed by *n*. Range of *n* is 0 to 31.

CSS:

EBCCH:

NEIGHbor:

ANAlog:

CELL:

TYPE:

CELL *n,m*
[CSS:EBCCH:NEIGHbor:ANAlog:CELL:TYPE:CELL]
Specifies CELLTYPE (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 3.

Provides a relative distinction by an operator to bias mobile station control channel reselection decisions in order to insure traffic flows according to an operator's desires.

CELL? *n*
Returns current value of CELL indexed by n (0-31).

NETwork *n,m*
[CSS:EBCCH:NEIGHbor:ANAlog:CELL:TYPE:NETwork]
Specifies Network Type (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 7.

Identifies which Network Types are supported on a control channel.

NETwork? *n*
Returns current value of NETwork indexed by *n*. Range of *n* is 0 to 31.

RETRY *n,m*
[CSS:EBCCH:NEIGHbor:ANAlog:CELL:RETRY]
Enables (*m* = 1) or disables (*m* = 0) Directed Retry Channel indexed by *n*. Range of *n* is 0 to 31.

When enabled, the neighbor list is to be considered for Directed Retry purposes.

RETRY? *n*
Returns current state of RETRY indexed by *n*. Range of *n* is 0 to 31.

CSS:

EBCCH:

NEIGHbor:

ANAlog:

CELL:

ACCess:

MS_PWR *n,m*
[CSS:EBCCH:NEIGHbor:ANAlog:CELL:ACCess:MS_PWR]
Specifies MS_ACC_PWR (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 15.

Mobile Station/Analog Control Channel Power. Identifies the maximum nominal output power that the mobile station shall use when accessing the BMI (Base Station, MSC and Interworking Function). MS_ACC_PWR is also used when determining criteria for control channel selection and reselection.

MS_PWR? *n*
Returns current value of MS_PWR indexed by *n*. Range of *n* is 0 to 31.

RSS_MIN *n,m*
[CSS:EBCCH:NEIGHbor:ANAlog:CELL:ACCess:RSS_MIN]
Specifies RSS_ACC_MIN (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 31.

Received Signal Strength/Analog Control Channel Minimum. Used for the cell (re)selection process. RSS_ACC_MIN is the minimum received signal strength required to access the cell.

RSS_MIN? *n*
Returns current value of RSS_MIN indexed by *n*. Range of *n* is 0 to 31.

OTHER:

Other Hyperband - This optional information element contains information specifying the digital channels the mobile station may measure in order to acquire service in an alternate frequency band.

NUMber *n*
[CSS:EBCCH:NEIGHbor:OTHER:NUMber]
Specifies Number of Neighbor Cells. Range of *n* is 0 to 31.

NUMber?
Returns current value of NUMber.

CSS:

EBCCH:

NEIGHbor:

OTHER:

CELL:

Provides neighbor DCCH specific information.

CHAN *n,m*
[CSS:EBCCH:NEIGHbor:OTHER:CELL:CHAN]
Specifies channel (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 2047.

CHAN? *n*
Returns current value of CHAN indexed by *n*. Range of *n* is 0 to 31.

PROTocol *n,m*
[CSS:EBCCH:NEIGHbor:OTHER:CELL:PROTocol]
Specifies Protocol Version (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 15.

Identifies the protocol version supported.

PROTocol?
Returns current value of PROTocol indexed by *n*. Range of *n* is 0 to 31.

DVCC *n,m*
[CSS:EBCCH:NEIGHbor:OTHER:CELL:DVCC]
Specifies Digital Verification Color Code (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 255.

DVCC?
Returns current value of DVCC indexed by *n*. Range of *n* is 0 to 31.

OFFset *n,m*
[CSS:EBCCH:NEIGHbor:OTHER:CELL:OFFset]
Specifies RESEL_OFFSET (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 127.

Used to increase or decrease the preference of a new candidate cell being considered for control channel reselection.

OFFset? *n*
Returns current value of OFFset indexed by *n*. Range of *n* is 0 to 31.

CSS:

EBCCH:

NEIGHbor:

OTHER:

CELL:

SS_SUFF *n,m*
[CSS:EBCCH:NEIGHbor:OTHER:CELL:SS_SUFF]
Specifies SS_SUFF (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 31.

Identifies the minimum signal strength sufficient for a candidate control channel to be considered for control channel reselection. SS_SUFF is used, in some instances, to control cell reselection using an absolute threshold.

SS_SUFF? *n*
Returns current value of SS_SUFF indexed by *n*. Range of *n* is 0 to 31.

DELAY *n,m*
[CSS:EBCCH:NEIGHbor:OTHER:CELL:DELAY]
Specifies DELAY (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 15.

Used for Control Channel Reselection purposes.

DELAY? *n*
Returns current value of DELAY indexed by *n*. Range of *n* is 0 to 31.

HL_FREQ *n,m*
[CSS:EBCCH:NEIGHbor:OTHER:CELL:HL_FREQ]
Enables (*m* = 1) or disables (*m* = 0) HL_FREQ indexed by *n*. Range of *n* is 0 to 31.

Used to determine the frequency of channel measurements.

HL_FREQ? *n*
Returns current state of HL_FREQ indexed by *n*. Range of *n* is 0 to 31.

CSS:

EBCCH:

NEIGHbor:

OTHER:

CELL:

SYNC *n,m*
[CSS:EBCCH:NEIGHbor:OTHER:CELL:SYNC]
Enables (*m* = 1) or disables (*m* = 0) SYNC indexed by *n*. Range of *n* is 0 to 31.

Indicates if a candidate DCCH is superframe-synchronized with the current DCCH. If this flag is enabled, the maximum time offset between superframes sent on the candidate DCCH and current DCCH shall be no more than 7.5 symbols, i.e., a mobile station shall expect to find synchronization on a candidate DCCH within +/- 7.5 symbols relative to its current DCCH superframe.

SYNC?
Returns current state of SYNC indexed by *n*. Range of *n* is 0 to 31.

TYPE:

CELL *n,m*
[CSS:EBCCH:NEIGHbor:OTHER:CELL:TYPE:CELL]
Specifies CELLTYPE (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 3.

Provides a relative distinction by an operator to bias mobile station control channel reselection decisions in order to insure traffic flows according to an operator's desires.

CELL? *n*
Returns current value of CELL indexed by *n*. Range of *n* is 0 to 31.

NETwork *n,m*
[CSS:EBCCH:NEIGHbor:OTHER:CELL:TYPE:NETwork]
Specifies Network Type (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 7.

Identifies which Network Types are supported on a control channel.

NETwork? *n*
Returns current value of NETwork indexed by *n*. Range of *n* is 0 to 31.

CSS:

EBCCH:

NEIGHbor:

OTHER:

CELL:

RETRY *n,m*
[CSS:EBCCH:NEIGHbor:OTHER:CELL:RETRY]
Enables (*m* = 1) or disables (*m* = 0) Directed Retry Channel indexed by *n*. Range of *n* is 0 to 31.

When enabled, the neighbor list is to be considered for Directed Retry purposes.

RETRY? *n*
Returns current state of RETRY indexed by *n*. Range of *n* is 0 to 31.

ACCess:

MS_PWR *n,m*
[CSS:EBCCH:NEIGHbor:OTHER:CELL:ACCess:MS_PWR]
Specifies MS_ACC_PWR (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* 0 to 15.

Mobile Station/Analog Control Channel Power. Identifies the maximum nominal output power that the mobile station shall use when accessing the BMI (Base Station, MSC and Interworking Function). MS_ACC_PWR is also used when determining criteria for control channel selection and reselection.

MS_PWR? *n*
Returns current value of MS_PWR indexed by *n*. Range of *n* is 0 to 31.

RSS_MIN *n,m*
[CSS:EBCCH:NEIGHbor:OTHER:CELL:ACCess:RSS_MIN]
Specifies RSS_ACC_MIN (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 31.

Received Signal Strength/Analog Control Channel Minimum. Used for the cell (re)selection process. RSS_ACC_MIN is the minimum received signal strength required to access the cell.

RSS_MIN? *n*
Returns current value of RSS_MIN indexed by *n*. Range of *n* is 0 to 31.

CSS:

EBCCH:

NEIGHbor:

OTHER:

CELL:

PSID_RSID:

Private/Residential System Identification

INDicator *n,m*
[CSS:EBCCH:NEIGHbor:OTHER:CELL:PSID_RSID:INDicator]
Enables (*m* = 1) or disables (*m* = 0) PSID/RSID Indicator indexed by *n*. Range of *n* is 0 to 31.

Indicates if the PSID/RSID related mapping fields are present for the associated DCCH neighbor. Specifically, if the PSID/RSID Indicator value is set to 1, the PSID/RSID Support Length and PSID/RSID Support fields are present. If the PSID/RSID Indicator value is set to 0, the PSID/RSID Support Length and PSID/RSID Support fields are not present.

INDicator? *n*
Returns current state of INDicator indexed by *n*. Range of *n* is 0 to 31.

LENGth *n,m*
[CSS:EBCCH:NEIGHbor:OTHER:CELL:PSID_RSID:LENGth]
Specifies PSID/RSID Support Length (*m*) indexed by *n*. Range of *n* is 0 to 31; range *m* is 0 to 15 (if present, 0 if not present).

The PSID/RSID Support Length field is only present when the PSID/RSID Indicator field is set to 1. When present, this field is always 4 bits long and is used to determine the length of the PSID/RSID Support field.

LENGth? *n*
Returns current value of LENGth indexed by *n*. Range of *n* is 0 to 31.

CSS:

EBCCH:

NEIGHbor:

OTHER:

CELL:

PSID_RSID:

SUPport *n,m*

[CSS:EBCCH:NEIGHbor:OTHER:CELL:PSID_RSID:SUPport]

Specifies PSID/RSID Support (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 1 to #hFFFF (if present, 0 if not present).

The PSID/RSID Support field is only present when the PSID/RSID Indicator field is set to 1. When present, the length is determined by adding 1 to the value of the PSID/RSID Support Length field.

The PSID/RSID Support field indicates which PSID/RSID values identified in the PSID/RSID Set information element of the System Identity message of the current DCCH are supported by the DCCH neighbor under consideration. The ordering of the bits in this field reflects the ordering of the PSID/RSID Set sent in the System Identity message in that the least significant bit is corresponds with the first PSID/RSID listed in the PSID/RSID Set. If a bit in this field is set to 1, then the corresponding PSID/RSID entry in the PSID/RSID Set is supported by the neighbor cell under consideration. If a bit in this field is set to 0, then the corresponding PSID/RSID entry in the PSID/RSID Set is **not** supported by the neighbor cell under consideration..

For example if 16 PSID/RSIDs are listed in the System Identity message and the neighbor cell supports the 1st, 2nd,3rd, and 5th PSID/RSIDs in the PSID/RSID Set, the values of PSID/RSID Support Length and PSID/RSID Support shall be:

PSID/RSID Support length = 0100
PSID/RSID Support = 10111

If 16 PSID/RSIDs are listed in the System Identity message and the neighbor cell supports 3rd and 16th PSID/RSID Set, the values of PSID/RSID Support Length and PSID/RSID Support shall be:

PSID/RSID Support length = 1111
PSID/RSID Support = 1000000000000100

SUPport? n

[CSS:EBCCH:NEIGHbor:OTHER:CELL:PSID_RSID:SUPport]

Returns current value of SUPport indexed by *n*. Range of *n* is 0 to 31.

CSS:

EBCCH:

NEIGHbor:

OTHER:

INFO:

Service Info - Provides service attribute information for TDMA neighbors.

HYPERband *n*
[CSS:EBCCH:NEIGHbor:OTHER:INFO:HYPERband]
Specifies HYPERband. Range of *n* is 0 to 3.

Provides service attribute information for Other Hyperband TDMA neighbors.

HYPERband?
Returns current value of HYPERband.

COUNt *n*
[CSS:EBCCH:NEIGHbor:OTHER:INFO:COUNt]
Specifies TDMA Neighbor Count. Range of *n* is 0 to 31.

Identifies the number of TDMA Service Map instances present in the Neighbor Service Info message. Set to 1 plus the value in this field..

COUNt?
Returns current value of COUNt.

SERVice:

Provides service information for one or more TDMA neighbors.

INDicator *n,m*
[CSS:EBCCH:NEIGHbor:OTHER:INFO:SERVice:INDicator]
Enables (*m* = 1) or disables (*m* = 0) Service Map Indicator indexed by *n*. Range of *n* is 0 to 31.

When enabled, indicates the Service Map field is present.

INDicator? *n*
Returns current state of INDicator.

MAP *n,m*
[CSS:EBCCH:NEIGHbor:OTHER:INFO:SERVice:MAP]
Specifies Service Map (*m*) indexed by *n*. Range of *n* is 0 to 31; range of *m* is 0 to 1023 (if present, 0 if not present).

This field, when present, provides service information.

MAP? *n*
Returns current value of MAP indexed by *n*. Range of *n* is 0 to 31.

CSS:

EBCCH:

RCI *n*
[CSS:EBCCH:RCI]
Specifies Regulatory Configuration. Range of *n* is 0 to 3.

RCI?
Returns current value of RCI.

CHANnel:

RF Channel Allocation - Indicates an RF channel allocation for system configurations that are not described in this standard.

NUMber *n*
[CSS:EBCCH:CHANnel:NUMber]
Specifies Number of Channel Groups. Range of *n* is 0 to 63.

NUMber?
Returns current value of NUMber.

GROUP:

Channel Group is an ordered pair indicating the first/last RF Channel Numbers assigned to the Channel Group.

FIRST *n,m*
[CSS:EBCCH:CHANnel:GROUP:FIRST]
Specifies First Channel (*m*) indexed by *n*. Range of *n* is 0 to 63; range of *m* is 0 to 2047.

FIRST? *n*
Returns current value of FIRST indexed by *n*. Range of *n* is 0 to 63.

LAST *n,m*
[CSS:EBCCH:CHANnel:GROUP:LAST]
Specifies Last Channel (*m*) indexed by *n*. Range of *n* is 0 to 63; range of *m* is 0 to 2047.

LAST? *n*
Returns current value of LAST indexed by *n*. Range of *n* is 0 to 63.

CSS:

EBCCH:

BSMC *n*
[CSS:EBCCH:BSMC]
Specifies Base Station Manufacture Code. Range of *n* is 0 to 255.

Identifies the assigned manufacture code. The BSMC value of 0 is reserved. A reserved BSMC value shall be considered an unknown base station manufacture code by the receiving mobile station.

BSMC?
Returns current value of BSMC.

CUSTOM:

LENGth *n*
[CSS:EBCCH:CUSTOM:LENGth]
Specifies Length of Custom Control in octets. Range of *n* is 1 to 64.

LENGth?
Returns current value of LENGth.

CONTrol *n,m*
[CSS:EBCCH:CUSTOM:CONTrol]
Specifies CONTrol (*m*) indexed by *n*. Range of *n* is 0 to 63; range of *m* is 0 to 255.

CONTrol?
Returns current value of CONTrol.

TEXT:

Text Message Data Unit - Contains the message to be broadcast.

LENGth *n*
[CSS:EBCCH:TEXT:LENGth]
Specifies Length Indicator. Range of *n* is 0 to 255.

LENGth?
Returns current value of LENGth.

ENCoding *n*
[CSS:EBCCH:TEXT:ENCoding]
Specifies Encoding Identifier. Range of *n* is 0 to 31.

ENCoding?
Returns current value of ENCoding.

CSS:

EBCCH:

TEXT:

REServed *n*
[CSS:EBCCH:TEXT:REServed]
Specifies Reserved. Range of *n* is 0 to 7.

REServed?
Returns current value of REServed.

CHARacter *n,m*
[CSS:EBCCH:TEXT:CHARacter]
Specifies Short Message Character (*m*) indexed by *n*. Range of *n* is 0 to 255; range of *m* is 0 to 255.

CHARacter? *n*
Returns current value of CHARacter indexed by *n*. Range of *n* is 0 to 255.

SIGnal:

Conveys alerting information to a mobile station.

PITCH *n*
[CSS:EBCCH:SIGnal:PITCH]
Specifies Signal Pitch. Range of *n* is 0 to 3.

PITCH?
Returns current value of PITCH.

CADence *n*
[CSS:EBCCH:SIGnal:CADence]
Specifies Signal Cadence. Range of *n* is 0 to 63.

CADence?
Returns current value of CADence.

DURation *n*
[CSS:EBCCH:SIGnal:DURation]
Specifies Signal Duration. Range of *n* is 0 to 15.

DURation?
Returns current value of DURation.

CSS:

EBCCH:

MACA:

Mobile Assisted Channel Allocation.

STATus *n*
[CSS:EBCCH:MACA:STATus]
Specifies MACA_STATUS. Range of *n* is 0 to 3.

Determines which MACA function combinations are enabled.

STATus?
Returns current value of STATus.

TYPE *n*
[CSS:EBCCH:MACA:TYPE]
Specifies MACA_TYPE. Range of *n* is 0 to 15.

Determines when MACA reporting is to take place.

TYPE?
Returns current value of TYPE.

EIGHT:

CONTrol *n*
[CSS:EBCCH:MACA:EIGHT:CONTrol]
Enables (*n* = 1) or disables (*n* = 0) CONTrol.

This information element, together with MACA_TYPE and MACA_STATUS, determines the number of channels reported

CONTrol?
Returns current state of CONTrol.

LIST:

Contains information regarding the channels (other than the current DCCH) the mobile station shall measure for mobile assisted channel allocation.

NUMber *n*
[CSS:EBCCH:MACA:LIST:NUMber]
Specifies Number of MACA Channels. Range of *n* is 0 to 15.

NUMber?
Returns current value of NUMber.

CSS:

EBCCH:

MACA:

LIST:

CHAN *n,m*
[CSS:EBCCH:MACA:LIST:CHAN]
Specifies Channel (m) indexed by *n*. Range of *n* is 0 to 15; range of *m* is 0 to 2047.

CHAN? *n*
Returns current value of CHAN indexed by *n*. Range of *n* is 0 to 15.

OTHER:

Contains information regarding the channels other than the current DCCH. A mobile station that is not capable of performing measurements on a channel specified in this list shall report a RSS value of 00000 for that channel.

HYPERband *n*
[CSS:EBCCH:MACA:LIST:OTHER:HYPERband]
Specifies Frequency band information. Range of *n* is 0 to 3.

HYPERband?
Returns current value of HYPERband.

NUMber *n*
[CSS:EBCCH:MACA:LIST:OTHER:NUMber]
Specifies Number of MACA Channels. Range of *n* is 0 to 15.

NUMber?
Returns current value of NUMber.

CHAN *n,m*
[CSS:EBCCH:MACA:LIST:OTHER:CHAN]
Specifies Channel (*m*) indexed by *n*. Range of *n* is 0 to 15; range of *m* is 0 to 2047.

CHAN? *n*
Returns current value of CHAN indexed by *n*. Range of *n* is 0 to 15.

CSS:

EBCCH:

MAP:

VPM *n*
[CSS:EBCCH:MAP:VPM]
Specifies Voice Privacy Mode Map. Range of *n* is 0 to 15.

Identifies the forms of voice privacy supported by the BMI.

VPM?
Returns current value of VPM.

DPM *n*
[CSS:EBCCH:MAP:DPM]
Specifies Data Privacy Mode Map. Range of *n* is 0 to 15.

Identifies the forms of data privacy supported by the BMI.

DPM?
Returns current value of DPM.

CODER *n*
[CSS:EBCCH:MAP:CODER]
Specifies Voice Coder Map. Range of *n* is 0 to 63.

Identifies the types of voice coders supported by the BMI.

CODER?
Returns current value of CODER.

MEA:

Message Encryption Algorithm Map - Identifies the message encryption algorithms, domain and keys supported by a DCCH or a DTC.

DOMAIN *n*
[CSS:EBCCH:MAP:MEA:DOMAIN]
Specifies Domain Map. Range of *n* is 0 to 255.

Identifies the number of instances and ordering of the Encryption Algorithms field.

DOMAIN?
Returns current value of DOMAIN.

ALGORithms *n,m*
[CSS:EBCCH:MAP:MEA:ALGORithms]
Specifies Encryption Algorithms (*m*) indexed by *n*. Range of *n* is 0 to 7; range of *m* is 0 to 15.

ALGORithms? *n*
Returns current value of ALGORithms indexed by *n*. Range of *n* is 0 to 7.

CSS:

EBCCH:

MAP:

MEK *n*
[CSS:EBCCH:MAP:MEK]
Specifies Message Encryption Key Map. Range of *n* is 0 to 15.

Identifies the message encryption keys supported by the BMI

MEK?
Returns current value of MEK.

MENU *n*
[CSS:EBCCH:MAP:MENU]
Specifies Menu Map. Range of *n* is 0 to #h3FF.

Identifies the services supported by the BMI

MENU?
Returns current value of MENU.

ARQ *n*
[CSS:EBCCH:MAP:ARQ]
Enables (*n* = 1) or disables (*n* = 0) FACCH/SACCH ARQ (Automatic Retransmission Request) Map.

Identifies if the BMI supports FACCH/SACCH ARQ mode on its IS-136 digital traffic channels.

ARQ?
Returns current state of ARQ.

USER *n*
[CSS:EBCCH:MAP:USER]
Enables (*n* = 1) or disables (*n* = 0) User Group Map.

Identifies whether or not the BMI supports User Group operation.

USER?
Returns current state of USER.

SMS *n*
[CSS:EBCCH:MAP:SMS]
Specifies Short Message Service Map. Range of *n* is 0 to 3.

Identifies the extent to which the BMI supports the CMT teleservice.

SMS?
Returns current value of SMS.

CSS:

EBCCH:

SOC *n*
[CSS:EBCCH:SOC]
Specifies System Operator Code. Range of *n* is 0 to 4095.

Identifies which operator is providing service. A reserved SOC value shall be considered an unknown system operator code by a receiving mobile station.

SOC?
Returns current value of SOC.

TIME *n*
[CSS:EBCCH:TIME]
Specifies TIME. Range of *n* is 0 to #hFFFFFFFF.

Time from Jan 1, 1980 - This information element is a non-critical sequential time counter in seconds elapsed since January 1, 1980, 00:00 hour, 0 seconds using Greenwich Mean Time as the reference point.

TIME?
Returns current value of TIME.

ZONE:

Time Zone Offset - Used to identify the time zone offset in minutes relative to Greenwich Mean Time (GMT).

DIRection *n*
[CSS:EBCCH:ZONE:DIRection]
Selects Time Zone Offset Direction. Range of *n* is 0 or 1.

Specifies whether to add or subtract minutes from the GMT.

DIRection?
Returns current setting of DIRection.

MINutes *n*
[CSS:EBCCH:ZONE:MINutes]
Specifies Minutes. Range of *n* is 0 to 1023.

Specifies the number of minutes.

MINutes?
Returns currents value of MINutes.

CSS:

EBCCH:

ZONE:

DST *n*
[CSS:EBCCH:ZONE:DST]
Selects Time Zone Offset Daylight Savings Indicator. Range of *n* is 0 or 1.

Indicates Standard or Daylight Savings Time.

DST?
Returns current setting of DST.

CHAN *n*
[CSS:EBCCH:CHAN]
Specifies E-BCCH CHAN. Range of *n* is 0 to 2047.

CHAN?
Returns current value of CHAN.

HYPERband:

INFO *n*
[CSS:EBCCH:HYPERband:INFO]
Specifies INFO. Range of *n* is 0 to 3.

Used to specify the Hyperband associated with the specified channel.

INFO?
Returns current value of INFO.

MCC *n*
[CSS:EBCCH:MCC]
Specifies Mobile Country Code. Range of *n* is 0 to 1023.

Indicates the Mobile Country Code of the current DCCH.

MCC?
Returns current value of MCC.

SID *n*
[CSS:EBCCH:SID]
Specifies System Identification. Range of *n* is 0 to 32767.

Provides a digital identification associated with a cellular system where each system is assigned a unique number.

SID?
Returns current value of SID.

CSS:

EBCCH:

ENABLE:

NONPublic *n*
[CSS:EBCCH:ENABLE:NONPublic]
Enables (*n* = 1) or disables (*n* = 0) Non-Public Probability Blocks.

NONPublic?
Returns current state of NONPublic.

NEIGHbor:

TDMA *n*
[CSS:EBCCH:ENABLE:NEIGHbor:TDMA]
Enables (*n* = 1) or disables (*n* = 0) Time Division Multiplex Access.

TDMA?
Returns current state of TDMA.

TDMA:

INFO *n*
[CSS:EBCCH:ENABLE:NEIGHbor:TDMA:INFO]
Enables (*n* = 1) or disables (*n* = 0) TDMA Info.

INFO?
Returns current state of INFO.

ANALOG *n*
[CSS:EBCCH:ENABLE:NEIGHbor:ANALOG]
Enables (*n* = 1) or disables (*n* = 0) Analog.

ANALOG?
Returns current state of ANALOG.

OTHER *n*
[CSS:EBCCH:ENABLE:NEIGHbor:OTHER]
Enables (*n* = 1) or disables (*n* = 0) Other Hyperband.

OTHER?
Returns current state of OTHER.

OTHER:

INFO *n*
[CSS:EBCCH:ENABLE:NEIGHbor:OTHER:INFO]
Enables (*n* = 1) or disables (*n* = 0) Other Hyperband Info.

INFO?
Returns current state of INFO.

CSS:

EBCCH:

ENABLE:

CHANnel *n*
[CSS:EBCCH:ENABLE:CHANnel]
Enables (*n* = 1) or disables (*n* = 0) RF Channel Allocation.

CHANnel?
Returns current state of CHANnel.

SIGnal *n*
[CSS:EBCCH:ENABLE:SIGnal]
Enables (*n* = 1) or disables (*n* = 0) alerting information to a mobile station user.

SIGnal?
Returns current state of SIGnal.

MACA:

EIGHT:

CONTrol *n*
[CSS:EBCCH:ENABLE:MACA:EIGHT:CONTrol]
Enables (*n* = 1) or disables (*n* = 0) MACA_8_CONTROL.

CONTrol?
Returns current state of CONTrol.

LIST *n*
[CSS:EBCCH:ENABLE:MACA:LIST]
Enables (*n* = 1) or disables (*n* = 0) MACA_LIST.

LIST?
Returns current state of LIST.

LIST:

OTHER *n*
[CSS:EBCCH:ENABLE:MACA:LIST:OTHER]
Enables (*n* = 1) or disables (*n* = 0) MACA_LIST Other Hyperband.

OTHER?
Returns current state of OTHER.

MCC *n*
[CSS:EBCCH:ENABLE:MCC]
Enables (*n* = 1) or disables (*n* = 0) Mobile Country Code.

MCC?
Returns current state of MCC.

CSS:

EBCCH:

ENABLE:

HYPERband:

INFO *n*
[CSS:EBCCH:ENABLE:HYPERband:INFO]
Enables (*n* = 1) or disables (*n* = 0) Hyperband Information.

INFO?
Returns current state of INFO.

7-16-5 SPACH COMMANDS

CSS:

SPACH:

BUILD:

HARD
[CSS:SPACH:BUILD:HARD]
Builds a Hard Page of any type.

ARQ
[CSS:SPACH:BUILD:ARQ]
Builds an ARQ SPACH Message of any type.

NONARQ
[CSS:SPACH:BUILD:NONARQ]
Builds a Non ARQ SPACH Message

LENGth:

HARD?
[CSS:SPACH:LENGth:HARD?]
Returns current value of hard page length (1 bit value).

ARQ?
[CSS:SPACH:LENGth:ARQ?]
Returns current value of ARQ SPACH message length (7 bit value).

NONARQ?
[CSS:SPACH:LENGth:NONARQ?]
Returns current value of NONARQ SPACH message length (7 bit value).

DATA:

HARD? *n*
[CSS:SPACH:DATA:HARD?]
Returns current 16 bit word (indexed by *n*) within a hard page. Range of *n* is 0 to 6.

ARQ? *n,m*
[CSS:SPACH:DATA:ARQ?]
Returns current 16 bit word (indexed by m) within a selected frame (*n*) of the ARQ SPACH message. Range of *n* is 0 to 255; range of *m* is 0 to 6.

NONARQ? *n,m*
[CSS:SPACH:DATA:NONARQ?]
Returns current 16 bit word (indexed by m) within a selected frame (*n*) of the NONARQ SPACH message. Range of *n* is 0 to 255; range of *m* is 0 to 6.

CSS:

SPACH:

PROGRAM:

HARD *dest*
[CSS:SPACH:PROGRAM:HARD]
Copies the hard page into the superframe. The location in the superframe is selected by *dest*. Range of *dest* is 0 to 31.

ARQ *dest,source,length*
[CSS:SPACH:PROGRAM:ARQ]
Copies the ARQ message into the superframe. The location in the superframe is selected by *dest*. The *source* selects the frame from the ARQ buffer. The number of frames moved is selected by *length*. Range of *dest* is 0 to 31; range of *source* is 0 to 255; range of length is 0 to 32.

NONARQ *dest,source,length*
[CSS:SPACH:PROGRAM:NONARQ]
Copies the NONARQ message into the superframe. The location in the superframe is selected by *dest*. The *source* selects the frame from the NONARQ buffer. The number of frames moved is selected by *length*. Range of *dest* is 0 to 31; range of *source* is 0 to 255; range of length is 0 to 32.

BU *n*
[CSS:SPACH:BU]
Specifies Burst Usage. Range of *n* is 0 to 7.

BU?
Returns current value of BU.

PCON *n,m*
[CSS:SPACH:PCON]
?? Paging Channel Continuation. *n* is (0-1) *m* is ( ).

PCON? *n*
Returns current value of PCON indexed by *n*. Range of *n* is 0 or 1.

BCN *n*
[CSS:SPACH:BCN]
Enables (*n* = 1) or disables (*n* = 0) Broadcast Channel Change Notification Flag.

BCN?
Returns current state of BCN.

PFM *n*
[CSS:SPACH:PFM]
Enables (*n* = 1) or disables (*n* = 0) Paging Frame Modifier.

PFM?
Returns current state of PFM.

CSS:

SPACH:

BT *n*
[CSS:SPACH:BT]
Specifies Burst Type. Range of *n* is 0 to 7.

BT?
Returns current value of BT.

IDT *n*
[CSS:SPACH:IDT]
Specifies Identity Type. Range of *n* is 0 to 3.

IDT?
Returns current value of IDT.

MSID:

MS *n,m*
[CSS:SPACH:MSID:MS]
Specifies the 18 Most Significant Bits (*m*) of Mobile Station Identification indexed by *n*. Range of *n* is 0 to 4; range of *m* is 0 to #h3FFFF.

MS? *n*
Returns the 18 Most Significant Bits of Mobile Station Identification indexed by *n*. Range of *n* is 0 to 4.

LS *n,m*
[CSS:SPACH:MSID:LS]
Specifies the 32 Least Significant Bits (*m*) of Mobile Station Identification indexed by *n*. Range of *n* is 0 to 4; range of *m* is 0 to #hFFFFFFFF.

LS? *n*
Returns the 32 Least Significant Bits of Mobile Station Identification indexed by *n*. Range of *n* is 0 to 4.

MIN1 "*n*"
[CSS:SPACH:MIN1]
Specifies MIN1 used in a SPACH Message. (ASCII string).
(example: "316/522-4981")

MIN1?
Returns current value of MIN1.

CSS:

SPACH:

MIN2 "*n*"
[CSS:SPACH:MIN2]
Specifies MIN2 used in a SPACH Message. (ASCII string).

(example: "316/522-4981")

MIN2?
Returns current value of MIN2.

MIN3 "*n*"
CSS:SPACH:MIN3
Specifies MIN3 used in a SPACH Message. (ASCII string).

(example: "316/522-4981")

MIN3?
Returns current value of MIN3.

UGID:

MS *n*
[CSS:SPACH:UGID:MS]
Specifies the 18 Most Significant Bits of User Group Identification. Range of *n* is 0 to #h3FFFF.

MS?
Returns the 18 Most Significant Bits of User Group Identification.

LS *n*
[CSS:SPACH:UGID:LS]
Sets the 32 Least Significant Bits of User Group Identification. Range of *n* is 0 to #hFFFFFFFF.

LS?
Returns the 32 Least Significant Bits of User Group Identification.

MM *n*
[CSS:SPACH:MM]
Enables (*n* = 1) or disables (*n* = 0) Message Mapping.

MM?
Returns current state of MM.

CSS:

SPACH:

PEA *n*
[CSS:SPACH:PEA]
Specifies Partial Echo Assigned. Range of *n* is 0 to 127.

Partial Echo value used by a mobile station during an ARQ mode transaction.

PEA?
Returns current value of PEA.

PI *n,m*
[CSS:SPACH:PI]
Enables (*m* = 1) or disables (*m* = 0) Polling Indicator indexed by *n*. Range of *n* is 0 to 79.

Indicates whether or not the BMI is soliciting a response (ARQ STAUS Frame) from the mobile station.

PI? *n*
Returns current state of PI indexed by *n*. Range of *n* is 0 to 79.

SRM *n*
[CSS:SPACH:SRM]
Enables (*n* = 1) or disables (*n* = 0) SPACH Response Mode.

Indicates how a mobile station is to respond once it has received all frames associated with a given SPACH message.

SRM?
Returns current state of SRM.

EHI *n*
[CSS:SPACH:EHI]
Enables (*n* = 1) or disables (*n* = 0) Extended Header Indicator.

EHI?
Returns current state of EHI.

MEA *n*
[CSS:SPACH:MEA]
Specifies Message Encryption Algorithm. Range of *n* is 0 to 3.

MEA?
Returns current value of MEA.

CSS:

SPACH:

MEK *n*
[CSS:SPACH:MEK]
Specifies Message Encryption Key. Range of *n* is 0 to 3.

MEK?
Returns current value of MEK.

RSVD:

Reserved

HEADER *n*
[CSS:SPACH:RSVD:HEADER]
Enables (*n* = 1) or disables (*n* = 0) reserved field in SPACH Header A.

HEADER?
Returns current state of HEADER.

ARQ *n*
[CSS:SPACH:RSVD:ARQ]
Specifies Automatic Retransmission Request. Range of *n* is 0 to 3.

ARQ?
Returns current value of ARQ.

ARM *n*
[CSS:SPACH:ARM]
Enables (*n* = 1) or disables (*n* = 0) ARQ Response Mode.

Indicates how a mobile station is to respond once it has received an ARQ frame with PI set to 1.

ARM?
Returns current state of ARM.

FRNO *n,m*
[CSS:SPACH:FRNO]
Specifies Frame Number (*m*) indexed by *n*. Range of *n* is 0 to 79; range of *m* is 0 to 31.

Uniquely identifies specific frames sent in support of an ARQ mode transaction.

FRNO? *n*
Returns current value of FRNO indexed by *n*. Range of *n* is 0 to 79.

CSS:

SPACH:

GA *n*
[CSS:SPACH:GA]
Enables (*n* = 1) or disables (*n* = 0) Go Away.
Indicates if the DCCH is barred.

GA?
Returns current state of GA.

MSGTYPE1:

<COMMAND FROM Table below>
[CSS:SPACH:MSGtype1:<Command from Table below>]
Specifies message type 1 used in a SPACH message.

MSGTYPE2:

<COMMAND FROM Table below>
[CSS:SPACH:MSGTYPE2:<Command from Table below>]
Specifies message type 2 used in a SPACH message.

MSGTYPE3:

<COMMAND FROM Table below>
[CSS:SPACH:MSGtype3:<Command from Table below>]
Specifies message type 3 used in a SPACH message.

MSGTYPE4:

<COMMAND FROM Table below>
[CSS:SPACH:MSGtype4:<Command from Table below>]
Specifies message type 4 used in a SPACH message.

ANALOG	AUDIT	BSCHALcon	BSMC
CAPability	DIGital	DRETRY	MSGWTG
PAGE	PU	RDATA	RDATA_ACCept
RDATA_REJect	RELease	REORDer	SOC
SPACHnotification	SSDUP	TESTreg	UCHAL
REG_ACCept	REG_REJect	USERalert	

CSS:

SPACH:

MEM *n*
[CSS:SPACH:MEM]
Enables (*n* = 1) or disables (*n* = 0) Message Encryption Mode.

Indicates whether or not message encryption algorithm A and message encryption domain A are enabled on the assigned voice channel.

MEM?
Returns current state of MEM.

SCC *n*
[CSS:SPACH:SCC]
Specifies SAT Color Code. Range of *n* is 0 to 3.

Defines SAT Color Code used on the assigned analog voice channel.

SCC?
Returns current value of SCC.

VMAC *n*
[CSS:SPACH:VMAC]
Specifies VMAC. Range of *n* is 0 to 15.

Indicates the power level to be used on the assigned analog voice channel.

VMAC?
Returns current value of VMAC.

CHAN *n*
[CSS:SPACH:CHAN]
Specifies Channel used in a Digital or Analog channel assignment. Range of *n* is 0 to 2047.

CHAN?
Returns current value of CHAN.

PROTocol *n*
[CSS:SPACH:PROTocol]
Specifies Protocol version. Range of *n* is 0 to 15.

PROTocol?
Returns current value of PROTocol.

CSS:

SPACH:

SUBaddress:

Identifies the subaddress of a called or calling party.

LENGth *n*
[CSS:SPACH:SUBaddress:LENGth]
Specifies Length of Subaddress Info content. Range of *n* is 0 to 255.

LENGth?
Returns current value of LENGth.

ODD_EVEN *n*
[CSS:SPACH:SUBaddress:ODD_EVEN]
Enables (*n* = 1) or disables (*n* = 0) Odd/Even Indicator.

ODD_EVEN?
Returns current state of ODD_EVEN.

TYPE *n*
[CSS:SPACH:SUBaddress:TYPE]
Specifies Type of Subaddress. Range of *n* is 0 to 7.

TYPE?
Returns current value of TYPE.

REServed *n*
[CSS:SPACH:SUBaddress:REServed]
Enables (*n* = 1) or disables (*n* = 0) Reserved.

REServed?
Returns current state of REServed.

ADDRess *n,m*
[CSS:SPACH:SUBaddress:ADDRess]
Specifies Subaddress (*m*) indexed by *n*. Range of *n* is 0 to 19; range of *m* is 0 to 255.

ADDRess? *n*
Returns current value of Subaddress indexed by *n*. Range of *n* is (0-19).

CSS:

SPACH:

DTX:

SUPport *n*
[CSS:SPACH:DTX:SUPport]
Specifies DTX Support. Range of *n* is 0 to 3.

Used to indicate DTX capabilities supported on the analog voice channel

SUPport?
Returns current value of SUPport.

DISPlay:

Used to supply display information that may be displayed to the mobile station user. The information contained in this information element is coded in IRA characters. If the mobile station receives this information element with a length exceeding the maximum length that it supports, the information element should be truncated

LENGth *n*
[CSS:SPACH:DISPlay:LENGth]
Specifies Length of Display info. Range of *n* is 0 to 81.

LENGth?
Returns current value of LENGth.

CHARacter *n,m*
[CSS:SPACH:DISPlay:CHARacter]
Specifies Display Character (m) indexed by *n*. Range of *n* is 0 to 255; range of *m* is 0 to 255.

Up to 82 instances may be sent.

CHARacter? *n*
Returns current value of CHARacter indexed by *n*. Range of *n* is 0 to 255.

REREG *n*
[CSS:SPACH:REREG]
Enables (*n* = 1) or disables (*n* = 0) Forced Re-registration.

Indicates if the mobile station is required to initiate a Registration attempt with Registration Type set to Forced.

REREG?
Returns current state of REREG.

CSS:

SPACH:

DEBUG *n*
[CSS:SPACH:DEBUG]
Enables (*n* = 1) or disables (*n* = 0) Debug Display Allowed.

When enabled, the mobile station is allowed to include a Display information element in the Audit Confirmation message.

DEBUG?
Returns current state of DEBUG.

AUTHBS *n*
[CSS:SPACH:AUTHBS]
Specifies AUTHBS. Range of *n* is 0 to #hFFFF.

Contains the output from the Authentication procedure.

AUTHBS?
Returns current value of AUTHBS.

BSMC *n*
[CSS:SPACH:BSMC]
Specifies Base Station Manufacture Code. Range of *n* is 0 to 255.

Identifies the assigned manufacture code. The BSMC value of 0 is reserved. A reserved BSMC value shall be considered an unknown base station manufacture code by the receiving mobile station.

BSMC?
Returns current value of BSMC.

CUSTOM:

LENGth *n*
[CSS:SPACH:CUSTOM:LENGth]
Specifies Length of Custom Control Control in octets. Range of *n* is 1 to 64.

LENGth?
Returns current value of LENGth.

CONTrol *n,m*
[CSS:SPACH:CUSTOM:CONTrol]
Specifies Custom Control (*m*) indexed by *n*. Range of *n* is 0 to 63; range of *m* is 0 to 255.

CONTrol?
Returns current value of CONTrol indexed by *n*.

CSS:

SPACH:

DVCC *n*
[CSS:SPACH:DVCC]
Specifies Digital Verification Color Code. Range of *n* is 0 to 255.

DVCC?
Returns current value of DVCC.

DMAC *n*
[CSS:SPACH:DMAC]
Specifies DMAC. Range of *n* is 0 to 15.

Indicates the power level to be used on the assigned digital traffic channel.

DMAC?
Returns current value of DMAC.

ATS *n*
[CSS:SPACH:ATS]
Specifies Assigned Time Slot. Range of *n* is 0 to 15.

ATS?
Returns current value of ATS.

SB *n*
[CSS:SPACH:SB]
Enables (*n* = 1) or disables (*n* = 0) Shortened Burst.

Defines whether the mobile station shall use the shortened burst initially on the assigned digital traffic channel.

SB?
Returns current state of SB.

TA *n*
[CSS:SPACH:TA]
Specifies Time Alignment. Range of *n* is 0 to 31.

Indicates the absolute timing offset from the standard offset reference (SOR) position.

TA?
Returns current value of TA.

CSS:

SPACH:

MODE:

DIC *n*
[CSS:SPACH:MODE:DIC]
Enables (*n* = 1) or disables (*n* = 0) Delay Interval Compensation Mode.

Used to control the application of the DIC mode in the mobile station. When received in the access parameters message, the domain of DIC application shall be the DCCH. When received in the Digital Traffic Channel Designation message, the domain of DIC application shall be the DTC

DIC?
Returns current state of DIC.

VOICE:

Identifies the mode to be used for the requested Voice Call.

VC *n*
[CSS:SPACH:MODE:VOICE:VC]
Specifies Voice Coder. Range of *n* is 0 to 7.

VC?
Returns current value of VC.

PM_V *n*
[CSS:SPACH:MODE:VOICE:VC]
Specifies Voice Privacy. Range of *n* is 0 to 7.

PM_V?
Returns current value of PM_V.

MEM:

Message Encryption Mode- Identifies the selected message encryption algorithm, key and domain.

MEA *n*
[CSS:SPACH:MODE:MEM:MEA]
Specifies Message Encryption Algorithm. Range of *n* is 0 to 7.

MEA?
Returns current value of MEA.

CSS:

SPACH:

MODE:

MEM:

MED *n*
[CSS:SPACH:MODE:MEM:MED]
Specifies Message Encryption Domain. Range of *n* is 0 to 7.

MED?
Returns current value of MED.

MEK *n*
[CSS:SPACH:MODE:MEM:MEK]
Specifies Message Encryption Key. Range of *n* is 0 to 7.

MEK?
Returns current value of MEK.

HYPERband:

If present, this information element is used to specify the Hyperband associated with the specified channel.

INFO *n*
[CSS:SPACH:MODE:HYPERband:INFO]
Specifies Hyperband Info. Range of *n* is 0 to 3.

Provides frequency band information

INFO?
Returns current value of INFO.

LT *n*
[CSS:SPACH:LT]
Enables (*n* = 1) or disables (*n* = 0) Last Try.

If a mobile station receives a Directed Retry and attempts a new access on another DCCH, it shall set the Last Try flag in the Origination or the Page Response message to the value of the Last Try flag received in the Directed Retry message. Otherwise, the mobile station shall set the Last Try flag to 0 at system access.

LT?
Returns current state of LT.

CSS:

SPACH:

RCF *n*
[CSS:SPACH:RCF]
Enables (*n* = 1) or disables (*n* = 0) Control Filler information.

When enabled, the mobile station is to read Control Filler information.

RCF?
Returns current state of RCF.

AUTH *n*
[CSS:SPACH:AUTH]
Enables (*n* = 1) or disables (*n* = 0) Authentication information.

When enabled, Authentication information is sent when making an access on an ACC as a result of a Directed Retry received on the DCCH.

AUTH?
Returns current state of AUTH.

MSGWTG:

NV *n*
[CSS:SPACH:MSGWTG:NV]
Specifies Number of Values. Range of *n* is 0 to 15.

NV?
Returns current value of NV.

TYPE *n,m*
[CSS:SPACH:MSGWTG:TYPE]
Specifies Type of Message Waiting (*m*) indexed by *n*. Range of *n* is 0 to 15; range of *m* is 0 to 15.

TYPE? *n*
Returns current value of TYPE indexed by *n*. Range of *n* is 0 to 15.

NUMber n,m
[CSS:SPACH:MSGWTG:NUMber]
Specifies Number of Messages Waiting (*m*) indexed by *n*. Range of *n* is 0 to 15; range of *m* is 0 to 63.

Up to 16 instances of this field may be sent.

NUMber? *n*
Returns current value of NUMber indexed by *n*. Range of *n* is 0 to 15.

CSS:

SPACH:

SERVice *n*
[CSS:SPACH:SERVice]
Specifies Service Code. Range of *n* is 0 to 15.

Indicates the requested service.

SERVice?
Returns current value of SERVice.

SIGnal:

PITCH *n*
[CSS:SPACH:SIGnal:PITCH]
Specifies Signal Pitch. Range of *n* is 0 to 3.

PITCH?
Returns current value of PITCH.

CADence *n*
[CSS:SPACH:SIGnal:CADence]
Specifies Signal Cadence. Range of *n* is 0 to 63.

CADence?
Returns current value of CADence.

DURation *n*
[CSS:SPACH:SIGnal:DURation]
Specifies Signal Duration. Range of *n* is 0 to 15.

DURation?
Returns current value of DURation.

CALLED:

Called Party - Identifies the called party associated with a mobile station.

TYPE *n*
[CSS:SPACH:CALLED:TYPE]
Specifies Type of Number. Range of *n* is 0 to 7.

TYPE?
Returns current value of TYPE.

PLANid *n*
[CSS:SPACH:CALLED:PLANid]
Specifies Numbering Plan Identification. Range of *n* is 0 to 15.

PLANid?
Returns current value of PLANid.

CSS:

SPACH:

CALLED:

ENCoding *n*
[CSS:SPACH:CALLED:ENCoding]
Enables (*n* = 1) or disables (*n* = 0) Address Encoding.

ENCoding?
Returns current state of ENCoding.

ADDRess "*n*"
[CSS:SPACH:CALLED:ADDRess]
Specifies Called Address. (ASCII string).

ADDRess?
Returns current string value of ADDRess.

SUBaddress:
Identifies the address of a called or calling party.

LENGth *n*
[CSS:SPACH:CALLED:SUBaddress:LENGth]
Specifies Length of Subaddress Info content. Range of *n* is 0 to 255.

LENGth?
Returns current value of LENGth.

ODD_EVEN *n*
[CSS:SPACH:CALLED:SUBaddress:ODD_EVEN]
Enables (*n* = 1) or disables (*n* = 0) Odd/Even Indicator.

ODD_EVEN?
Returns current state of ODD_EVEN.

TYPE *n*
[CSS:SPACH:CALLED:SUBaddress:TYPE]
Specifies Type of Subaddress. Range of *n* is 0 to 7.

TYPE?
Returns current value of TYPE.

REServed *n*
[CSS:SPACH:CALLED:SUBaddress:REServed]
Enables (*n* = 1) or disables (*n* = 0) Reserved.

REServed?
Returns current state of REServed.

CSS:

SPACH:

CALLED:

ADDRess *n,m*
[CSS:SPACH:CALLED:SUBaddress:ADDRess]
Specifies Called Subaddress (*m*) indexed by *n*. Range of *n* is 0 to 19; range of *m* is 0 to 255.

ADDRess? *n*
Returns current value of ADDRess indexed by *n*. Range of *n* is 0 to 19.

CALLING:

Calling Party - Identifies the calling party associated with a mobile station.

TYPE *n*
[CSS:SPACH:CALLING:TYPE]
Specifies Type of Number. Range of *n* is 0 to 7.

TYPE?
Returns current value of TYPE.

PLANid *n*
[CSS:SPACH:CALLING:PLANid]
Specifies Numbering Plan Identification. Range of *n* is 0 to 15.

PLANid?
Returns current value of PLANid.

ENCoding *n*
[CSS:SPACH:CALLING:ENCoding]
Enables (*n* = 1) or disables (*n* = 0) Address Encoding.

ENCoding?
Returns current state of ENCoding.

ADDRess "*n*"
[CSS:SPACH:CALLING:ADDRess]
Specifies Address. Range of *n* is 0 to 255.

Up to 254 instances of this field may be sent.

ADDRess?
Returns current value of ADDRess.

CSS:

SPACH:

CALLING:

SUBaddress:

Identifies the address of a called or calling party.

LENGth *n*
[CSS:SPACH:CALLING:SUBaddress:LENGth]
Specifies Length of Subaddress Info content. Range of *n* is 0 to 255.

LENGth?
Returns current value of LENGth.

ODD_EVEN *n*
[CSS:SPACH:CALLING:SUBaddress:ODD_EVEN]
Enables (*n* = 1) or disables (*n* = 0) Odd/Even Indicator.

ODD_EVEN?
Returns current state of ODD_EVEN.

TYPE *n*
[CSS:SPACH:CALLING:SUBaddress:TYPE]
Specifies Type of Subaddress. Range of *n* is 0 to 7.

TYPE?
Returns current value of TYPE.

REServed *n*
[CSS:SPACH:CALLING:SUBaddress:REServed]
Enables (*n* = 1) or disables (*n* = 0) Reserved.

REServed?
Returns current state of REServed.

ADDRess *n,m*
[CSS:SPACH:CALLING:SUBaddress:ADDRess]
Specifies Calling Subaddress (*m*) indexed by *n*. Range of *n* is 0 to 19; range of *m* is 0 to 255.

ADDRess? *n*
Returns current value of Calling Subaddress indexed by *n*. Range of *n* is 0 to 19.

CSS:

SPACH:

CALLING:

PRESentation:

Identifies the presentation restrictions and screening related to the Calling Party information element.

PI *n*
[CSS:SPACH:CALLING:PRESentation:PI]
Specifies Calling Party Number Presentation Indicator. Range of *n* is 0 to 3.

PI?
Returns current value of PI.

SI *n*
[CSS:SPACH:CALLING:PRESentation:SI]
Specifies Screening Indicator. Range of *n* is 0 to 3.

SI?
Returns current value of SI.

RN *n*
[CSS:SPACH:RN]
Specifies Request Number. Range of *n* is 0 to 15.

Used in a Parameter Update message to allow the mobile station to recognize duplicate Parameter Update messages.

RN?
Returns current value of RN.

RTRANSaction *n*
[CSS:SPACH:RTRANSaction]
Specifies R-Transaction Identifier. Range of *n* is 0 to 255.

Used to associate a R-DATA ACCEPT or a R-DATA REJECT message to the R-DATA message being acknowledged.

RTRANSaction?
Returns current value of RTRANSaction.

CSS:

SPACH:

RDATA_UNIT:

Contains the Higher Layer Protocol Data Unit and is mandatory in an R-DATA message.

LENGth *n*
[CSS:SPACH:RDATA_UNIT:LENGth]
Specifies Length Indicator. Range of *n* is 0 to 255.

LENGth?
Returns current value of LENGth.

HLP:

IDentifier *n*
[CSS:SPACH:RDATA_UNIT:HLP:IDentifier]
Specifies Higher Protocol Identifier. Range of *n* is 0 to 255.

Identifier?
Returns current value of IDentifier.

DATA *n,m*
[CSS:SPACH:RDATA_UNIT:HLP:DATA]
Specifies Higher Layer Protocol Data Unit (*m*) indexed by *n*. Range of *n* is 0 to 255; range of *m* is 0 to 255.

DATA? *n*
Returns current value of DATA indexed by *n*. Range of *n* is 0 to 255.

MESSage:

CENTer:

Identifies the Message Center Address for the message being sent.

TYPE *n*
[CSS:SPACH:MESSage:CENTer:TYPE]
Specifies Type of Number. Range of *n* is 0 to 7.

TYPE?
Returns current value of TYPE.

PLANid *n*
[CSS:SPACH:MESSage:CENTer:PLANid]
Specifies Numbering Plan Identification. Range of *n* is 0 to 15.

PLANid?
Returns current value of PLANid.

CSS:

SPACH:

MESSage:

CENTer:

ENCoding *n*
[CSS:SPACH:MESSage:CENTer:ENCoding]
Enables (*n* = 1) or disables (*n* = 0) Address Encoding.

ENCoding?
Returns current state of ENCoding.

ADDRess "*n*"
[CSS:SPACH:MESSage:CENTer:ADDRess]
Specifies Address. Range of *n* is 0 to 255.
Up to 254 instances of this field may be sent.

ADDRess?
Returns current value of ADDRess.

USER:

DEST:
User Destination Address.

TYPE *n*
[CSS:SPACH:USER:DEST:TYPE]
Specifies Type of Number. Range of *n* is 0 to 7.

TYPE?
Returns current value of TYPE.

PLANid *n*
[CSS:SPACH:USER:DEST:PLANid]
Specifies Numbering Plan Identification. Range of *n* is 0 to 15.

PLANid?
Returns current value of PLANid.

ENCoding *n*
[CSS:SPACH:USER:DEST:ENCoding]
Enables (*n* = 1) or disables (*n* = 0) Address Encoding.

ENCoding?
Returns current state of ENCoding.

CSS:

SPACH:

USER:

DEST:

ADDRess "*n*"
[CSS:SPACH:USER:DEST:ADDRess]
Specifies Address. Range of *n* is 0 to 255.

Up to 254 instances of this field may be sent.

ADDRess?
Returns current value of ADDRess.

SUBaddress:

Identifies the address of a called or calling party.

LENGth *n*
[CSS:SPACH:USER:DEST:SUBaddress:LENGth]
Specifies Length of Subaddress Info content. Range of *n* is 0 to 255.

LENGth?
Returns current value of LENGth.

ODD_EVEN *n*
[CSS:SPACH:USER:DEST:SUBaddress:ODD_EVEN]
Enables (*n* = 1) or disables (*n* = 0) Odd/Even Indicator.

ODD_EVEN?
Returns current state of ODD_EVEN.

TYPE *n*
[CSS:SPACH:USER:DEST:SUBaddress:TYPE]
Specifies Type of Subaddress. Range of *n* is 0 to 7.

TYPE?
Returns current value of TYPE.

REServed *n*
[CSS:SPACH:USER:DEST:SUBaddress:REServed]
Enables (*n* = 1) or disables (*n* = 0) Reserved.

REServed?
Returns current state of REServed.

CSS:

SPACH:

USER:

DEST:

SUBaddress:

ADDRess *n,m*
[CSS:SPACH:USER:DEST:SUBaddress:ADDRess]
Specifies User Destination Subaddress (*m*) indexed by *n*. Range of *n* is 0 to 19; range of *m* is 0 to 255.

ADDRess? *n*
Returns current value of ADDRess indexed by *n*. Range of *n* is 0 to 19.

GROUP:

Identifies the User Group ID that a mobile station has requested or has been allocated.

STATus *n*
[CSS:SPACH:USER:GROUP:STATus]
Specifies User Group Status. Range of *n* is 0 to 3.

STATUS?
Returns current value of STATUS.

TYPE *n*
[CSS:SPACH:USER:GROUP:TYPE]
Specifies User Group Type. Range of *n* is 0 to 3.

TYPE?
Returns current value of TYPE.

ID:

MS *n*
[CSS:SPACH:USER:GROUP:ID:MS]
Specifies the 18 Most Significant Bits of User Group Identification. Range of *n* is 0 to #h3FFFF.

MS?
Returns current value of MS.

CSS:

SPACH:

USER:

GROUP:

ID:

LS *n*
[CSS:SPACH:USER:GROUP:ID:LS]
Specifies 32 Least Significant Bits of User Group Identification. Range of *n* is 0 to #hFFFFFFFF.

LS?
Returns current value of LS.

ORIG:

TYPE *n*
[CSS:SPACH:USER:ORIG:TYPE]
Specifies Type of Number. Range of *n* is 0 to 7.

TYPE?
Returns current value of TYPE.

PLANid *n*
[CSS:SPACH:USER:ORIG:PLANid]
Specifies Numbering Plan Identification. Range of *n* is 0 to 15.

PLANid?
Returns current value of PLANid.

ENCoding *n*
[CSS:SPACH:USER:ORIG:ENCoding]
Enables (*n* = 1) or disables (*n* = 0) Address Encoding.

ENCoding?
Returns current state of ENCoding.

ADDRess "*n*"
[CSS:SPACH:USER:ORIG:ADDRess]
Specifies Address. Range of *n* is 0 to 255.

Up to 254 instances of this field may be sent.

ADDRess?
Returns current value of ADDRess.

CSS:

SPACH:

USER:

ORIG:

SUBaddress:

Identifies the address of a called or calling party.

LENGth *n*
[CSS:SPACH:USER:ORIG:SUBaddress:LENGth]
Specifies Length of Subaddress Info content. Range of *n* is 0 to 255.

LENGth?
Returns current value of LENGth.

ODD_EVEN *n*
[CSS:SPACH:USER:ORIG:SUBaddress:ODD_EVEN]
Enables (*n* = 1) or disables (*n* = 0) Odd/Even Indicator.

ODD_EVEN?
Returns current state of ODD_EVEN.

TYPE *n*
[CSS:SPACH:USER:ORIG:SUBaddress:TYPE]
Specifies Type of Subaddress. Range of *n* is 0 to 7.

TYPE?
Returns current value of TYPE.

REServed *n*
[CSS:SPACH:USER:ORIG:SUBaddress:REServed]
Enables (*n* = 1) or disables (*n* = 0) Reserved.

REServed?
Returns current state of REServed.

ADDRess *n,m*
[CSS:SPACH:USER:ORIG:SUBaddress:ADDRess]
Specifies User Origination Subaddress (*m*) indexed by *n*. Range of *n* is 0 to 19; range of *m* is 0 to 255.

ADDRess? *n*
Returns current value of ADDRess indexed by *n*. Range of *n* is 0 to 19.

CSS:

SPACH:

PFC:

ASSIGNment *n*
[CSS:SPACH:PFC:ASSIGNment]
Specifies PFC Assignment. Range of *n* is 0 to 3.

Identifies the Paging Frame Class that a mobile station may be assigned at registration.

ASSIGNment?
Returns current value of ASSIGNment.

RNUM:

Contains the registration number that is used to define a particular mobile station's VMLA (Virtual Mobile Location Area).

NUMber *n*
[CSS:SPACH:RNUM:NUMber]
Specifies Number of RNUMs. Range of *n* is 1 to 50.

NUMber?
Returns current value of NUMber.

LIST *n,m*
[CSS:SPACH:RNUM:LIST]
Specifies LIST (*m*) indexed by *n*. Range of *n* is 0 to 49; range of *m* is 0 to 1023.

Up to 50 instances of this field may be sent.

LIST? *n*
Returns current value of LIST indexed by *n*. Range of *n* is 0 to 49.

MSID:

Mobile Station Identification Assignment - Contains information specifying the MSID the mobile station shall use.

IDT *n*
[CSS:SPACH:MSID:IDT]
Specifies Identity Type. Range of *n* is 0 to 3.

IDT?
Returns current value of IDT.

CSS:

SPACH:

MSID:

ASSIGNment *n*
[CSS:SPACH:MSID:ASSIGNment]
Specifies MSID Assignment. Range of *n* is 0 to #hFFFFFF.

ASSIGNment?
Returns current value of ASSIGNment.

PSID_RSID:

Private/Residential System Identification.

AVAILable:

PSID/RSID Available

NUMBer *n*
[CSS:SPACH:PSID_RSID:AVAILable:NUMBer]
Specifies Number of PSID/RSID. Range of *n* is 0 to 15.

NUMBer?
Returns current value of NUMBer.

TYPE *n,m*
[CSS:SPACH:PSID_RSID:AVAILable:TYPE]
Enables (*m* = 1) or disables (*m* = 0) PSID/RSID Type Indicator indexed by *n*. Range of *n* is 0 to 15.

TYPE? *n*
Returns current state of TYPE indexed by *n*. Range of *n* is 0 to 15.

VALUE *n,m*
[CSS:SPACH:PSID_RSID:AVAILable:VALUE]
Specifies PSID/RSID Value (*m*) indexed by *n*. Range of *n* is 0 to 15; range of *m* is 0 to #hFFFF.

VALUE? *n*
Returns current value of VALUE indexed by *n*. Range of *n* is 0 to 15.

CSS:

SPACH:

PSID_RSID:

MAP *n*
[CSS:SPACH:PSID_RSID:MAP]
Specifies PSID/RSID Map. Range of *n* is 0 to #hFFFF.

This information is included in the Test Registration message to indicate which private/residential systems have been queried by the mobile station. This information is included in the Test Registration Response message to indicate the private/residential systems on which the mobile station may receive service. The ordering of the PSID/RSID Map reflects the ordering of the PSID/RSID Set sent on the system identity message.

MAP?
Returns current value of MAP.

DIRectory:

TYPE *n*
[CSS:SPACH:DIRectory:TYPE]
Specifies Type of Number. Range of *n* is 0 to 7.

TYPE?
Returns current value of TYPE.

PLANid *n*
[CSS:SPACH:DIRectory:PLANid]
Specifies Numbering Plan Identification. Range of *n* is 0 to 15.

PLANid?
Returns current value of PLANid.

ENCoding *n*
[CSS:SPACH:DIRectory:ENCoding]
Enables (*n* = 1) or disables (*n* = 0) Address Encoding.

ENCoding?
Returns current state of ENCoding.

ADDRess "*n*"
[CSS:SPACH:DIRectory:ADDRess]
Specifies Address. Range of *n* is 0 to 255.

Up to 254 instances of this field may be sent.

ADDRess?
Returns current value of ADDRess.

CSS:

SPACH:

DIRectory:

SUBaddress:

Identifies the address of a called or calling party.

LENGth *n*
[CSS:SPACH:DIRectory:SUBaddress:LENGth]
Specifies Length of Subaddress Info content. Range of *n* is 0 to 255.

LENGth?
Returns current value of LENGth.

ODD_EVEN *n*
[CSS:SPACH:DIRectory:SUBaddress:ODD_EVEN]
Enables (*n* = 1) or disables (*n* = 0) Odd/Even Indicator.

ODD_EVEN?
Returns current state of ODD_EVEN.

TYPE *n*
[CSS:SPACH:DIRectory:SUBaddress:TYPE]
Specifies Type of Subaddress. Range of *n* is 0 to 7.

TYPE?
Returns current value of TYPE.

REServed *n*
[CSS:SPACH:DIRectory:SUBaddress:REServed]
Enables (*n* = 1) or disables (*n* = 0) Reserved.

REServed?
Returns current state of REServed.

ADDRess *n,m*
[CSS:SPACH:DIRectory:SUBaddress:ADDRess]
Specifies Directory Subaddress (*m*) indexed by *n*. Range of *n* is 0 to 19; range of *m* is 0 to 255.

ADDRess? *n*
Returns current value of Directory Subaddress indexed by *n*. Range of *n* is 0 to 19.

CSS:

SPACH:

REJect:

REGistration:

Registration Reject message

CAUSE *n*
[CSS:SPACH:REJect:REGistration:CAUSE]
Specifies CAUSE. Range of *n* is 0 to 15.

Indicates the cause for a Registration Reject.

CAUSE?
Returns current value of CAUSE.

TIME:

Reject Time - Used by the system to indicate to a mobile station the interval of time when it will be allowed to register, again.

LOWer *n*
[CSS:SPACH:REJect:REGistration:TIME:LOWer]
Specifies Lower time boundary in 100 Superframe (SF). Range of *n* is 0 to 15.

LOWer?
Returns current value of LOWer.

UPPer *n*
[CSS:SPACH:REJect:REGistration:TIME:UPPer]
Specifies Upper time boundary in 100 Superframe (SF). Range of *n* is 0 to 15.

UPPer?
Returns current value of UPPer.

RDATA:

CAUSE *n*
[CSS:SPACH:REJect:RDATA:CAUSE]
Specifies CAUSE. Range of *n* is 0 to 127.

The cause for the R-DATA Reject.

CAUSE?
Returns current value of CAUSE.

CSS:

SPACH:

REJect:

RDATA:

SPARE *n*
[CSS:SPACH:REJect:RDATA:SPARE]
Enables (*n* = 1) or disables (*n* = 0) R-Cause Reserved field.

SPARE?
Returns current state of SPARE.

RELease:

Used when the BMI (Base Station, MSC and Interworking Function) clears a mobile station terminated call.

CAUSE *n*
[CSS:SPACH:RELease:CAUSE]
Specifies CAUSE. Range of *n* is 0 to 15.

Indicates the cause for a Release.

CAUSE?
Returns current value of CAUSE.

REorder:

Used when the BMI (Base Station, MSC and Interworking Function) rejects an Origination or a R-DATA message sent by the mobile station.

CAUSE *n*
[CSS:SPACH:REorder:CAUSE]
Specifies CAUSE. Range of *n* is 0 to 15.

Indicates the cause for a Reorder/Intercept.

CAUSE?
Returns current value of CAUSE.

TONE *n*
[CSS:SPACH:REorder:TONE]
Specifies Tone Indicator. Range of *n* is 0 to 3.

Used to indicate the type of tone to be generated by the mobile station.

TONE?
Returns current value of TONE.

CSS:

SPACH:

SOC *n*
[CSS:SPACH:SOC]
Specifies System Operator Code. Range of *n* is 0 to 4095.

Identifies which operator is providing service. A reserved SOC value shall be considered an unknown system operator code by a receiving mobile station.

SOC?
Returns current value of SOC.

NOTification *n*
[CSS:SPACH:NOTification]
Specifies SPACH Notification Type. Range of *n* is 0 to 63.

Contains the message type identifying the message that the BMI intends to deliver to the mobile station. The valid values for SPACH Notification Type shall be limited to the Message Type associated with Page, SSD Update and R-DATA.

NOTification?
Returns current value of NOTification.

RANDSSD1 *n*
[CSS:SPACH:RANDSSD1]
Specifies Shared Secret Data. Range of *n* is 0 to #hFFFFFF.

Identifies a random number generated by the BMI that is used in the SSD (Shared Secret Data) Update procedure.

RANDSSD1?
Returns the value of RANDSSD1.

RANDSSD2 *n*
[CSS:SPACH:RANDSSD2]
Specifies Shared Secret Data. Range of *n* is 0 to #hFFFFFFFF.

Identifies a random number generated by the BMI that is used in the SSD (Shared Secret Data) Update procedure.

RANDSSD2?
Returns current value of RANDSSD2.

CSS:

SPACH:

ALPHA:

SID "*n*"
[CSS:SPACH:ALPHA:SID]
Specifies Alphanumeric System ID. *n* is a phone number, e.g., 316/522-4981.

The purpose of the Alphanumeric System ID information element is to supply an alphanumeric system ID to each user. The information contained in this information element is coded in IRA characters..

SID?
Returns current value of SID.

PSID_RSID:

Alphanumeric PSID/RSID List - The purpose of this information element is to supply an Alphanumeric PSID/RSID to the user. The ordering of the Alphanumeric PSID/RSID list reflects the ordering of the PSID/RSID Set sent to the System Identity message. The information contained in this information element is coded in IRA characters.

NUMBer *n*
[CSS:SPACH:ALPHA:PSID_RSID:LENGth]
Specifies Length of Alphanumeric PSID/RSID List. Range of *n* is 0 to 16.

NUMBer?
Returns current value of LENGth.

NAME:

CHARacter *n*,"*m*"
[CSS:SPACH:ALPHA:PSID_RSID:NAME:CHARacter]
Specifies Display Character (*m*) indexed by *n*. Range of *n* is (0-16); *m* is an ASCII string.

CHARacter? *n*
Returns current value of CHARacter indexed by *n*. Range of *n* is 0 to 16.

RANDU *n*
[CSS:SPACH:RANDU]
Specifies RANDU. Range of *n* is 0 to #hFFFFFF.

Identifies the random number generated by the BMI that is used in the Unique Challenge Response procedure.

RANDU?
Returns current value of RANDU.

CSS:

SPACH:

ENABLE:

SUBaddress *n*
[CSS:SPACH:ENABLE:SUBaddress]
Enables (*n* = 1) or disables (*n* = 0) Subaddress.

SUBaddress?
Returns current state of SUBaddress.

DTX *n*
[CSS:SPACH:ENABLE:DTX]
Enables (*n* = 1) or disables (*n* = 0) DTX Support.

DTX?
Returns current state of DTX.

DISPlay *n*
[CSS:SPACH:ENABLE:DISPlay]
Enables (*n* = 1) or disables (*n* = 0) Display.

DISPlay?
Returns current state of DISPlay.

MODE:

VOICE *n*
[CSS:SPACH:ENABLE:MODE:VOICE]
Enables (*n* = 1) or disables (*n* = 0) Voice Mode.

VOICE?
Returns current state of VOICE.

MEM *n*
[CSS:SPACH:ENABLE:MODE:MEM]
Enables (*n* = 1) or disables (*n* = 0) Message Encryption Mode.

MEM?
Returns current state of MEM.

HYPERband:

INFO *n*
[CSS:SPACH:ENABLE:HYPERband:INFO]
Enables (*n* = 1) or disables (*n* = 0) Hyperband Info.

INFO?
Returns current state of INFO.

CSS:

SPACH:

ENABLE:

RCF_AUTH *n*
[CSS:SPACH:ENABLE:RCF_AUTH]
Enables (*n* = 1) or disables (*n* = 0) RCF and AUTH.

RCF_AUTH?
Returns current state of RCF_AUTH.

SIGnal *n*
[CSS:SPACH:ENABLE:SIGnal]
Enables (*n* = 1) or disables (*n* = 0) SIGnal.

SIGnal?
Returns current state of SIGnal.

CALLED:

ADDRess *n*
[CSS:SPACH:ENABLE:CALLED:ADDRess]
Enables (*n* = 1) or disables (*n* = 0) Called Party Address.

ADDRess?
Returns current state of ADDRess.

SUBaddress *n*
[CSS:SPACH:ENABLE:CALLED:SUBaddress]
Enables (*n* = 1) or disables (*n* = 0) Called Party Subaddress.

SUBaddress?
Returns current state of SUBaddress

CALLING:

ADDRess *n*
[CSS:SPACH:ENABLE:CALLING:ADDRess]
Enables (*n* = 1) or disables (*n* = 0) Calling Party Address.

ADDRess?
Returns current state of ADDRess.

SUBaddress *n*
[CSS:SPACH:ENABLE:CALLING:SUBaddress]
Enables (*n* = 1) or disables (*n* = 0) Calling Party Subaddress.

SUBaddress?
Returns current state of SUBaddress

CSS:

SPACH:

ENABLE:

CALLING:

PRESentation *n*
[CSS:SPACH:ENABLE:CALLING:PRESentation]
Enables (*n* = 1) or disables (*n* = 0) Calling Party Number Presentation Indicator.

PRESentation?
Returns current state of PRESentation.

MESSage:

CENTer:

ADDRess *n*
[CSS:SPACH:ENABLE:MESSage:CENTer:ADDRess]
Enables (*n* = 1) or disables (*n* = 0) Message Center Address.

ADDRess?
Returns current state of ADDRess.

USER:

DEST:

ADDRess *n*
[CSS:SPACH:ENABLE:USER:DEST:ADDRess]
Enables (*n* = 1) or disables (*n* = 0) User Destination Address.

ADDRess?
Returns current state of ADDress.

SUBaddress *n*
[CSS:SPACH:ENABLE:USER:DEST:SUBaddress]
Enables (*n* = 1) or disables (*n* = 0) User Destination Subaddress.

SUBaddress?
Returns current state of SUBaddress.

ORIG:

ADDRess *n*
[CSS:SPACH:ENABLE:USER:ORIG:ADDRess]
Enables (*n* = 1) or disables (*n* = 0) User Originating Address.

ADDRess?
Returns current state of ADDRess.

CSS:

SPACH:

ENABLE:

USER:

ORIG:

SUBaddress *n*
[CSS:SPACH:ENABLE:USER:ORIG:SUBaddress]
Enables (*n* = 1) or disables (*n* = 0) User Originating Subaddress.

SUBaddress?
Returns current state of SUBaddress.

GROUP *n*
[CSS:SPACH:ENABLE:USER:GROUP]
Enables (*n* = 1) or disables (*n* = 0) User Group.

GROUP?
Returns current state of GROUP.

PFC:

Paging Frame Class.

ASSIGNment *n*
[CSS:SPACH:ENABLE:PFC:ASSIGNment]
Enables (*n* = 1) or disables (*n* = 0) PFC Assignment.

ASSIGNment?
Returns current state of ASSIGNment.

RNUM:

Registration Number.

LIST *n*
[CSS:SPACH:ENABLE:RNUM:LIST]
Enables (*n* = 1) or disables (*n* = 0) RNUM List.

LIST?
Returns current state of LIST.

CSS:

SPACH:

ENABLE:

MSID:

Mobile Station Identification.

ASSIGNment *n*
[CSS:SPACH:ENABLE:MSID:ASSIGNment]
Enables (*n* = 1) or disables (*n* = 0) MSID Assignment.

ASSIGNment?
Returns current state of ASSIGNment.

PSID_RSID:

Private/Residential System Identification.

AVAILable *n*
[CSS:SPACH:ENABLE:PSID_RSID:AVAILable]
Enables (*n* = 1) or disables (*n* = 0) PSID/RSID Available.

AVAILable?
Returns current state of AVAILable.

DIRectory:

ADDRess *n*
[CSS:SPACH:ENABLE:DIRectory:ADDRess]
Enables (*n* = 1) or disables (*n* = 0) Directory Address.

ADDRess?
Returns current state of ADDRess.

SUBaddress *n*
[CSS:SPACH:ENABLE:DIRectory:SUBaddress]
Enables (*n* = 1) or disables (*n* = 0) Directory Subaddress.

SUBaddress?
Returns current state of SUBaddress.

REJect:

TIME *n*
[CSS:SPACH:ENABLE:REJect:TIME]
Enables (*n* = 1) or disables (*n* = 0) Reject Time.

TIME?
Returns current state of TIME.

CSS:

SPACH:

ENABLE:

ALPHA:

SID *n*
[CSS:SPACH:ENABLE:ALPHA:SID]
Enables (*n* = 1) or disables (*n* = 0) Alphanumeric System ID.

SID?
Returns current state of SID.

PSID_RSID *n*
[CSS:SPACH:ENABLE:ALPHA:PSID_RSID]
Enables (*n* = 1) or disables (*n* = 0) Alphanumeric PSID/RSID List.

PSID_RSID?
Returns current state of PSID_RSID.

7-17 RDCCH DATA MONITOR

7-17-1 SETUP COMMANDS

RDCCH:

SETup
[RDCCH:SETup]
Configures the FM/AM-1600CSA to receive on the RDCCH.

The 1600S is forced into the duplex mode of operation.

SLOT *n*
[RDCCH:SLOT]
Specifies SLOT pair on which to receive. Range of *n* is 1 to 3.

SLOT?
Returns current value of SLOT.

CHANnel *n*
[RDCCH:CHANnel]
Specifies Reverse Channel to monitor. Range of *n* is 1 to 1023.

The FM/AM-1600S is forced into Duplex Operation.

CHANnel?
Returns current value of CHANnel.

RATE *n*
[RDCCH:RATE]
Selects Transmission Rate: Full (*n* = 0) or Half (*n* = 1).

RATE?
Returns current setting of RATE.

DVCC *n*
[RDCCH:DVCC]
Specifies Digital Verification Color Code. Range of *n* is 1 to 255.

Digital Verification Color Code, which is used when calculating the CRC, must be specified to decode the message.

DVCC?
Returns current value of DVCC.

RDCCH:

LENGth:

NORMal
[RDCCH:LENGth:NORMal]
Configures the 1600CSA to decode normal length bursts on the RDCCH.

ABBREViated
[RDCCH:LENGth:ABBREViated]
Configures the 1600CSA to decode abbreviated length bursts on the RDCCH.

LENGth?
Returns current setting of LENGth: 0 = Normal, 1 = Abbreviated.

7-17-2 CONTINUOUS REMOTE RAW TIMESLOT DATA

The RDCCH timeslot data is presented exactly as received with no de-interleaving, error correction or formatting of data. The number of bits in a timeslot is 312 for a normal slot and 274 for an abbreviated slot, corresponding to 78 ASCII characters. The data on the RDCCH is present only when the mobile station is accessing the system. The baud rate should be set to 57600 for both the FM/AM-1600CSA and the RS-232 terminal. The following TMAC commands are used to start and stop this operation:

RDCCH:

REMote:

TIMEslot:

STARt
[RDCCH:REMote:TIMEslot:STARt]
Starts sending the received data out the RS-232 Connector.

STOP
[RDCCH:REMote:TIMEslot:STOP]
Stops sending the received data out the RS-232 Connector.

7-17-3 CONTINUOUS REMOTE RAW DATA

This mode differs from Continuous Remote Raw Timeslot Data in Section 7-17-2. This mode de-interleaves the data and performs forward error correction before presenting the data. The timeslot is broken into the various data fields and continuously transmitted out the RS-232 Connector. Following the data is a millisecond time stamp which provides relative time between slots of data.

The RDCCH data is presented in hexadecimal out the RS-232 Connector with each frame of data being separated by a newline character. The data is formatted as follows:

- The first 7 characters represent the 28 bits of the SYNC word.
- The next 6 characters are the hexadecimal value of the 24 bits of SYNC+.
- The next 30 characters are the hexadecimal value (left justified) of the 117 bits of data.
- The flag is followed by a millisecond time stamp.

The following TMAC commands are used to set up and start and stop this operation:

RDCCH:

REMote:

RAW:

DVCC *n*
[RDCCH:REMote:RAW:DVCC]
Specifies Digital Verification Color Code. Range of *n* is 1 to 255.

A DVCC, which is used when calculating the CRC, must be specified to decode the message.

LENGth:

ABBREViated
[RDCCH:REMote:RAW:LENGth:ABBREViated]
Configures the FM/AM-1600CSA to decode abbreviated length bursts on the RDCCH.

NORMal
[RDCCH:REMote:RAW:LENGth:NORMal]
Configures the FM/AM-1600CSA to decode normal length bursts on the RDCCH.

STARt
[RDCCH:REMote:RAW:STARt]
Starts sending the received, de-interleaved and decoded data out the RS-232 Connector.

STOP
[RDCCH:REMote:RAW:STOP]
Stops sending the received data out the RS-232 Connector.

An embedded macro named RRAW initiates the START when executed and sends a STOP when any key on the RS-232 Terminal is pressed. To use this macro, type in the command RRAW at the RS-232 terminal.

7-17-4 BUFFERED RAW DATA

The RDCCH Buffered Raw Data commands consist of the TMAC commands used for Layer 1 raw data buffering. Up to 100 frames of raw data can be captured (similar to FOCC raw data TMAC commands). The following commands are used for this operation:

RDCCH:

RAW:

COUNT?
[RDCCH:RAW:COUNT]
Returns current number of frames (0 to 100 bit value) received and stored into the raw buffer.

When COUNT is equal to DEPTH, the raw buffer is full. Allows the user to access or decode the raw data as it is received instead of waiting until the raw buffer is completely full.

DATA? *n,x*
[RDCCH:RAW:DATA]
Returns current raw data byte (8 bit value) (indexed by *x*) in selected raw data frame (*n*). Range of *n* is 0 to 99; range of *x* is 0 to 15.

DEPTH *n*
[RDCCH:RAW:DEPTH]
Specifies depth of the raw buffer. Range of *n* is 0 to 99.

PREAMble? *n*
[RDCCH:RAW:PREAMble]
Returns current value of Preamble (16 bit value) in selected raw data frame (*n*). Range of *n* is 0 to 99.

STARt
[RDCCH:RAW:STARt]
Starts capturing raw data on the RDCCH.

STOP
[RDCCH:RAW:STOP]
Stops capturing raw data on the RDCCH.

STOP happens automatically when the buffer is full.

SYNC? *n*
[RDCCH:RAW:SYNC]
Returns current value of SYNC (28 bit value) in selected raw data frame (*n*). Range of *n* is 0 to 99.

SYNCPLUS? *n*
[RDCCH:RAW:SYNCPLUS]
Returns current value of SYNC+ (24 bit value) in selected raw data frame (*n*). Range of *n* is 0 to 99.

RDCCH:

RAW:

TS? *n*
[RDCCH:RAW:TS]
Returns Time Stamp (in milliseconds) of selected raw data frame (*n*). Range of *n* is 0 to 99.

7-17-5 LAYER 2 DATA MONITOR

The Layer 2 Data Monitor is made up of the TMAC commands necessary to decode a Layer 2 message.

The process involves the following steps:

1. Capture 1 to 100 words into the raw data buffer for non real-time decoding into Layer 2 data fields. Use the RDCCH Buffered Raw data commands described in Section 7-17-4.
2. Select one frame of raw data from the raw buffer and decode the frame into Layer 2 fields:

RDCCH:

LAYER2:

DECode *n*
[RDCCH:LAYER2:DECode]
Decodes selected frame of data (*n*) in the raw buffer. Range of *n* is 0 to 99.

The decoded Layer 2 data (after issuing the **RDCCH:LAYER2:DECode** command) is accessed with the following commands:

RDCCH:

LAYER2:

RACH:

ARQ_RSVD?
[RDCCH:LAYER2:RACH:ARQ_RSVD]
Returns current value of Automatic Retransmission Request RSVD (2 bit value). Returns -1 if not available.

BT?
[RDCCH:LAYER2:RACH:BT]
Returns current value of Burst Type (3 bit value). Returns -1 if not available.

CI?
[RDCCH:LAYER2:RACH:CI]
Returns current state of Change Indicator (1 bit value). Returns -1 if not available.

EHI?
[RDCCH:LAYER2:RACH:EHI]
Returns current state of Extension Header Indicator (1 bit value). Returns -1 if not available.

EH_RSVD?
[RDCCH:LAYER2:RACH:EH_RSVD]
Returns current state of Extended Header RSVD (1 bit value). Returns -1 if not available.

END_RSVD?
[RDCCH:LAYER2:RACH:END_RSVD]
Returns current state of END frame RSVD (1 bit value). Returns -1 if not available.

FRNO_MAP?
[RDCCH:LAYER2:RACH:FRNO_MAP]
Returns current value of Frame Number Map (32 bit value). Returns -1 if not available.

IDT?
[RDCCH:LAYER2:RACH:IDT]
Returns current value of Identity Type (2 bit value). Returns -1 if not available.

RDCCH:

LAYER2:

RACH:

L3DATA? *n,x*
[RDCCH:LAYER2:RACH:L3DATA]
Returns current 8 bit word (indexed by *x*) of selected Layer 3 Data message (*n*). Returns -1 if not available. Range of *n* is 0 to 7; range of *x* is 0 to 15.

The number of Layer 3 data messages embedded within a Layer 2 frame can be just a portion of single data message on up to 8 full data messages. The maximum number of bytes in a Layer 2 frame is 16 bytes. The data returned is left justified.

The L3LENGTH command may be used to determine how many 8 bit "words" make up the Layer 3 data field in the frame currently being decoded.

L3LENGTH? *n*
[RDCCH:LAYER2:RACH:L3LENGTH]
Returns current 8 bit value of Length (indexed by *n*) in Layer 3 data field. Returns -1 if not available. Range of *n* is 0 to 7.

Length of the Layer 3 data field in the RACH Layer 2 frame currently being decoded. This is necessary because the L3LIs are all contained in the first frame of a multiframe RACH message and subsequent frames do not have the information in them specifying the length of the Layer 3 data field.

L3LI? *n*
[RDCCH:LAYER2:RACH:L3LI]
Returns current 8 bit value of Layer 3 Length Indicator indexed by *n*. Returns -1 if not available. Range of *n* is 0 to 7.

A RACH Layer 2 frame may contain up to eight Layer 3 Length Indicators.

MEA?
[RDCCH:LAYER2:RACH:MEA]
Returns current value of Message Encryption Algorithm (2 bit value). Returns -1 if not available.

MEK?
[RDCCH:LAYER2:RACH:MEK]
Returns current value of Message Encryption Key (2 bit value). Returns -1 if not available.

MIN?
[RDCCH:LAYER2:RACH:MIN]
Returns current value of Mobile Identification Number (ASCII string). Returns -1 if not available.

This command is associated with a 34 bit Mobile Station Identity.

RDCCH:

LAYER2:

RACH:

MSID:

LS?
[RDCCH:LAYER2:RACH:MSID:LS]
Returns the 32 Least Significant Bits of Mobile Station Identification. Returns -1 if not available.

MS?
[RDCCH:LAYER2:RACH:MSID:MS]
Returns the 18 Most Significant Bits of Mobile Station Identification. Returns -1 if not available.

NL3M?
[RDCCH:LAYER2:RACH:NL3M]
Returns current value of Number of Layer 3 Messages (3 bit value). Returns -1 if not available.

PEA?
[RDCCH:LAYER2:RACH:PEA]
Returns current value of Partial Echo Assigned (7 bit value). Returns -1 if not available.

7-17-6 RDCCH REAL TIME DATA MONITOR

RDCCH:

START
[RDCCH:START]
Starts a background task running that decodes the RDCCH message information elements. Elements that are decoded and returned with their corresponding TMAC command.

STOP
[RDCCH:STOP]
Stops the RDCCH decode background task.

SYNC?
[RDCCH:SYNC?]
Returns the last decoded value of the RDCCH sync word (28 bit value). Returns -1 if not available.

SYNCPlus?
[RDCCH:SYNCPLus?]
Returns the last decoded value of the RDCCH sync plus word. (24 bit value). Returns -1 if not available.

BT?
[RDCCH:BT?]
Returns current value of Burst Type (3 bit value). Returns -1 if not available.

CI?
[RDCCH:CI?]
Returns current state of Change Indicator (1 bit value). Returns -1 if not available.

EHI?
[RDCCH:EHI?]
Returns current state of Extension Header Indicator (1 bit value). Returns -1 if not available.

IDT?
Returns current value of Identity Type (2 bit value). Returns -1 if not available.

MSID:

MS?
[RDCCH:MSID:MS?]
Returns the 18 Most Significant bits of Mobile Station Identification. Returns -1 if not available.

LS?
[RDCCH:MSID:LS?]
Returns the 32 Least Significant bits of Mobile Station Identification. Returns -1 if not available.

RDCCH:

MIN?
[RDCCH:MIN?]
Returns current Mobile Identification Number (ASCII String) associated with a 34 bit Mobile Station Identity. Returns -1 if not available.

NL3M?
[RDCCH:NL3M?]
Returns current value of Number of Layer 3 Messages (3 bit value). Returns -1 if not available.

L3LI? *n*
[RDCCH:L3LI?]
Returns current 8 bit value of Layer 3 Length Indicator indexed by *n*. Returns -1 if not available. Range of *n* is 0 to 7.

A RACH layer 2 frame may contain up to 8 layer 3 length indicators.

PEA?
[RDCCH:PEA?]
Returns current value of Partial Echo Assigned (7 bit value). Returns -1 if not available.

MEA?
[RDCCH:MEA?]
Returns current value of Message Encryption Algorithm (2 bit value). Returns -1 if not available.

MEK?
[RDCCH:MEK?]
Returns current value of Message Encryption Key (2 bit value). Returns -1 if not available.

FRNO_MAP?
[RDCCH:FRNO_MAP?]
Returns current value of Frame Number Map (26 bit value). Returns -1 if not available.

RSVD:

EHI?
[RDCCH:RSVD:EHI?]
Returns current state of Extended Header Indicator RSVD (1 bit value). Returns -1 if not available.

ARQ?
[RDCCH:RSVD:ARQ?]
Returns current value of ARQ RSVD field (2 bit value). Returns -1 if not available.

END?
[RDCCH:RSVD:END?]
Returns current state of END frame RSVD field (1 bit value). Returns -1 if not available.

RDCCH:

CRC?
[RDCCH:CRC?]
Returns current value of Cyclic Redundancy Code (16 bit value). Returns -1 if not available.

L3DATA:

SELect *n*
[RDCCH:L3DATA:SELect]
Specifies Layer 3 data message from which to access data. Range of *n* is 0 to 7.

Up to eight Layer 3 messages can be included in one RDCCH message. Each message is decoded and stored. The following commands access the information elements from the selected layer 3 message.

SELect?
Returns current value of SELect.

Returns the last selection of eight Layer 3 messages.

MSGtype?
[RDCCH:MSGtype?]
Returns current Message Type.

The following are possible message types:

AUDITCON	AUTHENTICATION	BSCHALcon	BSMC
CAPability	MACA	ORIGINATION	PAGE RESPONSE
R-DATA	R-DATA ACCEPT	RDATA REJECT	REGISTRATION
SERIAL NUMBER	SOC	SPACHCON	SSD UPDATE
TEST	UCHAL		

PD?
[RDCCH:PD?]
Returns current value of Protocol Discriminator (2 bit value). Returns -1 if not available.

PFC_1?
[RDCCH:PFC_1?]
Returns current value of Paging Frame Class (3 bit value). Returns -1 if not available.

PSID_RSID:

SELect?
[RDCCH:PSID_RSID:SELect?]
Returns current value of Selected PSID/RSID (4 bit value). Returns -1 if not available.

MAP?
[RDCCH:PSID_RSID:MAP?]
Returns current value of PSID/RSID Map (16 bit value). Returns -1 if not available.

RDCCH:

DISPlay:

LENGth?
[RDCCH:DISPlay:LENGth?]
Returns current value of Length of Display Information (8 bit value). Returns -1 if not available.

CHARacter? *n*
[RDCCH:DISPlay:CHARacter?]
Returns current 8 bit value of Display Character indexed by *n*. Range of *n* is 0 to 81.

COUNt?
[RDCCH:COUNt?]
Returns current value of COUNt (6 bit value). Returns -1 if not available.

RANDC?
[RDCCH:RANDC?]
Returns current value of RANDC (8 bit value). Returns -1 if not available.

AUTHR?
[RDCCH:AUTHR?]
Returns current value of AUTHR (18 bit value). Returns -1 if not available.

RANDBS?
[RDCCH:RANDBS?]
Returns current value of RANDBS (32 bit value). Returns -1 if not available.

BSMC?
[RDCCH:BSMC?]
Returns current value of Base Station Manufacture Code (8 bit value). Returns -1 if not available.

PROTocol:

VERsion?
[RDCCH:PROTocol:VERsion?]
Returns current value of Protocol Version (4 bit value). Returns -1 if not available.

SCM?
[RDCCH:SCM?]
Returns current value of Station Class Mark (5 bit value). Returns -1 if not available.

VINtage:

SOFTware?
[RDCCH:VINtage:SOFTware?]
Returns current value of Software Vintage (6 bit value). Returns -1 if not available.

FIRMware?
[RDCCH:VINtage:FIRMware?]
Returns current value of Firmware Vintage (6 bit value). Returns -1 if not available.

RDCCH:

MODEL?
[RDCCH:MODEL?]
Returns current value of Model Number (4 bit value). Returns -1 if not available.

MANufacture?
[RDCCH:MANufacture?]
Returns current value of Manufacture Code (8 bit value). Returns -1 if not available.

SUPPort:

MAX:

PFC?
[RDCCH:SUPPort:MAX:PFC?]
Returns current value of MAX_SUPPORTED_PFC (3 bit value). Returns -1 if not available.

SUPPort:

SOC?
[RDCCH:SUPPort:SOC?]
Returns current value of SOC Support (1 bit value). Returns -1 if not available.

BSMC?
[RDCCH:SUPPort:BSMC?]
Returns current value of BSCM Support (1 bit value). Returns -1 if not available.

ASYNC?
[RDCCH:SUPPort:ASYNC?]
Returns current value of Async Data Support (1 bit value). Returns -1 if not available.

G3fax?
[RDCCH:SUPPort:G3fax?]
Returns current value of G3-Fax Support (1 bit value). Returns -1 if not available.

SMS?
[RDCCH:SUPPort:SMS?]
Returns current value of SMS Broadcast Support (1 bit value). Returns -1 if not available.

SUBaddress?
[RDCCH:SUPPort:SUBaddress?]
Returns current value of Subaddressing Support (1 bit value). Returns -1 if not available.

FREQuency:

BANDS?
[RDCCH:SUPPort:FREQuency:BANDS?]
Returns current value of Supported Frequency Bands (8 bit value). Returns -1 if not available.

RDCCH:

SUPPort:

IRA?
[RDCCH:SUPPort:IRA?]
Returns current value of IRA Support (1 bit value). Returns -1 if not available.

USER?
[RDCCH:SUPPort:USER?]
Returns current value of User Group Support (1 bit value). Returns -1 if not available.

ANA800?
[RDCCH:SUPPort:ANA800?]
Returns current value of 800 MHz Analog Speech Support (1 bit value). Returns -1 if not available.

HALF?
[RDCCH:SUPPort:HALF?]
Returns current value of Half-Rate DTC Support (1 bit value). Returns -1 if not available.

DOUBle?
[RDCCH:SUPPort:DOUBLe?]
Returns current value of Double Rate DTC Support (1 bit value). Returns -1 if not available.

TRIPle?
[RDCCH:SUPPort:TRIPLe?]
Returns current value of Triple Rate DTC Support (1 bit value). Returns -1 if not available.

MEASurement:

LTM:

WER?
[RDCCH:MEASurement:LTM:WER?]
Returns current value of LTM Measurement Receive Signal Strength (3 bit value). Returns -1 if not available.

BER?
[RDCCH:MEASurement:LTM:BER?]
Returns current value of LTM Measurement Bit Error Rate (3 bit value). Returns -1 if not available.

RSS?
[RDCCH:MEASurement:LTM:RSS?]
Returns current value of LTM Measurement Receive Signal Strength (5 bit value). Returns -1 if not available.

RDCCH:

MEASurement:

LTM:

FULL?
[RDCCH:MEASurement:LTM:FULL?]
Returns current value of LTM Measurement Full Measurement Indicator (1 bit value). Returns -1 if not available.

STM:

NV?
[RDCCH:MEASurement:STM:NV?]
Returns current value of STM Measurement Number of Values (4 bit value). Returns -1 if not available.

RSS? *n*
[RDCCH:MEASurement:STM:RSS?]
Returns current 5 bit value of STM Measurement Receive Signal Strength indexed by *n*. Range of *n* is 0 to 15. Returns -1 if not available.

EMERgency?
[RDCCH:EMERgency?]
Returns current value of Emergency Call (1 bit value). Returns -1 if not available.

LT?
[RDCCH:LT?]
Returns current value of Last Try (1 bit value). Returns -1 if not available.

SERVice?
[RDCCH:SERVice?]
Returns current value of Service Code (4 bit value). Returns -1 if not available.

MODE:

VOICe:

VC?
[RDCCH:MODE:VOICe:VC?]
Returns current value of Voice Mode VC (3 bit value). Returns -1 if not available.

PM?
[RDCCH:MODE:VOICe:PM?]
Returns current value of Voice Mode PM_V (3 bit value). Returns -1 if not available.

RDCCH:

MODE:

DATA:

PM?
[RDCCH:MODE:DATA:PM?]
Returns current value of Data Mode PM_D (3 bit value). Returns -1 if not available.

SAP?
[RDCCH:MODE:DATA:SAP?]
Returns current value of Data Mode SAP (1 bit value). Returns -1 if not available.

ACKED?
[RDCCH:MODE:DATA:ACKED?]
Returns current value of Data Mode Acked Data (1 bit value). Returns -1 if not available.

CRC?
[RDCCH:MODE:DATA:CRC?]
Returns current value of Data Mode CRC (2 bit value). Returns -1 if not available.

PART?
[RDCCH:MODE:DATA:PART?]
Returns current value of Data Mode Data Part (3 bit value). Returns -1 if not available.

RLP?
[RDCCH:MODE:DATA:RLP?]
Returns current value of Data Mode RLP (2 bit value). Returns -1 if not available.

MEM:

Message Encryption Mode- Identifies the selected message encryption algorithm, key and domain.

MEA?
[RDCCH:MEM:MEA?]
Returns current value of Message Encryption Algorithm (3 bit value). Returns -1 if not available.

MED?
[RDCCH:MEM:MED?]
Returns current value of Message Encryption Domain (3 bit value). Returns -1 if not available.

MEK?
[RDCCH:MEM:MEK?]
Returns current value of Message Encryption Key (3 bit value). Returns -1 if not available.

RDCCH:

BANDWidth?
[RDCCH:BANDWidth?]
Returns current value of Bandwidth (3 bit value). Returns -1 if not available.

CALLED:

Called Party - Identifies the called party associated with a mobile station.

ADDRess?
[RDCCH:CALLED:ADDRess?]
Returns current string value of ADDRess (ASCII string). Returns -1 if not available.

ENCoding?
[RDCCH:CALLED:ENCoding?]
Returns current value of ENCoding (1 bit value). Returns -1 if not available.

PLANid?
[RDCCH:CALLED:PLANid?]
Returns current value of PLANid (4 bit value). Returns -1 if not available.

TYPE?
[RDCCH:CALLED:TYPE?]
Returns current value of TYPE (3 bit value). Returns -1 if not available.

SUBaddress:

Identifies the address of a called or calling party.

LENGth?
[RDCCH:CALLED:SUBaddress:LENGth?]
Returns current value of LENGth (8 bit value). Returns -1 if not available.

ODD_EVEN?
[RDCCH:CALLED:SUBaddress:ODD_EVEN?]
Returns current value of ODD_EVEN (1 bit value). Returns -1 if not available.

TYPE?
[RDCCH:CALLED:SUBaddress:TYPE?]
Returns current value of TYPE (3 bit value). Returns -1 if not available.

ADDRess? *n*
[RDCCH:CALLED:SUBaddress:ADDRess?]
Returns last decoded 8 bit value of Called Subaddress from selected L3DATA Message (*n*). Range of *n* is 0 to 19. Returns -1 if not available.

RDCCH:

CALLING:

Calling Party - Identifies the calling party associated with a mobile station.

ADDRess?
[RDCCH:CALLING:ADDRess?]
Returns current string value of ADDRess (ASCII string). Returns -1 if not available.

ENCoding?
[RDCCH:CALLING:ENCoding?]
Returns current value of ENCoding (1 bit value). Returns -1 if not available.

PRESentation:

Identifies the presentation restrictions and screening related to the Calling Party information element.

PI?
[RDCCH:CALLING:PRESentation:PI?]
Returns current value of PI (2 bit value). Returns -1 if not available.

SI?
[RDCCH:CALLING:PRESentation:SI?]
Returns current value of SI (2 bit value). Returns -1 if not available.

PLANid?
[RDCCH:CALLING:PLANid?]
Returns current value of PLANid (4 bit value). Returns -1 if not available.

TYPE?
[RDCCH:CALLING:TYPE?]
Returns current value of TYPE (3 bit value). Returns -1 if not available.

SUBaddress:

Identifies the address of a called or calling party.

LENGth?
[RDCCH:CALLING:SUBaddress:LENGth?]
Returns current value of LENGth (8 bit value). Returns -1 if not available.

ODD_EVEN?
[RDCCH:CALLING:SUBaddress:ODD_EVEN?]
Returns current value of ODD_EVEN (1 bit value). Returns -1 if not available.

TYPE?
[RDCCH:CALLING:SUBaddress:TYPE?]
Returns current value of TYPE (3 bit value). Returns -1 if not available.

RDCCH:

CALLING:

SUBaddress:

ADDRess? *n*
[RDCCH:CALLING:SUBaddress:ADDRess?]
Returns last decoded 8 bit value of Calling Subaddress from selected L3DATA Message (*n*). Range of *n* is 0 to 19. Returns -1 if not available.

RTRANSaction?
[RDCCH:RTRANSaction?]
Returns current value of RTRANSaction (8 bit value). Returns -1 if not available.

RDATA_UNIT:

Contains the Higher Layer Protocol Data Unit and is mandatory in an R-DATA message.

LENGth?
[RDCCH:RDATA_UNIT:LENGth?]
Returns current value of LENGth (8 bit value). Returns -1 if not available.

HLP:

IDentifier?
[RDCCH:RDATA_UNIT:HLP:IDentifier?]
Returns current value of IDentifier (8 bit value). Returns -1 if not available.

DATA? *n*
[RDCCH:RDATA_UNIT:HLP:DATA?]
Returns current 8 bit value of DATA indexed by *n*. Range of *n* is 0 to 255. Returns -1 if not available.

RCAUSE?
[RDCCH:RCAUSE?]
Returns current value of RCAUSE (8 bit value). Returns -1 if not available.

MESSage:

CENTer:

Identifies the Message Center Address for the message being sent.

ADDRess?
[RDCCH:MESSage:CENTer:ADDRess?]
Returns current string value of ADDRess (ASCII string). Returns -1 if not available.

ENCoding?
[RDCCH:MESSage:CENTer:ENCoding?]
Returns current value of ENCoding (1 bit value). Returns -1 if not available.

RDCCH:

MESSage:

CENTer:

PLANid?
[RDCCH:MESSage:CENTer:PLANid?]
Returns current value of PLANid (4 bit value). Returns -1 if not available.

TYPE?
[RDCCH:MESSage:CENTer:TYPE?]
Returns current value of TYPE (3 bit value). Returns -1 if not available.

USER:

GROUP:
Identifies the User Group ID that a mobile station has requested or has been allocated.

STATUS?
[RDCCH:USER:GROUP:STATUS?]
Returns current value of STATUS (2 bit value). Returns -1 if not available.

TYPE?
[RDCCH:USER:GROUP:TYPE?]
Returns current value of TYPE (2 bit value). Returns -1 if not available.

UGID:

MS?
[RDCCH:USER:GROUP:UGID:MS?]
Returns the 18 Most Significant Bits of User Group Identification.

LS?
[RDCCH:USER:GROUP:UGID:LS?]
Returns the 32 Least Significant Bits of User Group Identification.

DEST:

ADDRess?
[RDCCH:USER:DEST:ADDRess?]
Returns current string value of ADDRess (ASCII string). Returns -1 if not available.

ENCoding?
[RDCCH:USER:DEST:ENCoding?]
Returns current value of ENCoding (1 bit value). Returns -1 if not available.

PLANid?
[RDCCH:USER:DEST:PLANid?]
Returns current value of PLANid (4 bit value). Returns -1 if not available.

RDCCH:

USER:

DEST:

TYPE?
[RDCCH:USER:DEST:TYPE?]
Returns current value of TYPE (3 bit value). Returns -1 if not available.

SUBaddress:

Identifies the address of a called or calling party.

LENGth?
[RDCCH:USER:DEST:SUBaddress:LENGth?]
Returns current value of LENGth (8 bit value). Returns -1 if not available.

ODD_EVEN?
[RDCCH:USER:DEST:SUBaddress:ODD_EVEN?]
Returns current value of ODD_EVEN (1 bit value). Returns -1 if not available.

TYPE?
[RDCCH:USER:DEST:SUBaddress:TYPE?]
Returns current value of TYPE (3 bit value). Returns -1 if not available.

ADDRess? *n*
[RDCCH:USER:DEST:SUBaddress:ADDRess?]
Returns last decoded 8 bit value of User Destination Subaddress from selected L3DATA Message (*n*). Range of *n* is 0 to 19. Returns -1 if not available.

ORIG:

ADDRess?
[RDCCH:USER:ORIG:ADDRess?]
Returns current string value of ADDRess (ASCII string). Returns -1 if not available.

ENCoding?
[RDCCH:USER:ORIG:ENCoding?]
Returns current value of ENCoding (1 bit value). Returns -1 if not available.

PLANid?
[RDCCH:USER:ORIG:PLANid?]
Returns current value of PLANid (4 bit value). Returns -1 if not available.

TYPE?
[RDCCH:USER:ORIG:TYPE?]
Returns current value of TYPE (3 bit value). Returns -1 if not available.

RDCCH:

USER:

ORIG:

SUBaddress:

Identifies the address of a called or calling party.

LENGth?
[RDCCH:USER:ORIG:SUBaddress:LENGth?]
Returns current value of LENGth (8 bit value). Returns -1 if not available.

ODD_EVEN?
[RDCCH:USER:ORIG:SUBaddress:ODD_EVEN?]
Returns current value of ODD_EVEN (1 bit value). Returns -1 if not available.

TYPE?
[RDCCH:USER:ORIG:SUBaddress:TYPE?]
Returns current value of TYPE (3 bit value). Returns -1 if not available.

ADDRess? ***n***
[RDCCH:USER:ORIG:SUBaddress:ADDRess?]
Returns last decoded 8 bit value of User Origination Subaddress from selected L3DATA Message (*n*). Range of *n* is 0 to 19. Returns -1 if not available.

REG:

TYPE?
[RDCCH:REG:TYPE?]
Returns current value of Registration Type (4 bit value). Returns -1 if not available.

CNUMBer:

ADDRess?
[RDCCH:CNUMber:ADDRess?]
Returns current string value of C-Number Address (ASCII string). Returns -1 if not available.

ENCoding?
[RDCCH:CNUMber:ENCoding?]
Returns current value of C-Number Address Encoding (1 bit value). Returns -1 if not available.

PLANid?
[RDCCH:CNUMber:PLANid?]
Returns current value of C-Number Identification Plan (4 bit value). Returns -1 if not available.

RDCCH:

TYPE?
[RDCCH:CNUMber:TYPE?]
Returns current value of C-Number Type of Number (3 bit value). Returns -1 if not available.

PFC:

REQuest?
[RDCCH:PFC:REQuest?]
Returns current value of Paging Frame Class Request (3 bit value). Returns -1 if not available.

DCCH_MEM:

ALGORithm?
[RDCCH:DCCH_MEM:ALGORithms?]
Returns current value of DCCH Message Encryption Algorithm (4 bit value). Returns -1 if not available.

DOMAIN?
[RDCCH:DCCH_MEM:DOMAIN?]
Returns current value of DCCH Message Encryption Domain (8 bit value). Returns -1 if not available.

KEY?
[RDCCH:DCCH_MEM:KEY?]
Returns current value of DCCH Message Encryption Key (4 bit value). Returns -1 if not available.

ESN?
[RDCCH:ESN?]
Returns current value of Electronic Serial Number (32 bit value). Returns -1 if not available.

SOC?
[RDCCH:SOC?]
Returns current value of SOC (12 bit value). Returns -1 if not available.

CUSTom:

LENGth?
[RDCCH:CUSTOM:LENGth?]
Returns current value of Custom Control Length (8 bit value). Returns -1 if not available.

CONTrol? *n*
[RDCCH:CUSTOM:CONTrol?]
Returns current 8 bit value of Custom Control indexed by. Range of *n* is 0 to 252) Returns -1 if not available.

RDCCH:

CONFIRMed:

MSGtype?
[RDCCH:CONFIRMed:MSGtype?]
Returns current value of Confirmed Message Type (6 bit value). Returns -1 if not available.

SSDUP:

STATus?
[RDCCH:SSDUP:STATus?]
Returns current value of SSD Update Status (2 bit value). Returns -1 if not available.

AUTHU?
[RDCCH:AUTHU?]
Returns current value of AUTHU (18 bit value). Returns -1 if not available.

SECTION 8 - CSA QUICK REFERENCE LIST

The CSA Quick Reference List is a brief listing of the Specific and most General TMAC commands used with the FM/AM-1600CSA. The Quick Reference List is an aid to the experienced TMAC user. If more detailed information is needed, refer to the specified page.

COMMAND	RANGE	PAGE	DESCRIPTION
BER COMMANDS			
BER:			
RDTC:			
BER?		7-61	Returns Bit Error Rate (percentage).
BITS?		7-61	Returns number of bits.
CHANnel *n*	1 to 1023	7-61	Selects Digital Traffic Channel.
CLEAR		7-62	Clears current results.
DATA:			
USER		7-61	Sends user selected RDTC data.
PSUEDO		7-61	Sends pseudo-random data.
LOOPBACK		7-61	Sends data from Base Station back to Base Station.
45MHZ_OFFset		7-61	Downconverts frequency 45 MHz and retransmits data.
ERRORS?		7-61	Returns number of bit errors.
GO		7-61	Starts Base Station Digital Traffic BER test.
RFLVL *n*	-30.0 to -127.0	7-61	Sets RF Level in dBm.
SETup		7-61	Sets FM/AM-1600CSA as when entering Base Station Digital Traffic BER screen.
SLOT *n*	1 to 3	7-61	Selects Digital Traffic Timeslot.
STATUS?		7-62	Returns synchronous data status (1 cannot sync or 0).
STOP		7-61	Stops Base Station Digital Traffic BER test.
CELL SITE SIMULATION COMMANDS			
CSS:			
CALL:			
CHANnel *n*	1 to 1023	7-34	Selects Mobile Station call Channel.
CHANnel?		7-34	Returns Mobile Station call Channel setting.
DEViation *n*	0.0 to 4.0	7-34	Sets SAT Deviation, in kHz.
DEViation?		7-34	Returns SAT Deviation setting, in kHz.
DMAC *n*	0 to 7	7-34	Selects Digital Mobile Attenuation Code.
DMAC?		7-34	Returns Digital Mobile Attenuation Code setting.
DVCC *n*	0 to 255	7-34	Selects Digital Verification Color Code.
DVCC?		7-34	Returns Digital Verification Color Code setting.
EF *n*	1 or 0	7-34	Sets Extended Protocol Forward Channel Indicator.
EF?		7-34	Returns Extended Protocol Forward Channel Indicator setting.
MEM *n*	1 or 0	7-34	Sets Message Encryption Mode.
MEM?		7-34	Returns Message Encryption Mode setting.
MIN *"n"*		7-34	Selects Mobile Identification Number to call.
MIN?		7-34	Returns MIN currently selected.
PM *n*	1 or 0	7-34	Sets Privacy Mode bit.
PM?		7-35	Returns Privacy Mode bit setting.
PROCess:			
ASSIGNment		7-36	Sends initial channel designation command.
FDTC:			
HANDoff?		7-37	Performs digital to digital or analog handoff.
FVC:			
HANDoff		7-36	Performs analog to analog handoff.
SLOT1		7-36	Performs analog to digital handoff (Timeslot 1).
SLOT2		7-36	Performs analog to digital handoff (Timeslot 2).
SLOT3		7-36	Performs analog to digital handoff (Timeslot 3).
MOBINIT		7-36	Sets FM/AM-1600CSA for Mobile Station initiated call.
PAGE		7-36	Simulates Base Station (Cellsite) initiated call.
REGistration		7-36	Causes Mobile Station to send Registration Message.
SAT *n*	5965 to 6035	7-35	Sets Supervisory Audio Tone in Hz.
SAT?		7-35	Returns Supervisory Audio Tone setting.
SLOT *n*	1 to 3	7-35	Selects Timeslot.
SLOT?		7-35	Returns Timeslot setting.
TYPE *n*	1 or 0	7-35	Selects channel type (1 [digital] or 0 [analog]).
TYPE?		7-35	Returns channel type setting.

COMMAND	RANGE	PAGE	DESCRIPTION
CSS:			
CALL:			
VMAC *n*	0 to 7	7-35	Selects Voice Mobile Attenuation Code.
VMAC?		7-35	Returns Voice Mobile Attenuation Code setting.
CHANnel *n*	1 to 1023	7-31	Selects Forward Control Channel.
FDTC:			
AMT:			
CONNect		7-45	Selects Connect Acknowledgment message.
RELease		7-45	Selects Release Acknowledgment message.
STATus		7-45	Selects Status Acknowledgment message.
AMT?		7-46	Returns Acknowledgment Message Type setting.
AUTHBS *n*	0 to 262143	7-46	Sets AUTHBS value.
AUTHBS?		7-46	Returns AUTHBS value setting.
CALLING:			
NUM *"n"*		7-46	Sets Calling Party Number (string).
NUM?		7-46	Returns Calling Party Number setting.
PI *n*	0 to 3	7-46	Sets Calling Party Number Presentation Indicator.
PI?		7-46	Returns Calling Party Number Presentation Indicator setting.
PLANid *n*	0 to 15	7-46	Sets Calling Party Numbering Plan Identification.
PLANid?		7-46	Returns Calling Party Numbering Plan Identification setting.
SI *n*	0 to 3	7-46	Sets Calling Party Screening Indicator.
SI?		7-46	Returns Calling Party Screening Indicator setting.
TYpe *n*	0 to 7	7-46	Sets Calling Party Type.
TYpe?		7-46	Returns Calling Party Type setting.
CONTROL *n*	0 to 31	7-46	Sets Local Control field.
CONTROL?		7-47	Returns Local Control field setting.
DIC *n*	1 or 0	7-47	Sets Delay Interval Compensation bit.
DIC?		7-47	Returns Delay Interval Compensation bit setting.
DMAC *n*	0 to 10	7-47	Sets Digital Mobile Attenuation Code.
DMAC?		7-47	Returns Digital Mobile Attenuation Code setting.
DTX *n*	1 or 0	7-47	Sets Discontinuous Transmission bit.
DTX?		7-47	Returns Discontinuous Transmission bit setting.
DVCC *n*	1 to 255	7-47	Sets Digital Verification Color Code.
DVCC?		7-47	Returns Digital Verification Color Code setting.
ENABLE:			
CALLING:			
NUM *n*	1 or 0	7-47	Enables or disables Calling Party Number field.
NUM?		7-47	Returns Calling Party Number Enable setting.
DIC *n*	1 or 0	7-46	Enables or disables Delay Interval Compensation bit.
DIC?		7-47	Returns Delay Interval Compensation bit Enable setting.
DMAC *n*	1 or 0	7-47	Enables or disables Digital Mobile Attenuation Code.
DMAC?		7-47	Returns Digital Mobile Attenuation Code Enable setting.
DTX *n*	1 or 0	7-47	Enables or disables Discontinuous Transmission bit.
DTX?		7-47	Returns Discontinuous Transmission bit Enable setting.
LDP:			
BSACK *n*	1 or 0	7-48	Enables or disables LDP in BS ACK message.
BSACK?		7-48	Returns LDP Enable setting for BS ACK message.
FLASHACK *n*	1 or 0	7-48	Enables or disables LDP in FLASH ACK message.
FLASHACK?		7-48	Returns LDP Enable setting for FLASH ACK message.
SBDA *n*	1 or 0	7-48	Enables or disables LDP SBDA message.
SBDA?		7-48	Returns LDP Enable setting for SBDA message.
NOMW *n*	1 or 0	7-48	Enables or disables Number of Messages Waiting field.
NOMW?		7-48	Returns Number of Messages Waiting Enable setting.
RFCHAN *n,x*	0 to 11, 1 or 0	7-48	Enables or disables selected RF Channel index.
RFCHAN? *n*	0 to 11	7-48	Returns selected RF Channel index Enable setting.
SIGNAL *n*	1 or 0	7-48	Enables or disables Signal field.
SIGNAL?		7-48	Returns Signal field Enable setting.
STATUS:			
CMODE *n*	1 or 0	7-48	Enables or disables Call Mode field in Status Request.
CMODE?		7-48	Returns Call Mode field Enable setting.
ESN *n*	1 or 0	7-49	Enables or disables Electronic Serial Number field in Status Request message.
ESN?		7-49	Returns Electronic Serial Number field Enable setting.
MEM *n*	1 or 0	7-49	Enables or disables Message Encryption Mode field.
MEM?		7-49	Returns Message Encryption Mode field Enable setting.

COMMAND	RANGE	PAGE	DESCRIPTION
CSS:			
FDTC:			
ENABLE:			
STATUS:			
TI *n*	1 or 0	7-49	Enables or disables Terminal Information field.
TI?		7-49	Returns Terminal Information field Enable setting.
VPM *n*	1 or 0	7-49	Enables or disables Voice Privacy Mode bit.
VPM?		7-49	Returns Voice Privacy Mode bit Enable setting.
TA *n*	1 or 0	7-49	Enables or Disables Time Alignment field.
TA?		7-49	Returns Time Alignment field Enable setting.
FACCH:			
ALERT		7-43	Sends Alert with Information message.
AUDIT		7-43	Sends Audit message.
BSACK		7-43	Sends Base Station Acknowledgment message.
BSCHALCON		7-43	Sends Base Station Challenge Confirmation message.
FLASH		7-43	Sends Flash with Information message.
FLASHACK		7-43	Sends Flash Acknowledgment message.
HANDoff		7-43	Sends Handoff message.
LC		7-43	Sends Local Control message.
MAINTenance		7-43	Sends Maintenance message.
MEASure		7-43	Sends Measurement message.
PLC		7-43	Sends Physical Layer Control message.
PU		7-43	Sends Parameter Update message.
RELease		7-44	Sends Release message.
SBDA		7-44	Sends Send Burst DTMF Acknowledge message.
SCDA		7-44	Sends Send Continuous DTMF Acknowledge message.
SMEASure		7-44	Sends Stop Measurement message.
SR		7-44	Sends Status Request message.
SSDUP		7-44	Sends Shared Secret Data Update message.
UCHAL		7-44	Sends Unique Challenge message.
HANDoff:			
CHANnel *n*	1 to 1023	7-49	Sets Channel for Handoff.
CHANnel?		7-49	Returns Channel for Handoff.
LDP *n*	0 to 15	7-49	Sets Last Decoded Parameter.
LDP?		7-49	Returns Last Decoded Parameter setting.
MEM *n*	1 or 0	7-50	Sets Message Encryption Mode.
MEM?		7-50	Returns Message Encryption Mode setting.
NOMW *n*	0 to 63	7-50	Sets Number of Messages Waiting field.
NOMW?		7-50	Returns Number of Messages Waiting field setting.
RANDSSD *"n"*	0 to FFFFFFFFFFFFFFFF	7-50	Sets RANDSSD 56 bit value in hexadecimal (string).
RANDSSD?		7-50	Returns RANDSSD 56 bit value setting.
RANDU *n*	0 to 16777215	7-50	Sets 24 bit Random Number in UCHAL message.
RANDU?		7-50	Returns UCHAL 24 bit Random Number setting.
RATe *n*	1 or 0	7-50	Sets Channel Rate (1 [half-rate] or 0 [full-rate]).
RATe?		7-50	Returns Channel Rate setting.
RFCHAN *n,x*	0 to 11, 1 to 1023	7-50	Sets selected RF Channel index.
RFCHAN? *n*	0 to 11	7-50	Returns selected RF Channel index setting.
RN *n*	0 to 15	7-50	Sets Request Number.
RN?		7-50	Returns Request Number setting.
SACCH:			
ALERT		7-44	Sends Alert with Information message.
AUDIT		7-44	Sends Audit message.
BSACK		7-44	Sends Base Station Acknowledgment message.
BSCHALCON		7-44	Sends Base Station Challenge Confirmation message.
FLASH		7-44	Sends Flash with Information message.
FLASHACK		7-44	Sends Flash Acknowledgment message.
HANDoff		7-44	Sends Handoff message.
LC		7-44	Sends Local Control message.
MAINTenance		7-45	Sends Maintenance message.
MEASure		7-45	Sends Measurement message.
PLC		7-45	Sends Physical Layer Control message.
PU		7-45	Sends Parameter Update message.
RELease		7-45	Sends Release message.
SBDA		7-45	Sends Send Burst DTMF Acknowledge message.
SCDA		7-45	Sends Send Continuous DTMF Acknowledge message.
SMEASure		7-45	Sends Stop Measurement message.
SR		7-45	Sends Status Request message.

COMMAND	RANGE	PAGE	DESCRIPTION
CSS:			
FDTC:			
SACCH:			
SSDUP		7-45	Sends Shared Secret Data Update message.
UCHAL		7-45	Sends Unique Challenge message.
SBI *n*	0 to 3	7-50	Sets Shortened Burst Indicator.
SBI?		7-50	Returns Shortened Burst Indicator setting.
SIGNAL:			
CADENCE *n*	0 to 63	7-51	Sets on, off pattern of Alert tone.
CADENCE?		7-51	Returns Alert tone on, off pattern setting.
PITCH *n*	0 to 3	7-51	Sets pitch of Alert tone.
PITCH?		7-51	Returns Alert tone pitch setting.
SLOT *n*	1 to 3	7-51	Sets Timeslot.
SLOT?		7-51	Returns Timeslot setting.
TA *n*	0 to 31	7-51	Sets Time Alignment.
TA?		7-51	Returns Time Alignment setting.
TALK:			
DELAY *n*	0 to 250	7-52	Adds delay between receiving and transmitting in 20 ms intervals.
START		7-52	Starts Talkback operation.
STOP		7-52	Stops Talkback operation.
TI *n*	0 to 6	7-51	Sets Timeslot Indicator. (0 is analog.)
TI?		7-51	Returns Timeslot Indicator setting.
VPM *n*	1 or 0	7-51	Sets Voice Privacy Mode bit.
VPM?		7-51	Returns Voice Privacy Mode bit setting.
FOCC:			
AUTH *n*	1 or 0	7-31	Sets Authentication bit.
AUTH?		7-31	Returns Authentication bit setting.
B_I *n*	1 or 0	7-31	Sets Busy-Idle bit.
B_I?		7-31	Returns Busy-Idle bit setting.
CMAC *n*	0 to 7	7-31	Sets Control Mobile Attenuation Code.
CMAC?		7-31	Returns Control Mobile Attenuation Code setting.
CMAX *n*	1 to 32	7-31	Sets maximum Number of Access Channels to be scanned.
CMAX?		7-31	Returns maximum Number of Access Channels setting.
DCC *n*	0 to 3	7-32	Sets Digital Color Code.
DCC?		7-32	Returns Digital Color Code setting.
E *n*	1 or 0	7-32	Sets Extended Address bit.
E?		7-32	Returns Extended Address bit setting.
EP *n*	1 or 0	7-32	Sets Extended Protocol bit.
EP?		7-32	Returns Extended Protocol bit setting.
N *n*	1 to 32	7-32	Sets Number of Paging Channels to be scanned.
N?		7-32	Returns Number of Paging Channels setting.
PCI *n*	1 or 0	7-32	Sets Protocol Capability Indicator.
PCI?		7-32	Returns Protocol Capability Indicator setting.
RCF *n*	1 or 0	7-32	Sets Read Control Filler bit.
RCF?		7-32	Returns Read Control Filler bit setting.
REGH *n*	1 or 0	7-32	Sets Home Registration bit.
REGH?		7-32	Returns Home Registration bit setting.
REGR *n*	1 or 0	7-32	Sets Roaming Registration bit.
REGR?		7-32	Returns Roaming Registration bit setting.
S *n*	1 or 0	7-33	Sets Serial Number bit.
S?		7-33	Returns Serial Number bit setting.
SDCC1 *n*	0 to 3	7-33	Sets Supplementary Digital Color Code 1.
SDCC1?		7-33	Returns Supplementary Digital Color Code 1 setting.
SDCC2 *n*	0 to 3	7-33	Sets Supplementary Digital Color Code 2.
SDCC2?		7-33	Returns Supplementary Digital Color Code 2 setting.
SID *n*	0 to 32767	7-33	Sets System Identification Number.
SID?		7-33	Returns System Identification Number setting.
WFOM *n*	1 or 0	7-33	Sets Wait For Overhead Message bit.
WFOM?		7-33	Returns Wait For Overhead Message bit setting.
FVC:			
AUTHBS *n*	0 to 262143	7-39	Sets AUTHBS value.
AUTHBS?		7-39	Returns AUTHBS value setting.

COMMAND	RANGE	PAGE	DESCRIPTION
CSS:			
FVC:			
CALLING:			
NUM *"n"*		7-40	Sets number of calling party (string).
NUM?		7-40	Returns number of calling party setting.
PI *n*	0 to 3	7-40	Sets Calling Party Number Presentation Indicator.
PI?		7-40	Returns Calling Party Number Presentation Indicator.
SI *n*	0 to 3	7-40	Sets Calling Party Screening Indicator.
SI?		7-40	Returns Calling Party Screening Indicator setting.
DMAC *n*	0 to 10	7-40	Sets Digital Mobile Attenuation Code.
DMAC?		7-40	Returns Digital Mobile Attenuation Code setting.
DVCC *n*	1 to 255	7-40	Sets Digital Verification Color Code.
DVCC?		7-40	Returns Digital Verification Color Code setting.
EF *n*	1 or 0	7-40	Sets Extended Protocol Forward Channel Indicator.
EF?		7-40	Returns EF setting.
ENABLE:			
VOICEPrivacy *n*	1 or 0	7-40	Sets Voice Privacy in CALLMODEACK message.
VOICEPrivacy?		7-40	Returns Voice Privacy setting.
HANDoff:			
CHANnel *n*	1 to 1023	7-41	Sets Channel for Handoff.
CHANnel?		7-41	Returns Channel for Handoff setting.
LOCAL *n*	0 to 31	7-41	Sets Local Control field.
LOCAL?		7-41	Returns Local Control field setting.
MEM *n*	1 or 0	7-41	Sets Message Encryption Mode.
MEM?		7-41	Returns Message Encryption Mode setting.
MT *n*	0 to 31	7-41	Sets Message Type field.
MT?		7-41	Returns Message Type field setting.
ORDER:			
ALERT		7-38	Sends Alert message.
ALERTWinfo		7-38	Sends Alert with Information message.
AUDIT		7-38	Sends Audit message.
BSCHALCON		7-38	Sends Base Station Challenge Confirmation message.
CALLMODEACK		7-38	Sends Call Mode Acknowledgment message.
DISDTMF		7-38	Sends Disable DTMF message.
DISMEM		7-38	Sends Disable Message Encryption Mode message.
ENAMEM		7-38	Sends Enable Message Encryption Mode message.
FLASHWinfo		7-38	Sends Flash with Information message.
HANDoff		7-38	Sends Handoff message.
LC		7-38	Sends Local Control message.
MAINTenance		7-38	Sends Maintenance message.
MSGWTG		7-38	Sends Message Waiting message.
PAGE		7-38	Sends Page message.
PU		7-39	Sends Parameter Update message.
PWRLVL		7-39	Sends Power Level message.
RELease		7-39	Sends Release message.
SALERT		7-39	Sends Stop Alert message.
SLOT1		7-39	Sends Slot message (assigns DTC Timeslot 1).
SLOT2		7-39	Sends Slot message (assigns DTC Timeslot 2).
SLOT3		7-39	Sends Slot message (assigns DTC Timeslot 3).
SNDAddr		7-39	Sends Send Called Address message.
SNRreq		7-39	Sends Serial Number Request message.
SSDUP		7-39	Sends Shared Secret Data Update message.
UCHAL		7-39	Sends Unique Challenge message.
PM *n*	1 or 0	7-41	Sets Privacy Mode bit.
PM?		7-41	Returns Privacy Mode setting.
PSCC *n*	0 to 2	7-41	Sets Present SAT Color Code.
PSCC?		7-41	Returns Present SAT Color Code setting.
PWRLVL *n*	0 to 7	7-41	Sets Power Level requested of Mobile Station.
PWRLVL?		7-41	Returns requested Power Level setting.
RANDSSD *"n"*	0 to FFFFFFFFFFFFFFF	7-41	Sets RANDSSD 56 bit value in hexadecimal (string).
RANDSSD?		7-42	Returns RANDSSD 56 bit value setting.
RANDU *n*	0 to 16777215	7-42	Sets 24 bit Random Number in UCHAL message.
RANDU?		7-42	Returns UCHAL 24 bit Random Number setting.
SBI *n*	0 to 3	7-42	Sets Shortened Burst Indicator.
SBI?		7-42	Returns Shortened Burst Indicator setting.
SCC *n*	0 to 2	7-42	Sets Supervisory Audio Tone Color Code.
SCC?		7-42	Returns Supervisory Audio Tone Color Code setting.

COMMAND	RANGE	PAGE	DESCRIPTION
CSS:			
FVC:			
SIGNAL:			
CADENCE *n*	0 to 63	7-42	Sets on, off pattern of Alert tone.
CADENCE?		7-42	Returns Alert tone on, off pattern setting.
PITCH *n*	0 to 3	7-42	Sets Alert tone pitch.
PITCH?		7-42	Returns Alert tone pitch setting.
TA *n*	0 to 31	7-42	Sets Time Alignment.
TA?		7-42	Returns Time Alignment setting.
VMAC *n*	0 to 7	7-42	Sets Voice Mobile Attenuation Code.
VMAC?		7-42	Returns Voice Mobile Attenuation Code setting.
GLACT:			
ACTion:			
ACCess *n*	1 or 0	7-53	Enables or disables Access Attempt Parameters msg.
ACCess?		7-53	Returns Access Attempt Parameters message Enable setting.
BIS *n*	1 or 0	7-53	Enables or disables Access Type Parameters message.
BIS?		7-53	Returns Access Type Parameters message Enable setting.
LOCAID *n*	1 or 0	7-53	Enables or disables Location Area message.
LOCAID?		7-53	Returns Location Area message Enable setting.
LOCAL1 *n*	1 or 0	7-53	Enables or disables Local Control 1 message.
LOCAL1?		7-53	Returns Local Control 1 message Enable setting.
LOCAL2 *n*	1 or 0	7-53	Enables or disables Local Control 2 message.
LOCAL2?		7-54	Returns Local Control 2 message Enable setting.
NEWACC *n*	1 or 0	7-54	Enables or disables New Access Channel Set message.
NEWACC?		7-54	Returns New Access Channel Set message Enable setting.
OLC *n*	1 or 0	7-54	Enables or disables Overload Control message.
OLC?		7-54	Returns Overload Control message Enable setting.
RANDA *n*	1 or 0	7-54	Enables or disables Random Challenge A message.
RANDA?		7-54	Returns Random Challenge A message Enable setting.
RANDB *n*	1 or 0	7-54	Enables or disables Random Challenge B message.
RANDB?		7-54	Returns Random Challenge B message Enable setting.
REGINCR *n*	1 or 0	7-54	Enables or disables Registration Increment message.
REGINCR?		7-54	Returns Registration Increment message Enable setting.
RESCAN *n*	1 or 0	7-54	Enables or disables Rescan message.
RESCAN?		7-54	Returns Rescan message Enable setting.
BIS *n*	1 or 0	7-54	Sets Busy-Idle Status bit.
BIS?		7-54	Returns Busy-Idle Status bit setting.
LOCAID *n*	0 to 4095	7-54	Sets Cell Site Location Area Identification.
LOCAID?		7-55	Returns Cell Site Location Area Identification setting.
LOCALcntl *n*	0 to 65535	7-55	Sets Local Control bits.
LOCALcntl?		7-55	Returns value set for Local Control bits.
LREG *n*	1 or 0	7-55	Sets Local Area Identification Registration bit.
LREG?		7-55	Returns Local Area Identification Registration bit.
MAXBusy:			
OTHer *n*	0 to 15	7-55	Sets Max # of Busy incidents (Non-Page responses).
OTHer?		7-55	Returns Max # of Busy incidents (Non-Page responses).
PGR *n*	0 to 15	7-55	Sets Max # of Busy incidents (Page responses).
PGR?		7-55	Returns Max # of Busy incidents (Page responses).
MAXSztr:			
OTHer *n*	0 to 15	7-55	Sets Max # of Seizure Tries (Non-Page responses).
OTHer?		7-55	Returns Max # of Seizure Tries (Non-Page responses).
PGR *n*	0 to 15	7-55	Sets Max # of Seizure Tries (Page responses).
PGR?		7-55	Returns Max # of Seizure Tries (Page responses).
NEWACC *n*	0 to 2047	7-56	Sets New Access Channel starting point.
NEWACC?		7-56	Returns New Access Channel starting point setting.
OLC *n*	0 to 32767	7-56	Sets Overhead Class field.
OLC?		7-56	Returns Overhead Class field setting.
PDREG *n*	1 or 0	7-56	Sets Power Down Registration bit.
PDREG?		7-56	Returns Power Down Registration bit setting.
PUREG *n*	1 or 0	7-56	Sets Power Up Registration bit.
PUREG?		7-56	Returns Power Up Registration bit setting.
RAND1_A *n*	0 to 32767	7-56	Sets value of 16 most significant bits of RAND.
RAND1_A?		7-56	Returns value set for 16 most significant bits of RAND.

COMMAND	RANGE	PAGE	DESCRIPTION
CSS:			
GLACT:			
RAND1_B *n*	0 to 32767	7-56	Sets value of 16 least significant bits of RAND.
RAND1_B?		7-56	Returns value set for 16 least significant bits of RAND.
REGINCR *n*	0 to 4095	7-56	Sets Registration Increment field.
REGINCR?		7-56	Returns Registration Increment field setting.
REPEAT:			
OFF		7-53	Sets to send Global Action message once.
ON		7-53	Sets to send Global Action message continuously.
SEND		7-53	Starts sending Global Action message.
STOP		7-53	Stops sending Global Action message.
MSCM:			
AUTHBS *n*	0 to 262143	7-58	Sets AUTHBS value.
AUTHBS?		7-58	Returns AUTHBS value setting.
CHAN *n*	1 to 1023	7-58	Selects RF Channel.
CHAN?		7-58	Returns RF Channel setting.
CHANPos *n,x*	0 to 5, 0 to 127	7-58	Sets selected Channel Position field sent in Directed Retry message.
CHANPos? *n*	0 to 5	7-58	Returns selected Channel Position field setting.
DMAC *n*	0 to 10	7-58	Sets Digital Mobile Attenuation Code.
DMAC?		7-58	Returns Digital Mobile Attenuation Code setting.
DVCC *n*	1 to 255	7-58	Sets Digital Verification Color Code.
DVCC?		7-58	Returns Digital Verification Color Code setting.
EF *n*	1 or 0	7-59	Sets Extended Protocol Forward Channel Indicator.
EF?		7-59	Returns Extended Protocol Forward Channel Indicator setting.
LOCAL *n*	0 to 31	7-59	Sets Local Control field.
LOCAL?		7-59	Returns Local Control field setting.
MEM *n*	1 or 0	7-59	Sets Message Encryption Mode.
MEM?		7-59	Returns Message Encryption Mode setting.
MIN *"n"*		7-59	Selects Mobile Identification Number (string).
MIN?		7-59	Returns Mobile Identification Number string setting.
ORDER:			
A_ALERT		7-57	Selects Abbreviated Alert message.
AUDIT		7-57	Selects Audit message.
BSCHALCON		7-57	Selects Base Station Challenge Confirmation message.
DIR_RTRY		7-57	Selects Directed-Retry message.
INTRCPT		7-57	Selects Intercept message.
LC		7-57	Selects Local Control message.
MSG_WTG		7-57	Selects Message Waiting message.
REG_CNF		7-57	Selects Registration Confirmation message.
RELease		7-57	Selects Release message.
REORDER		7-57	Selects Reorder message.
SLOT1		7-58	Selects Digital Channel Assignment to Timeslot 1 message.
SLOT2		7-58	Selects Digital Channel Assignment to Timeslot 2 message.
SLOT3		7-58	Selects Digital Channel Assignment to Timeslot 3 message.
SSD_UP		7-58	Selects Shared Secret Data Update message.
UCHAL		7-58	Selects Unique Challenge message.
VC_DES		7-58	Selects Voice Channel Designation message.
ORDQ *n*	0 to 7	7-59	Sets Order Qualifier field.
ORDQ?		7-59	Returns Order Qualifier field setting.
PM *n*	1 or 0	7-59	Sets Privacy Mode Indicator bit.
PM?		7-59	Returns Privacy Mode Indicator bit setting.
RANDSSD1 *n*	0 to 16777215	7-59	Sets value of most significant bits of RANDSSD.
RANDSSD1?		7-59	Returns value of most significant bits of RANDSSD.
RANDSSD2 *n*	0 to 16777215	7-59	Sets value of bits 8 through 31 of RANDSSD.
RANDSSD2?		7-60	Returns value of bits 8 through 31 of RANDSSD.
RANDSSD3 *n*	0 to 255	7-60	Sets value of least significant bits of RANDSSD.
RANDSSD3?		7-60	Returns value of least significant bits of RANDSSD.
RANDU *n*	0 to 16777215	7-60	Sets 24 bit Random Number in UCHAL message.
RANDU?		7-60	Returns UCHAL message Random Number setting.

COMMAND	RANGE	PAGE	DESCRIPTION
CSS:			
MSCM:			
REPEAT:			
OFF		7-57	Sets to send Mobile Station Control message once.
ON		7-57	Sets to send Mobile Station Control message continuously.
SBI *n*	0 to 3	7-60	Sets Shortened Burst Indicator.
SBI?		7-60	Returns Shortened Burst Indicator setting.
SCC *n*	0 to 2	7-60	Sets Supervisory Audio Tone Color Code.
SCC?		7-60	Returns Supervisory Audio Tone Color Code setting.
SEND		7-57	Starts sending Mobile Station Control message.
STOP		7-57	Stops sending Mobile Station Control message.
VMAC *n*	0 to 7	7-60	Sets Voice Mobile Attenuation Code.
VMAC?		7-60	Returns Voice Mobile Attenuation Code setting.
RECC:			
STATus?		7-37	Returns access with Mobile Station status.
RFLvl *n*	-30.0 to -127.0	7-31	Sets RF Output Level, in dBm, of Overhead Message.
SETup		7-31	Sets FM/AM-1600CSA as when entering first Cell Site Simulation screen.
STARt		7-31	Starts Cell Site Simulation transmission.
STOP		7-31	Stops Cell Site Simulation transmission.

DEFINE COMMANDS

*DMC "*name, command*; ...		3-48	Defines a macro.
CONST *name,expression*		3-54	Declares and defines a constant.
DATA *name={expression,...}*		3-55	Declares and defines a data array.
FORMat *X*	BINary, DECimal, HEXadecimal or OCTal	3-64	Sets data format.
STRING *name,...,name*		3-84	Declares string variables.
STRING *name[index],name[index]*		3-84	Declares string arrays with *index*. ([*index*] mandatory, not optional.)
VAR *name,...,name*		3-86	Declares variables.
VAR *name[index],...,name[index]*		3-86	Declares arrays with *index*. ([*index*] mandatory, not optional.)

DISPLAY COMMANDS

BCOLOR *b*	0 to 15 or name	3-25	Sets background color.
BOX *f, x1, y1, x2, y2, c*	1 or 0, 0 to 639, 0 to 349, 0 to 639, 0 to 349, 0 to 15	3-51	Creates box on CRT display.
CENTER *element,x,y,pw*	string or text in quotes, 0 to 639, 0 to 349, 1 to (640-*x*)	3-31	Centers and displays text element in defined area.
CLS		3-54	Clears display screen or selected window of everything except active readings.
COLOR *f,c*	0 to 15 or name, 0 to 15 or name	3-25	Sets foreground and character contrast colors.
COLOR?		3-25	Returns current foreground color.
DRAW *x1, y1, x2, y2, c*	0 to 639, 0 to 349, 0 to 639, 0 to 349, 0 to 15	3-58	Creates line on CRT display.
ERASE:TEXT *x,y,pw*	0 to 639, 0 to 349, 1 to (640-*x*)	3-31	Erases defined CRT display area.
HEIGHT *n*	1 to 4	3-64	Sets height of text from 0.175 to 0.7 inches.
HPRINT *%0nl, expression,\ expression, ... ,expression,*		3-66	Prints expression on CRT display.
LJPRINT *value,x,y,pw*		3-31	Displays value, left justified, in defined area on CRT.
PIXEL		3-77	Displays point at current xy position on CRT.
PIXLEN? *value*		3-31	Returns number of pixels required to display value.
PRINT *%0nl, expression, ...*		3-79	Prints expression out CSA RS-232 Connector.
RJPRINT *value,x,y,pw*		3-31	Displays value, right justified, in defined area on CRT.
USER		3-26	Clears display screen or active window of everything.
XY *n, m*	0 to 639, 0 to 349	3-90	Sets xy position for **HPRINT** and **PIXEL** commands.
XYPRINT *x,y,%0nl, expression,\ expression, ... ,expression,*		3-91	Prints expression at selected xy position on CRT.

FORWARD DIGITAL TRAFFIC CHANNEL (FDTC) MONITOR COMMANDS

NOTE: Queries for received data, return -1 if data is not available or has already been read.

COMMAND	RANGE	PAGE	DESCRIPTION
FDTC:			
CHANnel *n*	1 to 1023	7-15	Selects Forward Digital Traffic Channel to monitor.
DVCC?		7-15	Returns Digital Verification Color Code.
FACCH:			
AMT?		7-15	Returns Message Type acknowledged.
AUTHBS?		7-15	Returns AUTHBS value.
CALLING:			
NUM?		7-16	Returns number of calling party.
PI?		7-16	Returns Calling Party Presentation Indicator.
PLANid?		7-16	Returns Calling Party Numbering Plan Identification.
SI?		7-16	Returns Calling Party Screening Indicator.
SPare?		7-16	Returns value of Calling Party Number spare bits.
TYpe?		7-16	Returns Calling Party Number Type.
CNPC?		7-16	Returns Calling Number Presentation Code.
DIC?		7-16	Returns Delay Interval Compensation.
DIGits? *n*	0 to 2	7-16	Returns selected digit set.
DMAC?		7-16	Returns Digital Mobile Attenuation Code.
DTX?		7-16	Returns Discontinuous Transmission.
HDVCC?		7-16	Returns Handoff Digital Verification Color Code.
LC?		7-16	Returns Local Control.
LDP?		7-16	Returns Last Decoded Parameter.
MEM?		7-16	Returns Message Encryption Mode.
MSGtype?		7-15	Returns Message Type.
NOMW?		7-16	Returns Number of Messages Waiting.
NV? *n*	0 to 5	7-17	Returns Number of Values for selected element.
PD?		7-17	Returns Protocol Discriminator.
PT? *n*	0 to 5	7-17	Returns Parameter Type for selected element.
RANDSSD1?		7-17	Returns value of 24 most significant bits of RANDSSD.
RANDSSD2?		7-17	Returns value of bits 8 through 31 of RANDSSD.
RANDU?		7-17	Returns RANDU value.
RATE?		7-17	Returns Channel Rate.
RFCHAN? *n*	0 to 11	7-17	Returns RF Channel for selected index.
RL? *n*	0 to 2	7-17	Returns Remaining Length in selected segment.
RN?		7-17	Returns Request Number.
SBI?		7-17	Returns Shortened Burst Indicator.
SIGnal?		7-17	Returns Signal field.
TA?		7-17	Returns Time Alignment.
TI?		7-17	Returns Timeslot Indicator. (0 is analog.)
VPM?		7-17	Returns Voice Privacy Mode.
IS54:			
CDVCC? *n*	0 to 99	7-21	Returns Coded Digital Verification Color Code from selected data word.
COUNt?		7-21	Returns number of data words received.
DATA? *n,x*	0 to 99, 1 to 65	7-21	Returns character data from selected data word.
SACCH? *n*	0 to 99	7-21	Returns 12 bits of SACCH message.
START		7-21	Starts IS-54 timeslot raw data Capture.
STOP		7-21	Stops IS-54 timeslot raw data Capture.
SYNC? *n*	0 to 99	7-21	Returns Synchronization bits from selected data word.
TIME? *n*	0 to 99	7-21	Returns Time Stamp in ms of selected data word.
R0?		7-15	Returns VSELP frame energy value.
RAW:			
CF? *n*	0 to 99	7-20	Returns Continuation Flag bit from selected data word.
COUNt?		7-20	Returns number of data words received.
DEPTH *n*	1 to 100	7-20	Selects size (number of data words) of data buffer.
DVCC? *n*	0 to 99	7-21	Returns Digital Verification Color Code from selected data word.
MESSage? *n,x*	0 to 99,0 to 5	7-21	Returns selected message byte from selected data word.
RSVD? *n*	0 to 99	7-21	Returns Reserved bits from selected data word.
SELect:			
FACCH		7-20	Selects Fast Associated Control Channel for raw data.
SACCH		7-20	Selects Slow Associated Control Channel for raw data.
START		7-20	Starts raw data Capture.

COMMAND	RANGE	PAGE	DESCRIPTION
FDTC:			
RAW:			
STOP		7-20	Stops raw data Capture.
TIME? *n*	0 to 99	7-21	Returns Time Stamp in ms of selected data word.
SACCH:			
AMT?		7-18	Returns Message Type acknowledged.
AUTHBS?		7-18	Returns AUTHBS value.
CALLING:			
NUM?		7-18	Returns number of calling party.
PI?		7-18	Returns Calling Party Presentation Indicator.
PLANid?		7-18	Returns Calling Party Numbering Plan Identification.
SI?		7-18	Returns Calling Party Screening Indicator.
SPare?		7-18	Returns value of Calling Party Number spare bits.
TYpe?		7-18	Returns Calling Party Number Type.
CNPC?		7-18	Returns Calling Number Presentation Code.
DIC?		7-18	Returns Delay Interval Compensation.
DIGits? *n*	0 to 2	7-18	Returns selected digit set.
DMAC?		7-18	Returns Digital Mobile Attenuation Code.
DTX?		7-18	Returns Discontinuous Transmission.
HDVCC?		7-19	Returns Handoff Digital Verification Color Code.
LC?		7-19	Returns Local Control.
LDP?		7-19	Returns Last Decoded Parameter.
MEM?		7-19	Returns Message Encryption Mode.
MSGtype?		7-18	Returns Message Type.
NOMW?		7-19	Returns Number of Messages Waiting.
NV? *n*	0 to 5	7-19	Returns Number of Values for selected element.
PD?		7-19	Returns Protocol Discriminator.
PT? *n*	0 to 5	7-19	Returns Parameter Type for selected element.
RANDSSD1?		7-19	Returns value of 24 most significant bits of RANDSSD.
RANDSSD2?		7-19	Returns value of bits 8 through 31 of RANDSSD.
RANDU?		7-19	Returns RANDU value.
RATE?		7-19	Returns Channel Rate.
RFCHAN? *n*	0 to 11	7-19	Returns RF Channel for selected index.
RL? *n*	0 to 2	7-19	Returns Remaining Length octets in selected segment.
RN?		7-19	Returns Request Number.
SBI?		7-20	Returns Shortened Burst Indicator.
SIGnal?		7-20	Returns Signal Field.
TA?		7-20	Returns Time Alignment.
TI?		7-20	Returns Timeslot Indicator. (0 is analog.)
VPM?		7-20	Returns Voice Privacy Mode.
SETup		7-15	Sets FM/AM-1600CSA as when entering Forward Digital Traffic Channel screen.
SLOT *n*	1 to 3	7-15	Selects Digital Traffic Channel Timeslot.
STARt		7-15	Starts decoding FDTC data.
STOP		7-15	Stops decoding FDTC data.

FORWARD CONTROL CHANNEL (FOCC) MONITOR COMMANDS

NOTE: Queries for received data, return -1 if data is not available or has already been read.

FOCC:			
ACT?		7-5	Returns Global Action value.
AUTH?		7-5	Returns Authentication bit.
AUTHBS?		7-5	Returns AUTHBS value.
BIDle?		7-5	Returns Busy-Idle bit.
BIS?		7-5	Returns Busy-Idle Status bit.
CAPTure:			
A_ALERT		7-3	Selects Abbreviated Alert message for Capture.
AUDIT		7-3	Selects Audit message for Capture.
AUT_REG		7-3	Selects Autonomous Registration Confirmation message for Capture.
BSCHALCON		7-3	Selects Base Station Challenge Confirmation Message for Capture.
CLEAR		7-3	Restarts Capture Function.
DIR_RTRY		7-3	Selects Directed-Retry message for Capture.
INTRCPT		7-3	Selects Intercept message for Capture.
LC		7-3	Selects Local Control message for Capture.

COMMAND	RANGE	PAGE	DESCRIPTION
FOCC:			
CAPTure:			
MIN *"n"*		7-4	Selects Mobile Identification Number to Capture.
MIN?		7-4	Returns MIN currently selected for Capture.
MODE?		7-3	Returns current Capture Mode: 0-NONE, 1-ORDER, 2-MIN or 3-BOTH (MIN and ORDER).
MSG_WTG		7-3	Selects Message Waiting message for Capture.
N_AUT_REG		7-4	Selects Non-autonomous Registration Confirmation message for Capture.
ORDer?		7-4	Returns name of order currently selected for Capture.
PAGE		7-4	Selects Page for Capture.
RELease		7-4	Selects Release message for Capture.
REORDER		7-4	Selects Reorder message for Capture.
SELect:			
BOTH		7-3	Sets Capture Mode to BOTH (MIN and ORDER).
MIN		7-3	Sets Capture Mode to MIN.
NONE		7-3	Sets Capture Mode to NONE.
ORDER		7-3	Sets Capture Mode to ORDER.
SLOT_1		7-4	Selects Slot 1 Handoff message for Capture.
SLOT_2		7-4	Selects Slot 2 Handoff message for Capture.
SLOT_3		7-4	Selects Slot 3 Handoff message for Capture.
SSD_UPdate		7-4	Selects Shared Secret Data Update for Capture.
UCHAL		7-4	Selects Unique Challenge message for Capture.
VC_DES		7-4	Selects Voice Channel Designation message for Capture.
CAPTure?		7-2	Returns Capture condition (1 [occurred] or 0).
CHANnel *n*	1 to 1023	7-2	Selects Forward Control Channel to monitor.
CHAN?		7-5	Returns Channel Number field.
CHANPOS1?		7-5	Returns Channel Position 1.
CHANPOS2?		7-5	Returns Channel Position 2.
CHANPOS3?		7-5	Returns Channel Position 3.
CHANPOS4?		7-5	Returns Channel Position 4.
CHANPOS5?		7-5	Returns Channel Position 5.
CHANPOS6?		7-5	Returns Channel Position 6.
CMAC?		7-5	Returns Control Mobile Attenuation Code.
CMAX_1?		7-5	Returns number of Access Channels minus one.
CPA?		7-6	Returns Combined Paging/Access.
DCC?		7-6	Returns Digital Color Code.
DMAC?		7-6	Returns Digital Mobile Attenuation Code.
DTX?		7-6	Returns Discontinuous Transmission.
DVCC?		7-6	Returns Digital Verification Color Code.
E?		7-6	Returns Extended Address.
EF?		7-6	Returns Extended Protocol Forward Channel Indicator.
END?		7-6	Returns End Indication.
EP?		7-6	Returns Extended Protocol.
LOC CONTRL?		7-6	Returns Local Control Message.
LOCAID?		7-6	Returns Location Area Identity.
LOCAL_MT?		7-6	Returns Local Control/Message Type.
LREG?		7-6	Returns Location Registration.
MBUSY:			
OTH?		7-6	Returns Max # of Busy incidents (Non-Page responses).
PGR?		7-6	Returns Max # of Busy incidents (Page responses).
MEM?		7-6	Returns Message Encryption Mode.
MIN?		7-7	Returns Mobile Identification Number.
MSZTR:			
OTH?		7-7	Returns Max # of Seizure Tries (Non-Page responses).
PGR?		7-7	Returns Max # of Seizure Tries (Page responses).
N_1?		7-7	Returns number of Paging Channels minus one.
NAWC?		7-7	Returns Number of Additional Words Coming.
NEWACC?		7-7	Returns New Access Channel Starting Point.
OLC?		7-7	Returns Overload Control Class.
ORDER?		7-7	Returns Order.
ORDERCD?		7-7	Returns Order Code.
ORDQ?		7-7	Returns Order Qualifier.
PCI?		7-7	Returns Protocol Capability Indicator.
PDREG?		7-7	Returns Power Down Registration.
PM?		7-7	Returns Privacy Mode.

COMMAND	RANGE	PAGE	DESCRIPTION
FOCC:			
PUREG?		7-7	Returns Power Up Registration.
RANDSSD_1?		7-7	Returns value of 24 most significant bits of RANDSSD.
RANDSSD_2?		7-7	Returns value of bits 8 through 31 of RANDSSD.
RANDSSD_3?		7-8	Returns value of 8 least significant bits of RANDSSD.
RANDU?		7-8	Returns 24 bit Random Number UCHAL message.
RAND1_A?		7-8	Returns 16 most significant bits of RAND Value.
RAND1_B?		7-8	Returns 16 least significant bits of RAND Value.
RAW:			
A:			
CHECK? *n*	0 to 99	7-11	Returns CRC Check for selected Stream A data word.
DATA? *n*	0 to 99	7-11	Returns Raw Data word from Stream A.
PARITY? *n*	0 to 99	7-11	Returns Parity for selected data word from Stream A.
B:			
CHECK? *n*	0 to 99	7-11	Returns CRC Check for selected Stream B data word.
DATA? *n*	0 to 99	7-11	Returns Raw Data word from Stream B.
PARITY *n*	0 to 99	7-11	Returns Parity for selected data word from Stream B.
B_I? *n*	0 to 99	7-11	Returns Busy-Idle bit for selected data word.
CAPTure:			
A_ALERT		7-9	Selects Abbreviated Alert Message for Capture.
AUDIT		7-9	Selects Audit Message for Capture.
AUT_REG		7-9	Selects Autonomous Registration Confirmation message for Capture.
BSCHALCON		7-9	Selects Base Station Challenge Confirmation Message for Capture.
DIR_RTRY		7-9	Selects Directed-Retry message for Capture.
INDex?		7-10	Returns position of Captured Order in buffer.
INTRCPT		7-9	Selects Intercept message for Capture.
LC		7-9	Selects Local Control Message for Capture.
MSG_WTG		7-10	Selects Message Waiting message for Capture.
N_AUT_REG		7-10	Selects Non-Autonomous Registration Confirmation message for Capture.
NONE		7-9	Sets Capture Mode to NONE (no message selected).
ORDer?		7-10	Returns name of order currently selected for Capture.
PAGE		7-10	Selects Page message for Capture.
RELease		7-10	Selects Release message for Capture.
REORDER		7-10	Selects Reorder message for Capture.
SLOT_1		7-10	Selects Slot 1 Handoff message for Capture.
SLOT_2		7-10	Selects Slot 2 Handoff message for Capture.
SLOT_3		7-10	Selects Slot 3 Handoff message for Capture.
SSD_UPdate		7-10	Selects Shared Secret Data Update for Capture.
UCHAL		7-10	Selects Unique Challenge message for Capture.
VC_DES		7-10	Selects Voice Channel Designation message for Capture.
CAPTure?		7-11	Returns Capture condition (1 [occurred] or 0).
FULL?		7-11	Returns buffer status (1 [full] or 0).
STARt		7-10	Starts raw data Capture.
STOP		7-10	Stops raw data Capture.
TRIGger *n*	0 to 4	7-10	Selects position of Captured Order in buffer.
TS? *n*	0 to 99	7-11	Returns Time Stamp in ms of selected data word.
WORD:			
A		7-9	Selects raw data from Stream A to monitor.
B		7-9	Selects raw data from Stream B to monitor.
BOTH		7-9	Selects raw data from Streams A and B to monitor.
RCF?		7-8	Returns Read Control-Filler.
REGH?		7-8	Returns Registration for Home Mobile Stations.
REGID?		7-8	Returns Registration Identification.
REGINCR?		7-8	Returns Registration Increment.
REGR?		7-8	Returns Registration for Roaming Mobile Stations.
REMote:			
STARt		7-2	Redirects Forward Control Channel data out RS-232 Connector.
STOP		7-2	Stops sending Forward Control Channel data out RS-232 Connector.
S?		7-8	Returns Serial Number bit value.
SCC?		7-8	Returns Supervisory Audio Tone Color Code.
SDCC1?		7-8	Returns Supplementary Digital Color Code 1.

COMMAND	RANGE	PAGE	DESCRIPTION
FOCC:			
SDCC2?		7-8	Returns Supplementary Digital Color Code 2.
SETup		7-2	Sets FM/AM-1600CSA as when entering Forward Control Channel Screen.
SID?		7-8	Returns System Identification Number.
STARt		7-2	Starts decoding FOCC data.
STREAM:			
A		7-2	Selects data from Stream A.
B		7-2	Selects data from Stream B.
STOP		7-2	Stops decoding FOCC data.
VMAC?		7-8	Returns Voice Mobile Attenuation Code.
WFOM?		7-8	Returns Wait For Overhead Message.
WORD:			
A		7-2	Selects Stream A to monitor.
B		7-2	Selects Stream B to monitor.
BOTH		7-2	Selects Streams A and B to monitor.

FORWARD VOICE CHANNEL (FVC) MONITOR COMMANDS

FVC:			
AUTHBS?		7-12	Returns Output Response of Base Station Authentication Algorithm.
CHANnel *n*	1 to 1023	7-12	Selects Forward Voice Channel to monitor.
CHAN?		7-12	Returns Channel Number field.
CHAR1?		7-12	Returns first Characters in Called or Calling Party Number message.
CHAR2?		7-12	Returns last Characters in Called or Calling Party message.
CPN_RL?		7-13	Returns Calling Party Number Remaining Length.
DMAC?		7-13	Returns Digital Mobile Attenuation Code.
DVCC?		7-13	Returns Digital Verification Color Code.
EF?		7-13	Returns Extended Protocol Forward Channel Indicator.
LOCAL_MT?		7-13	Returns Local Control/Message Type.
MEM?		7-13	Returns Message Encryption Mode.
ORDER?		7-13	Returns Order.
ORDERCD?		7-13	Returns Order Code.
ORDQ?		7-13	Returns Order Qualifier.
PI?		7-13	Returns Calling Party Number Presentation Indicator.
PM?		7-13	Returns Privacy Mode.
PSCC?		7-13	Returns Present SAT Color Code.
PWRL?		7-13	Returns requested Power Level.
RANDSSD1?		7-13	Returns value of 24 most significant bits of RANDSSD.
RANDSSD2?		7-13	Returns value of bits 8 through 31 of RANDSSD.
RANDSSD3?		7-13	Returns value of 8 least significant bits of RANDSSD.
RANDU?		7-13	Returns RANDU value.
RAW:			
CHECK? *n*	0 to 99	7-14	Returns CRC Check for selected data word (0 [good]).
COUNT? *n*	0 to 99	7-14	Returns number of data words received.
DATA? *n*	0 to 99	7-14	Returns raw data word (28 bits).
DEPTH *n*	1 to 100	7-14	Selects size of data buffer.
PARITY? *n*	0 to 99	7-14	Returns Parity for selected data word.
STARt		7-14	Starts raw data Capture.
STOP		7-14	Stops raw data Capture.
TS? *n*	0 to 99	7-14	Returns Time Stamp of selected data word in seconds from 0 data word.
RL_W?		7-13	Returns Remaining Length in number of Words.
SBI?		7-14	Returns Shortened Burst Indicator.
SCC *n*	5955 to 6044	7-12	Specifies SCC with corresponding SAT frequency.
SCC?		7-14	Returns Supervisory Audio Tone Color Code.
SETup		7-12	Sets FM/AM-1600CSA as when entering Forward Voice Channel Screen.
SI?		7-14	Returns Calling Party Screening Indicator.
SIGNAL?		7-14	Returns Signal field.
STARt		7-12	Starts decoding Forward Voice Channel.
STOP		7-12	Stops decoding Forward Voice Channel.
TA?		7-14	Returns Time Alignment.
VMAC?		7-14	Returns Voice Mobile Attenuation Code.

COMMAND	RANGE	PAGE	DESCRIPTION
GENERIC MEASURE COMMANDS			
MEASure:			
SAT?		7-62	Returns Supervisory Audio Tone frequency in Hz.
ST?		7-62	Returns Signal Tone frequency in Hz.
GPIB COMMANDS			
GPIB:ADDRess *a*	0 to 31	3-36	Sets GPIB address of FM/AM-1600CSA.
GPIB:MASK *n*	0 to 255	3-36	Sets SRQ interrupt mask.
GPIB:SRQ *n*	0 to 255	3-36	Sets user defined Status Byte to trigger SRQ.
HOST COMMANDS			
HOST "*string*"		7-1	Sends 1600S command as string.
HOST? "*string*?"		7-1	Sends 1600S query as string.
IEEE 488.2 COMPLIANCE COMMANDS			
*CLS		3-46	Sets user defined Status Byte to zero.
*DMC "*name, command*; ...		3-48	Defines a macro.
*IDN?		3-49	Returns FM/AM-1600CSA identification.
*PMC		3-49	Deletes all macros (except predefined) and declared variables from FM/AM-1600CSA.
KEY COMMANDS			
KEY		3-32	Returns keycode of next control terminal key pressed.
KEY?		3-32	Returns 1 if control terminal key is pressed or 0.
KEYPAD COMMAND			
KEYPAD:LABel *n*, "*Label*"	1 to 6	3-74	Creates *Label* for Soft Function Key F*n*.
MACRO FUNCTION COMMANDS			
BEGIN		3-50	Allows macro to have multiple lines.
CASE variable		3-52	Provides predetermined decision point.
OF *value1: command*			
OF *value2: command*			
OTHERWISE: *command*			
DELAY *t*		3-56	Provides time delay in ms.
DO *sequence* UNTIL *condition*		3-56	Performs set of commands.
END		3-59	Ends macro.
ENDCASE		3-52	Ends Case command.
EXEC *address* or EXEC &*name*		3-59	Executes a macro.
FLUSH		3-60	Returns pending responses.
FOR *variable=initial* TO *ending* STEP *step*		3-61	Performs set of commands repeatedly.
sequence			
NEXT *variable*			
FORGET "*name*"		3-63	Deletes macro called "*name*" from memory.
HFLUSH		3-65	Quickly sends **HOST** commands through SCSI.
IF *condition sequence*; ENDIF		3-68	Provides decision point.
IF *condition sequence*; ELIF *condition sequence*		3-69	Provides decision points.
ELIF *condition sequence*;ENDIF			
IF *condition sequence*		3-70	Provides decision point.
ELSE *condition sequence*;ENDIF			
INPUT *variable*		3-72	Allows data entry through RS-232 Connector during macro execution.
INTERP "*string;string;...;string*"		3-72	Executes commands, unknown when macro is defined.
RETURN		3-81	Returns value from function or returns macro execution to previous macro.
ROOM		3-82	Used with **PRINT** to return memory bytes available.
WHILE *condition sequence*;WEND		3-87	Performs set of commands repeatedly until conditional expression is false.

COMMAND	RANGE	PAGE	DESCRIPTION
MATHEMATICAL FUNCTION COMMANDS			
ABS(n)		3-13	Produces absolute value of n.
CALCULATE? *expression*		3-52	Calculates expression functions and returns result.
COS(n)		3-14	Produces cosine of n in radians.
EXP(n)		3-14	Produces e to the power of n.
FALSE,OFF		3-12	Produces 0.
FLOOR(n)		3-13	Produces truncated value of n.
LN(n)		3-13	Produces natural logarithm (base e) of n.
LOG(n)		3-13	Produces base 10 logarithm of n.
RAND x		3-12	Sets seed (starting point) to x for random generator.
RND(n)		3-12	Produces random integer between 0 and n.
SIGN(n)		3-14	Produces -1 if n<0, 0 if n=0 or 1 if n>0.
SIN(n)		3-14	Produces sine of n in radians.
SQR(n)		3-14	Produces positive square root of n.
TRUE,ON		3-12	Produces 1.
MASS MEMORY COMMANDS			
MMEMory:			
CATalog:			
ENTRY? n	0 to 512	7-64	Returns line n from JBox Flash Files Directory.
FREE?		7-64	Returns available file space in bytes.
USED?		7-64	Returns used file space in bytes.
CATalog?		7-64	Returns Flash Memory status.
DELete "f"		7-64	Deletes file with file name f.
INITialize		7-64	Erases all files stored in Flash Memory.
INITialize?		7-64	Returns 1 for file system initialized or 0 otherwise.
LOAD:			
CALibration "f"		7-64	Loads Calibration Data from Flash to CSA Memory.
MACRo "m","f"		7-64	Loads macros and variables stored as file name f from Flash to CSA memory, executing macro m.
STATe "f"		7-64	Loads CSA State stored as f from Flash Memory.
PACK		7-64	Frees memory space from deleted files.
STORe:			
CALibration "f"		7-64	Stores current Calibration Data as f in Flash Memory.
MACRo "m","f"		7-64	Stores currently loaded macros and variables as file name f in Flash Memory with designated macro m to activate from JBox Flash Files Directory.
STATe "f"		7-64	Stores current CSA settings as f in Flash Memory.
TYPE? "f"		7-64	Returns file type of file name f.
MODULATION ACCURACY COMMANDS			
MODacc:			
FDTC:			
CHANnel *n*	1 to 1023	7-62	Sets RF Channel.
COMPlete?		7-62	Returns test status (1 [complete] or 0).
EVM?		7-62	Returns RMS Error Vector Magnitude in percent.
FREQ_ERRor?		7-62	Returns Frequency Error in hertz.
IQ_OFFset?		7-62	Returns I/Q Offset in dB.
MAG_ERRor?		7-62	Returns RMS Magnitude Error in percent.
PHASE_ERRor?		7-62	Returns RMS Phase Error in degrees.
RUN?		7-62	Starts measurement and returns adjusted AGC value.
SETup		7-62	Sets FM/AM-1600CSA as when entering Modulation Accuracy screen.
MULTITASKING MACROS COMMANDS			
ACTIVATE "*name*"		3-50	Puts specified task in schedule queue.
SLEEP "*name*"		3-83	Removes specified task from schedule queue.
STOP		3-84	Stops all macros.
TASK "*name*"		3-85	Declares specified loaded macro to be a task.
TPAUSE		3-85	Passes command execution to next task in queue.
TSTOP		3-85	Stops and removes current task from queue.
WAKE "*name*"		3-86	Puts specified task in queue/executes from last stop.

COMMAND	RANGE	PAGE	DESCRIPTION
REVERSE DIGITAL TRAFFIC CHANNEL (RDTC) MONITOR COMMANDS			
RDTC:			
CHANnel *n*	1 to 1023	7-25	Selects Reverse Digital Traffic Channel to monitor.
FACCH:			
AMT?		7-26	Returns Acknowledge Message Type.
AUTHU?		7-26	Returns AUTHU value.
BER?		7-26	Returns Bit Error Rate value.
CALLED:			
NUM?		7-26	Returns number of Called Party.
PLANid?		7-26	Returns Called Party Numbering Plan Identification.
SPare?		7-26	Returns value of Called Party Number spare bit.
TYpe?		7-26	Returns Called Party Number Type.
CALLING:			
NUM?		7-26	Returns number of Calling Party.
PLANid?		7-26	Returns Calling Party Numbering Plan Identification.
SPare?		7-26	Returns value of Calling Party Number spare bits.
TYpe?		7-26	Returns Calling Party Number Type.
CM?		7-27	Returns Call Mode.
DIC?		7-27	Returns Delay Interval Compensation.
DIGits?		7-27	Returns up to 32 digits.
DMAC?		7-27	Returns Digital Mobile Attenuation Code.
DTX?		7-27	Returns Discontinuous Transmission.
ESN?		7-27	Returns Electronic Serial Number.
FI?		7-27	Returns Feature Indicator.
KF?		7-27	Returns Keypad Facility (32 digits).
LDP?		7-27	Returns Last Decoded Parameter.
MEM?		7-27	Returns Memory Encryption Mode.
MSGtype?		7-27	Returns Message Type received from Mobile Station.
NV?		7-27	Returns Numbers of Values bits.
PD?		7-27	Returns Protocol Discriminator.
PT?		7-27	Returns Parameter Types.
RANDBS?		7-27	Returns RANDBS value.
RFCHAN? *n*	0 to 11	7-28	Returns RF Channel for selected index.
RL?		7-28	Returns Remaining Length.
RN?		7-28	Returns Number of Requests.
RR?		7-28	Returns Release Reason.
RSSI? *n*	1 to 1023	7-28	Returns Received Signal Strength Indicator for selected RF Channel index.
RSSIC?		7-28	Returns Received Signal Strength Indicator for current RF Channel.
SSDUP?		7-28	Returns Shared Secret Data Update.
TA?		7-28	Returns Time Alignment.
TERMinf?		7-28	Returns Terminal Information of Mobile Station.
VPM?		7-28	Returns Voice Privacy Mode.
R0?		7-25	Returns VSELP frame energy value.
REMote:			
STARt		7-25	Redirects VSELP data out RS-232 Connector.
STOP		7-25	Stops VSELP data out RS-232 Connector.
SACCH:			
AMT?		7-28	Returns Acknowledge Message Type.
AUTHU?		7-28	Returns AUTHU value.
BER?		7-28	Returns Bit Error Rate value.
CALLED:			
NUM?		7-28	Returns number of Called Party.
PLANid?		7-28	Returns Called Party Numbering Plan Identification.
SPare?		7-29	Returns value of Called Party Number spare bit.
TYpe?		7-29	Returns Called Party Number Type.
CALLING:			
NUM?		7-29	Returns number of Calling Party.
PLANid?		7-29	Returns Calling Party Numbering Plan Identification.
SPare?		7-29	Returns value of Calling Party Number spare bits.
TYpe?		7-29	Returns Calling Party Number Type.
CM?		7-29	Returns Call Mode.
DIC?		7-29	Returns Delay Interval Compensation.
DIGits?		7-29	Returns up to 32 digits.
DMAC?		7-29	Returns Digital Mobile Attenuation Code.

COMMAND	RANGE	PAGE	DESCRIPTION
RDTC:			
SACCH:			
DTX?		7-29	Returns Discontinuous Transmission.
ESN?		7-29	Returns Electronic Serial Number.
FI?		7-29	Returns Feature Indicator.
KF?		7-29	Returns Keypad Facility (32 digits).
LDP?		7-29	Returns Last Decoded Parameter.
MEM?		7-29	Returns Memory Encryption Mode.
MSGtype?		7-30	Returns Message Type received from Mobile Station.
NV?		7-30	Returns Numbers of Values.
PD?		7-30	Returns Protocol Discriminator.
PT?		7-30	Returns Parameter Types.
RANDBS?		7-30	Returns RANDBS value.
RFCHAN?*n*	0 to 11	7-30	Returns RF Channel for selected index.
RL?		7-30	Returns Remaining Length.
RN?		7-30	Returns Number of Requests.
RR?		7-30	Returns Release Reason.
RSSI?*n*	0 to 11	7-30	Returns Received Signal Strength Indicator for selected RF Channel index.
RSSIC?		7-30	Returns Received Signal Strength Indicator for current RF Channel.
SSDUP?		7-30	Returns Shared Secret Data Update.
TA?		7-30	Returns Time Alignment.
TERMinf?		7-30	Returns Terminal Information of Mobile Station.
VPM?		7-30	Returns Voice Privacy Mode.
SETup		7-25	Sets FM/AM-1600CSA as when entering Reverse Digital Traffic Channel Screen.
SLOT *n*	1 to 3	7-25	Selects Timeslot .
STARt		7-25	Starts decoding RDTC data.
STOP		7-25	Stops decoding RDTC data.

REVERSE CONTROL CHANNEL (RECC) MONITOR COMMANDS

RECC:			
AUTHR?		7-22	Returns AUTHR value.
AUTHU?		7-22	Returns AUTHU value.
CHANnel *n*	1 to 1023	7-22	Selects Reverse Control Channel to monitor.
COUNt?		7-22	Returns modulo-64 count sent from Mobile Station.
DCC?		7-22	Returns Digital Color Code.
DIGITS1?		7-22	Returns up to 16 digits.
DIGITS2?		7-22	Returns up to 16 digits.
E?		7-22	Returns Extended Address.
EP?		7-22	Returns Extended Protocol.
ER?		7-22	Returns Extended Protocol Reverse Channel Indicator.
ESN?		7-22	Returns Electronic Serial Number.
LOCAL_MT?		7-23	Returns Local Control/Message Type.
LT?		7-23	Returns Last Try.
MIN?		7-23	Returns Mobile Identification Number.
MPCI?		7-23	Returns Mobile Protocol Capability Indicator.
ORDERCD?		7-23	Returns Order Type.
ORDQ?		7-23	Returns Order Qualifier.
RANDBS?		7-23	Returns RANDBS value.
RANDC?		7-23	Returns RANDC value.
S?		7-23	Returns Serial Number bit value.
SCM?		7-23	Returns Station Class Mark.
SDCC1?		7-23	Returns Supplementary Digital Color Code 1.
SDCC2?		7-23	Returns Supplementary Digital Color Code 2.
SETup		7-22	Sets FM/AM-1600CSA as when entering Reverse Control Channel Screen.
STARt		7-22	Starts decoding RECC data.
STOP		7-22	Stops decoding RECC data.
TORDer?		7-23	Returns combination of T and Order fields.

COMMAND	RANGE	PAGE	DESCRIPTION
REVERSE VOICE CHANNEL (RVC) MONITOR COMMANDS			
RVC:			
AUTHu?		7-24	Returns AUTHU value.
CHANnel *n*	1 to 1023	7-24	Selects Reverse Voice Channel to monitor.
DIGITS?		7-24	Returns up to 32 digits.
ESN?		7-24	Returns Electronic Serial Number.
LOCAL_MT?		7-24	Returns Local Control/Message Type.
ORDERCD?		7-24	Returns Order Type.
ORDQ?		7-24	Returns Order Qualifier.
RANDbs?		7-24	Returns RANDBS value.
SETup		7-24	Sets FM/AM-1600CSA as when entering Reverse Voice Channel Screen.
STARt		7-24	Starts decoding RVC data.
STOP		7-24	Stops decoding RVC data.
TORDer?		7-24	Returns combination of T and Order fields.
SOUND COMMAND			
SOUND *f,t*		3-83	Creates and sends audio tone to Test Set Speaker (frequency set in Hz and duration time set in ms).
STRING FUNCTION COMMANDS			
ASC($)		3-17	Returns ASCII value of first character in string $.
CHR(n)		3-16	Returns character equivalent of ASCII number n.
LEN($)		3-17	Returns length of string $ in number of characters.
PIXLEN($)		3-17	Returns length of string $ in pixels.
STR(n)		3-17	Returns string equivalent of number n.
STRPOS($1,$2)		3-18	Returns position of string $2 inside string $1.
TAB(n)		3-16	Returns blank string containing n number of spaces.
VAL($)		3-17	Returns numeric value represented by string $.
SYSTEM COMMANDS			
SYSTem:			
COMMunicate:			
SERial:			
BAUD *n*	300 to 38400	3-34	Sets baud rate.
BITS *n*	7 or 8	3-34	Sets number of data bits per word.
ECHo *b*	1 or 0	3-34	Enables or disables RCI echo.
PACe *type*	XON or NONE	3-34	Sets software handshake mode.
PARity *type*	NONE, ODD or EVEN MARK or SPACE	3-34	Sets parity check.
SBITs *n*	1 or 2	3-34	Sets number of stop bits per word.
DATE *year,month,day*	1992 to 9999, 1 to 12, 1 to 31	3-33	Sets date.
DATE?		3-33	Returns date.
ERRor?		3-33	Returns error number and message.
TIME *hour,minute,second*	0 to 23, 0 to 59, 0 to 59	3-33	Sets time.
TIME?		3-33	Returns time.
WINDOW COMMANDS			
CLS		3-54	Clears everything except active readings from selected window.
USER		3-26	Clears everything from active window.
WCLOSE *n*	1 to 16	3-87	Erases selected generated window(s) from CRT.
WINDOW?		3-88	Returns number of currently selected window.
WMOVE *x,y*	0 to 639,0 to 349	3-89	Moves currently selected window to selected new top left xy location.
WOPEN *c,x1,y1,x2,y2*	0 to 15,0 to 639, 0 to 349,0 to 639, 0 to 349	3-89	Creates windows.
WSEL *n,h*	0 to 15,0 or 1	3-90	Selects and hides or shows window.

SECTION 9 - CSA PROGRAM EXAMPLES

9-1 GENERAL

This Section provides some general guidelines and examples of remote operation and macro programs using FM/AM-1600CSA specific commands. All macros are executed remotely through RS-232.

NOTE: A file containing these macros is available on the IFR BBS (316-524-0270).

9-2 FORWARD CONTROL CHANNEL (FOCC)

9-2-1 MONITORING DECODED DATA

MACRO NAME: focc_page

SYNTAX: focc_page *n*
(*n*=FOCC channel number, 1 to 1023.)

PURPOSE: Look for Page orders on the FOCC.
If and when found, print out the MIN and time of day for each occurrence.
(Macro continues until key is pressed on the host RS-232 terminal.)

EXAMPLE: focc_page 332

MACRO:

```
*dmc "focc_page",begin          // Define macro named focc_page.
focc:setup                      // Setup for receiving FOCC.
focc:start                      // Start decoding FOCC data.
focc:chan $1                    // Set RF Channel to value entered with macro.
focc:word:both                  // Decode both Streams A and B.
do                              // Initiate do loop.
 tpause                         // Allow time for decoding.
 focc:stream:a                  // Select Stream A for response.
 $=focc:order?                  // Return order as string variable $.
 if $="PAGE"                    // Look for a Page order.
  focc:stream:a                 // Select Stream A for response.
  $=focc:min?                   // Return MIN as string variable $.
  print $," ",syst:time?        // Print MIN and time to RS-232 Connector.
 endif
 tpause                         // Allow time for decoding.
 focc:stream:b                  // Select Stream B for response.
 $=focc:order?                  // Return order as string variable $.
 if $="PAGE"                    // Look for a Page order.
  focc:stream:B                 // Select Stream B for response.
  $=focc:min?                   // Return MIN as string variable $.
  print $," ",syst:time?        // Print MIN and time to RS-232 Connector.
 endif
until key?                      // Do until key pressed on RS-232 terminal.
end                             // End macro focc_page.
```

9-2-2 MONITORING RAW DATA

MACRO NAME: focc_raw

SYNTAX: focc_raw *n*
(*n*=FOCC channel number, 1 to 1023.)

PURPOSE: Monitor FOCC until Page order is detected.
If and when found, fill raw data buffer (100 words) and place Page order data at the 50% point.

EXAMPLE: focc_raw 332

MACRO:

```
*dmc "focc_raw",begin          // Define macro named focc_raw.
focc:setup                     // Setup for receiving FOCC.
delay 1500
focc:chan $1                   // Set RF Channel to value entered with macro.
focc:raw:word:both             // Monitor Streams A and B.
focc:capt:sel:order            // Select ORDER to capture on.
focc:raw:capt:page             // Select Page as order for capture.
focc:raw:trig 2                // Select middle trigger position.
focc:raw:start                 // Start monitoring raw FOCC data.
print "WAITING FOR PAGE"       // Print out through RS-232 Connector.
do                             // Initiate do loop.
 tpause                        // Allow time to capture PAGE order.
until key? or focc:raw:capt?   // Wait for capture condition to end do loop.
print "CAPTURED PAGE"          // Indicate Page order was found.
do                             // Initiate do loop.
 tpause                        // Allow time to fill data buffer.
until key? or focc:raw:full?   // Wait for full data buffer before ending do loop.
print "CAPTURE BUFFER FULL"    // Indicate data buffer is full.
i=focc:raw:capt:index?         // Return Page order position in data buffer as i.
a=focc:raw:ts? i               // Return time stamp of Page order as a.
print "A Data  Parity  Check   B Data  Parity  Check   Busy/Idle  TS"
                               /* Print out header for raw data. */
for i=0 to 99                  // Print out raw data through RS-232 Connector.
 print %07h,focc:raw:a:data? i," ",
 print %03h,focc:raw:a:parity? i,"
 if (focc:raw:a:check? i)=0
  print "GOOD     ",
 else
  print "BAD      ",
 endif
 print %07h,focc:raw:b:data? i," ",
 print %03h,focc:raw:b:parity? i,"
 if (focc:raw:b:check? i)=0
  print "GOOD     ",
 else
  print "BAD      ",
 endif
 print %2d,focc:raw:b_i? i,"       ",
 print %6d,(focc:raw:ts? i)-a
next i
focc:raw:stop                  // Stops raw data capture.
end                            // End macro focc_raw.
```

9-3 FORWARD VOICE CHANNEL (FVC)

9-3-1 MONITORING DECODED DATA

Specify SAT Color Code (**FVC:SCC**) before requesting decoded data.

MACRO NAME: fvc_char

SYNTAX: fvc_char *n*
(*n*=FVC channel number, 1 to 1023.)

PURPOSE: Monitor FVC until Flash with Information order is detected.
If and when found, display the characters sent in the Flash with Infomation order message.
(Macro continues until key is pressed on the host RS-232 terminal.)

EXAMPLE: fvc_char 3

MACRO:

```
*dmc "fvc_char",begin          // Define macro named fvc_char.
fvc:setup                      // Setup for receiving FVC.
fvc:start                      // Start decoding FVC data.
fvc:chan $1                    // Set RF Channel to variable entered with macro.
fvc:scc :meas:sat?             // Select SCC to match SAT.
do                             // Initiate do loop.
 tpause                        // Allow time for FM/AM-1600CSA tasks to run.
 $=fvc:order?                  // Return last decoded order as string variable $.
 if $="FLASH W/INFO"           // Compare order with Flash with Information.
  delay 3000                   // Allow time for order to complete first 16
  $=fvc:char1?                 //  characters.
  if $!="-1"                   // When available, print first 16 characters
   print $,                    //  out RS-232 Connector.
  endif
  $=fvc:char2?                 // Receive second set of 16 characters.
  if $!="-1"                   // When available, print second 16 characters
   print $,                    //  out RS-232 Connector.
  else                         // If no second set of 16 characters received,
   print ""                    //  send carriage return out RS-232 Connector.
  endif
 endif
until key?                     // Do until key pressed on RS-232 terminal.
end                            // End macro fvc_char.
```

9-3-2 MONITORING RAW DATA

Set the FVC Capture Mode to NONE (**FVC:CAPT:SEL:NONE**) prior to initiating raw data queries.

MACRO NAME: fvc_raw

SYNTAX: fvc_raw *n*
(*n*=FVC channel number, 1 to 1023.)

PURPOSE: Collect raw data on the FVC.
Display five data words, starting with the first word as soon as received.

EXAMPLE: fvc_raw 3

MACRO:

```
*dmc "fvc_raw",begin          // Define macro named fvc_raw.
fvc:setup                     // Setup for receiving FVC.
fvc:chan $1                   // Set RF Channel to value entered with macro.
focc:capt:sel:none            // Eliminate FOCC capture interference.
fvc:raw:depth 5               // Set size of raw data buffer to 5 data words.
fvc:raw:start                 // Start collecting raw FVC data.
print "WAIT"                  // Indicate waiting out RS-232 Connector.
a=0                           // Set variable a to 0.
do                            // Initiate do loop.
 tpause                       // Allow time for FM/AM-1600CSA tasks to run.
 b=fvc:raw:count?             // Return number of words in raw data buffer.
 if a<b                       // When new data is available, print data
  for i = a to b-1            //  information out RS-232 Connector.
   print %07h,fvc:raw:data? i,"  "
   print %03h,fvc:raw:parity? i,"  "
   a=fvc:raw:check? i
   if (a)                     // If a=1 (true), then CRC check is bad.
    print "BAD   ",           // Indicate bad CRC check out RS-232 Connector.
   else                       // If a=0 (false), then CRC check is good.
    print "GOOD  ",           // Indicate good CRC check out RS-232 Connector.
   endif
   print %5d,fvc:raw:ts? i    // Indicate time of data word out RS-232 Connector.
  next i                      // Show data information for all 5 data words.
  a=b                         // Prevent displaying same data twice.
 endif
until key? or b=5             /* Do until key pressed on RS-232 terminal
                                   or raw data buffer is full. */
end                           // End macro fvc_raw.
```

9-4 FORWARD DIGITAL TRAFFIC CHANNEL (FDTC)

9-4-1 MONITORING DECODED DATA

MACRO NAME: fdtc_msg

SYNTAX: fdtc_msg *n,x*
(*n*=FDTC channel number, 1 to 1023. *x*=timeslot, 1 to 3)

PURPOSE: Monitor FDTC.
Display FACCH and SACCH message types, each with respective DVCC.
(Macro continues until key is pressed on the host RS-232 terminal.)

EXAMPLE: fdtc_msg 100,3

MACRO:

```
*dmc "fdtc_msg",begin          // Define macro named fdtc_msg.
fdtc:setup                     // Setup for receiving FDTC.
fdtc:chan $1                   // Set RF Channel to 1st value entered with macro.
fdtc:slot $2                   // Set Timeslot to 2nd value entered with macro.
fdtc:start                     // Start FDTC decoding.
do                             // Initiate do loop.
 tpause                        // Allow time for decoding.
 $=fdtc:facch:msg?             // Return FACCH message type as string variable.
 if $!="-1"                    // When available, indicate FACCH message type.
  print "Facch Message type is ",$
  print "DVCC is ",%1d,fdtc:dvcc?
 endif
 $=fdtc:sacch:msg?             // Return SACCH message type as string variable.
 if $!="-1"                    // When available, indicate SACCH message type.
  print "Facch Message type is ",$
  print "DVCC is ",%1d,fdtc:dvcc?
 endif
until key?                     // Do until key pressed on RS-232 terminal.
end                            // End macro fdtc_msg.
```

9-4-2 MONITORING RAW DATA

Set size of data buffer (**FDTC:RAW:DEPTH**) prior to using raw data queries. Limit selected data word to a range from 0 to the **FDTC:RAW:DEPTH** setting minus one, for raw data queries.

MACRO NAME: fdtc_raw

SYNTAX: fdtc_raw *n,x*
(*n*=FDTC channel number, 1 to 1023. *x*=timeslot, 1 to 3)

PURPOSE: Collect FACCH raw data from FDTC.
Display 20 FACCH raw data words, starting with the first word as soon as received.

EXAMPLE: fdtc_raw 100,3

MACRO:

```
*dmc "fdtc_raw",begin          // Define macro named fdtc_raw.
fdtc:setup                     // Setup for receiving FDTC.
fdtc:chan $1                   // Set RF Channel to 1st value entered with macro.
fdtc:slot $2                   // Set Timeslot to 2nd value entered with macro.
fdtc:raw:sel:facch             // Select FACCH raw data for data buffer.
fdtc:raw:depth 20              // Set size of raw data buffer to 20 data words.
fdtc:raw:start                 // Start collecting raw FDTC data.
print "WAIT"                   // Indicate waiting out RS-232 Connector.
a=0                            // Set variable a to 0.
do                             // Initiate do loop.
 tpause                        // Allow time for FM/AM-1600CSA tasks to run.
 b=fdtc:raw:count?             // Return number of words in raw data buffer.
 if a<b                        // When new data is available, print data
  for i = a to b-1             //  information out RS-232 Connector.
   print %2d,fdtc:raw:cf? i," "
   for j=0 to 5                // Set range of j to indicate message byte.
    print %02h,fdtc:raw:mess? i,j
   next j                      // Show data for all 5 bytes.
   print " ",%4d,fdtc:raw:dvcc? i, " ",
   print %3d,fdtc:raw:time? i
  next i                       // Show data information for all 20 data words.
  a=b                          // Prevent displaying same data twice.
 endif
until key? or b=20             /* Do until key pressed on RS-232 terminal
                                    or raw data buffer is full. */
end                            // End macro fdtc_raw.
```

9-4-3 MONITORING IS-54 RAW DATA

MACRO NAME: fdtc_is54

SYNTAX: fdtc_is54 *n,x*
(*n*=FDTC channel number, 1 to 1023. *x*=timeslot, 1 to 3)

PURPOSE: Collect FDTC raw data corresponding to the IS-54B timeslot.
Display 100 raw data words when raw data buffer is full.
(Macro continues until key is pressed on the host RS-232 terminal.)

EXAMPLE: fdtc_is54 100,3

MACRO:

```
*dmc "fdtc_is54",begin          // Define macro named fdtc_is54.
fdtc:setup                      // Setup for receiving FDTC.
fdtc:chan $1                    // Set RF Channel to 1st value entered with macro.
fdtc:slot $2                    // Set Timeslot to 2nd value entered with macro.
do                              // Initiate do loop.
 fdtc:is54:start                // Start collecting raw FDTC timeslot data.
 do                             // Initiate internal do loop
  tpause                        // Allow time to fill data buffer.
 until key? or fdtc:is54:count?=100
 for i=0 to 99
  print %07h,fdtc:is54:sync? i," ",
  print %03h,fdtc:is54:cdvcc? i," ",
  print %03h,fdtc:is54:sacch? i," ",
  print %5f,fdtc:is54:time? i
  for j=1 to 65
   print %01h,fdtc:is54:data? i,j,
  next j                        // Show all 65 characters.
  print
 next i                         // Show data information for all 100 data words.
until key?                      // Do until key pressed on RS-232 terminal.
end                             // End macro fdtc_is54.
```

9-5 REVERSE CONTROL CHANNEL (RECC)

MACRO NAME: recc_min

SYNTAX: recc_min *n*
(*n*=RECC channel number, 1 to 1023.)

PURPOSE: Monitor RECC.
Display the MIN of the Mobile Station.
(Macro continues until key is pressed on the host RS-232 terminal.)

EXAMPLE: recc_min 100

MACRO:

```
*dmc "recc_min",begin          // Define macro named recc_min.
recc:setup                     // Setup for receiving RECC.
recc:chan $1                   // Set RF Channel to value entered with macro.
recc:start                     // Start decoding RECC data.
do                             // Initiate do loop.
 tpause                        // Allow time for FM/AM-1600CSA tasks to run.
 $=recc:min?                   // Return MIN as string variable $.
 if $!="-1"                    // When available, indicate MIN out RS-232
  print "MIN received is ",$   // Connector.
 endif
until key?                     // Do until key pressed on RS-232 terminal.
end                            // End macro recc_min.
```

9-6 REVERSE VOICE CHANNEL (RVC)

MACRO NAME: rvc_order

SYNTAX: rvc_order *n*
(*n*=RVC channel number, 1 to 1023.)

PURPOSE: Monitor RVC.
Display name of each order received.
(Macro continues until key is pressed on the host RS-232 terminal.)

EXAMPLE: rvc_order 100

MACRO:

```
*dmc "rvc_order",begin         // Define macro named rvc_order.
rvc:setup                      // Setup for receiving RVC.
rvc:chan $1                    // Set RF Channel to value entered with macro.
rvc:start                      // Start decoding RVC data.
do                             // Initiate do loop.
 tpause                        // Allow time for FM/AM-1600CSA tasks to run.
 $=rvc:torder?                 // Return last decoded order as string variable $.
 if $!="-1"                    // When available, indicate last decoded order out
  print "ORDER RECEIVED ON RVC IS ",$
 endif                         // RS-232 Connector.
until key?                     // Do until key pressed on RS-232 terminal.
end                            // End macro rvc_order.
```

9-7 REVERSE DIGITAL TRAFFIC CHANNEL (RDTC)

MACRO NAME: rdtc_maho

SYNTAX: rdtc_maho *n,x*
(*n*=RDTC channel number, 1 to 1023. *x*=timeslot, 1 to 3)

PURPOSE: Monitor RDTC.
Display mobile assisted handoff information returned in Channel Quality messages.
(Macro continues until key is pressed on the host RS-232 terminal.)

EXAMPLE: rdtc_maho 100,2

MACRO:

```
*dmc "rdtc_maho",begin          // Define macro named rdtc_maho.
var i,reading
rdtc:setup                      // Setup for receiving RDTC.
rdtc:chan $1                    // Set RF Channel to 1st value entered with macro.
rdtc:slot $2                    // Set Timeslot to 2nd value entered with macro.
rdtc:start                      // Start decoding RDTC data.

print "Mobile Assisted Handoff Information"
print "Press any key to stop."

do                              // Initiate do loop.
 tpause                         // Allow time for background decoding.
 i=0                            // Set initial i value to 0.
 while i<12                     // Phone measures RSSI on up to 12 channels.
  reading=rdtc:sacch:rssi? i // Return RSSI of selected channels.
  if reading!=-1
   print "RSSI",%1d,i+1,"=",reading
  endif
  ++i
 wend
 reading=rdtc:sacch:rssic?   // Report RSSI of current channel.
 if reading!=-1
  print "Current RSSI=",%1d,reading
 endif
 reading=rdtc:sacch:ber?     // Report BER of current channel.
 if reading!=-1
  print "BER=",%1d,reading
 endif
until key?                      // Do until key pressed on RS-232 terminal.
end                             // End macro rdtc_maho.
```

9-8 CELL SITE SIMULATION (CSS)

Using the **SETup**, **STARt** and **STOP** commands of the RECC, RDTC and RVC nodes will stop the Cell Site Simulation. The RECC, RDTC and RVC nodes are automatically set up by the Cell Site Simulation process at the appropriated time:

- Transmitting on the FOCC activates the RECC node.
- Assigning an Analog Voice Channel activates the RVC node.
- Assigning a Digital Voice Channel activates the RDTC node.

Cell Site Simulation starts with the FM/AM-1600CSA transmitting on the FOCC.

9-8-1 GENERAL OVERHEAD MESSAGE

MACRO NAME: css_focc

SYNTAX: css_focc *n,x*
(*n*=FOCC channel number, 1 to 1023. *x*=RF level in dBm, -127.0 to -30.0)

PURPOSE: Transmit an overhead message on the FOCC.

EXAMPLE: css_focc 332,-55.0

MACRO:

```
*dmc "css_focc",begin          // Define macro named css_focc.
css:setup                      // Setup for Cell Site Simulation.
css:chan $1                    // Set RF Channel to 1st value entered with macro.
css:rflvl $2                   // Set RF Output Level to 2nd value entered.
css:start                      // Start Cell Site Simulation tranmission.
css:focc:dcc 0                 // Set Digital Color Code to 0.
css:focc:pci 1                 // Set Protocol Capability Indicator to 1.
css:focc:rcf 1                 // Activate Read Control Filler bit.
css:focc:sid 30                // Set System Identification Number to 30.
css:focc:s 1                   // Activate Serial Number bit.
css:focc:e 1                   // Activate Extended Address bit.
css:focc:regh 0                // Deactivate Home Registration bit.
css:focc:regr 0                // Deactivate Roaming Registration bit.
css:focc:n 20                  /* Set Number of Paging Channels to be scanned
                                  by Mobile Station to 20. */
css:focc:cmax 20               /* Set Maximum Number of Access Channels to be
                                  scanned by Mobile Station to 20. */.
css:focc:auth 1                // Activate Authentication bit.
css:focc:ep 0                  // Deactivate Extended Protocol bit.
css:focc:cmac 2                /* Set Control Mobile Attenuation Code (Mobile
                                  Station Power Level [-2 dBW nominal]). */
css:focc:sdcc1 0               // Set Supplementary Digital Color Code 1 to 0.
css:focc:sdcc2 0               // Set Supplementary Digital Color Code 2 to 0.
css:focc:wfom 1                // Activate Wait For Overhead Message bit.
css:focc:b_i 1                 // Activate Busy-Idle bit (RVC idle).
css:glact:stop                 // Deactivate any global action messages.
end                            // End macro css_focc.
```

9-8-2 GLOBAL ACTION OVERHEAD MESSAGE

MACRO NAME: css_glact

SYNTAX: css_glact *n,x*
(*n*=FOCC channel number, 1 to 1023. *x*=RF level in dBm, -127.0 to -30.0)

PURPOSE: Transmit an overhead message on the FOCC (9-8-1).
Add two global action messages to the overhead message train.

EXAMPLE: css_glact 332,-55.0

MACRO:

```
*dmc "css_glact",begin          // Define macro named css_glact.
css_focc $1,$2                  // Run css_focc macro to send overhead message.
css:glact:action:locaid 1       // Enable Location Area ID global action message.
css:glact:pureg 0               // Disable Power Up Registration bit.
css:glact:pdreg 0               // Disable Power Down Registration bit.
css:glact:lreg 1                // Enable Local Area Registration bit.
css:glact:locaid 240            // Set Location Area ID.
css:glact:action:access 1       /* Enable Access Attempt Parameters global action
                                    message. */
css:glact:maxb:pgr 8            // Set Maximum Busy occurrences for Page responses.
css:glact:maxb:oth 8            /* Set Maximum Busy occurrences for other
                                    responses. */
css:glact:naxs:pgr 9            // Set Maximum Seizures for Page responses.
css:glact:maxs:oth 10           // Set Maximum Seizures for other responses.
css:glact:repeat:on             /* Set to send global action messages continuously
                                    as part of overhead message train. */
css:glact:send                  // Start sending global action messages.
end                             // End macro css_glact.
```

9-8-3 MOBILE STATION CONTROL

MACRO NAME: css_mscm

SYNTAX: css_mscm

PURPOSE: Replace one instance of the system parameter overhead message with a Message Waiting command. (This macro assumes the FM/AM-1600CSA is already transmitting an overhead message on the FOCC.)

MACRO:

```
*dmc "css_mscm",begin           // Define macro named css_mscm.
css:mscm:order:msg_wtg          // Select Message Waiting message.
css:mscm:min "111/111-0111"     // Select MIN of designated Mobile Staion (phone).
css:mscm:local 5                // Set Message Type to indicate 5 messages waiting.
css:mscm:repeat:off             // Set to send message only once.
css:mscm:send                   // Send Message Waiting message.
end                             // End macro css_mscm.
```

9-8-4 MOBILE STATION INITIATED CALL

When assigning a Digital Voice Channel, send a Physical Layer Control (PLC) message to successfully complete the Mobile Station initiated call.

MACRO NAME: minit

SYNTAX: minit *n,x*
(*n*=FOCC channel number, 1 to 1023. *x*=RF level in dBm, -127.0 to -30.0)

PURPOSE: Set FM/AM-1600CSA to transmit an overhead message on the FOCC (9-8-1). Prepare the FM/AM-1600CSA to respond to an access from the Mobile Station. If the access is an Origination, assign the Mobile Station to a digital channel.

EXAMPLE: minit 332,-55.0

MACRO:

```
*dmc "minit",begin              // Define macro named minit.
var n
css_focc $1,$2                  // Run css_focc macro to send overhead message.
css:call:type 1                 // Set for digital channel assignment.
css:call:chan 5                 // Set for RF Channel 5 assignment.
css:call:proc:mobinit           // Start mobile initiated call processing.
do
 delay 500                      // Look for successful access by Mobile Station.
until key? or css:recc:status?
$=recc:torder?
if $="ORIGIN"                   // See if Access was an Origination.
 $=recc:min?                    // Store MIN of Mobile Station (phone) as $.
 css:call:proc:assign           // Assign to digital channel 5.
 delay 500                      // Allow time for phone and CSA to respond.
 n=3                            // Set initial n (count) value to 3.
 do
  css:fdtc:facch:plc            // Send Physical Layer Control message ≤3 times.
  delay 500
 until !(--n) or (rdtc:facch:msgtype?="PLC ACK")
 print "Completed mobile initiated call to ",$
else
 print "Unsuccessful"
endif
end                             // End macro minit.
```

9-8-5 BASE STATION INITIATED CALL

When assigning a Digital Voice Channel, send a PLC message and an Alert message to complete the Base Station initiated call. When assigning an Analog Voice Channel, send an Alert message to complete the Base Station initiated call.

After an Analog or Digital Voice Channel has been assigned, any data field on the respective Reverse Channel can be decoded and any message on the respective Forward Channel can be sent.

MACRO NAME: page

SYNTAX: page *n,x*
(*n*=FOCC channel number, 1 to 1023. *x*=RF level in dBm, -127.0 to -30.0)

PURPOSE: Set FM/AM-1600CSA to transmit a system parameter overhead message followed by a Page order.
Prepare the FM/AM-1600CSA for an access from the Mobile Station.
When a Page response is received from the Mobile Station, assign an analog channel to the Mobile Station and send an Alert message.

EXAMPLE: page 332,-55.0

MACRO:

```
*dmc "page",begin              // Define macro named page.
var st,sat
css_focc $1,$2                 // Setup overhead message with entered values.
css:call:type 0                // Set type for analog.
css:call:chan 600              // Set for RF Channel 600 (TX 888 MHz,RX 843 MHz)
css:call:sat 5970              // Setup for SAT on analog channel.
css:call:dev 2.0               // Setup for SAT deviation of 2.0 kHz.
css:call:vmac 3                /* Set Voice Mobile Attenuation Code (Mobile
                                    Station Power Level [-6 dBW nominal]). */.
css:call:min "111/111-0111"    // Select MIN of Mobile Station being called.
css:call:proc:page             // Start processing for Page.
do
 delay 500                     // Wait for access.
until key? or css:recc:status?
$=recc:torder?                 // Return last decoded order on RECC as variable $.
if $="PAGE RESP"               // Look for a Page response.
 css:call:proc:assign          // Assign to FVC 600 and set SAT to 5970 Hz.
 do
  sat=meas:sat?                // Return the SAT frequency.
 until key? or (sat>5960 and sat<6040)
 n=3                           // Set initial n (count) value to 3.
 do
  css:fvc:order:alert          // Send an Alert message on the FVC.
  delay 300
  st=meas:st?                  // Measure the signalling tone frequency.
 until key? or !(--n) or (st>9990 and st<10010)
 print "Call successfully completed!"
else
 print "Unsuccessful!"
endif
end                            // End macro page.
```

9-8-6 HANDOFF

Analog to Digital Handoff Example:

MACRO NAME: ad_handoff

SYNTAX: ad_handoff *n*
(*n*=handoff channel number, 1 to 1023.)

PURPOSE: Send a handoff message to the Mobile Station, handing off the call to a digital channel. (This macro assumes the FM/AM-1600CSA is on an analog channel with the Mobile Station.)

EXAMPLE: ad_handoff 5

MACRO:

```
*dmc "ad_handoff",begin        // Define macro named ad_handoff.
css:fvc:hand:chan $1           // Set handoff channel to value entered with macro.
css:fvc:ta 0                   // Set Time Alignment offset.
css:fvc:sbi 2                  /* Set Shortened Burst Indicator for transmit
                                  shortened burst after cell-to-cell handoff. */
css:fvc:dvcc 225               // Set Digital Verification Color Code.
css:fvc:dmac 2                 /* Set Digital Mobile Attenuation Code (Mobile
                                  Station Power Level [-2 dBW nominal]). */
css:fvc:mem 0                  // Set Memory Encryption Mode off.
css:fvc:pm 0                   // Set Privacy Mode off.
css:call:proc:fvc:slot1        // Process call handoff to digital channel.
delay 500
do
 css:fdtc:facch:plc            // Send Physical Layer Control message on FDTC.
 delay 500
until !(--n) or (rdtc:facch:msgtype?="PLC ACK")
end                            // End macro ad_handoff.
```

Digital to Analog Handoff Example:

MACRO NAME: da_handoff

SYNTAX: da_handoff *n*
(*n*=handoff channel number, 1 to 1023.)

PURPOSE: Send a handoff message to the Mobile Station, handing off the call to an analog channel. (This macro assumes the FM/AM-1600CSA is on a digital channel with the Mobile Station.)

EXAMPLE: da_handoff 5

MACRO:

```
*dmc "da_handoff",begin        // Define macro named da_handoff.
css:fdtc:ti 0                  // ti=0 means analog handoff.
css:fdtc:dvcc 2                // Set 6030 Hz SAT for SCC on analog handoff.
css:fdtc:dmac 2                // Set VMAC for power level 2 on analog handoff.
css:fdtc:hand:chan $1          // Set handoff channel to value entered with macro.

css:fvc:pscc 2                 // Set present SAT Color Code to match DVCC setup.
css:fvc:vmac 2                 // Set VMAC to match DMAC setup.
css:fvc:mem 0                  // Set Memory Encryption Mode off.
css:fvc:pm 0                   // Set Privacy Mode off.

css:call:sat 6030              // Set SAT frequency to correspond with SCC.
css:call:dev 2.0               // Set SAT deviation to 2.0 kHz.

a=css:call:proc:fdtc:handoff? // Perform handoff.
if a                           // Indicate handoff status out RS-232 Connector.
 print "Handoff is successful!"
else
 print "Unsuccessful!"
endif
end                            // End macro da_handoff.
```

9-9 BIT ERROR RATE (BER)

MACRO NAME: ber_rdtc

SYNTAX: ber_rdtc

PURPOSE: Perform BER test (measure BER of a Base Station receiver) on RDTC (RF Channel 1). (This macro assumes the UUT will loop back the data from the receiver to the transmitter.)

MACRO:

```
*dmc "ber_rdtc",begin          // Define macro named ber_rdtc.
ber:rdtc:setup                 // Setup for Digital Traffic BER.
ber:rdtc:slot 1                // Set to Timeslot 1.
ber:rdtc:chan 1                // Set to RF Channel 1 (TX 825.030 MHz).
ber:rdtc:data:pseudo           // Set to send pseudo-random data on RDTC.
do                             // Initiate do loop.
 ber:rdtc:go                   // Send pseudo-random RDTC data and start BER test.
 delay 100                     // Allow time for Base Station to sync up with data
until key? or !ber:rdtc:status?
delay 2000                     /* or until key is pressed on RS-232 Terminal. */
ber:rdtc:clear                 // Clear current results.
while !key?                    // Display current results until key is pressed on
 tpause                        // RS-232 Terminal.
 a=ber:rdtc:ber?
 if a!=-1
  print "Bit Error Rate = ",%4.3d,a
wend                           //
ber:rdtc:stop                  // Stop RDTC data transmission and BER test.
end                            // End macro ber_rdtc.
```

9-10 MODULATION ACCURACY (MODACC)

MACRO NAME: moderr

SYNTAX: moderr

PURPOSE: Perform Modulation Accuracy test (measure accuracy of π/4 DQPSK signal) on FDTC (RF channel 1).

MACRO:

```
*dmc "moderr",begin            // Define macro named moderr.
mod:fdtc:setup                 // Setup to monitor FDTC for modulation accuracy.
mod:fdtc:chan 1                // Set to RF Channel 1 (monitor 870.030 MHz).
a=mod:fdtc:run?                // Run Modulation Accuracy test until complete or
while !key? and !mod:fdtc:comp?
wend                           // key is pressed on RS-232 Terminal.
print mod:fdtc:evm?            // Indicate RMS Error Vector Magnitude percentage.
end                            // End macro moderr.
```

9-11 GPIB

LANGUAGE: HP™ Basic

PURPOSE: Perform Cell Site Simulation testing across GPIB.

FILE:

```
10    Dev=705
20    OUTPUT Dev;":GPIB:MASK 1"
30    ON INTR 7 GOSUB 620
40    ENABLE INTER 7;2
50    CLEAR SCREEN
60    PRINT "START"
70    DIM COMM$[60]
80    OUTPUT Dev;"CSS:SETUP"
90    OUTPUT Dev;"CSS:CHAN 333"
100   OUTPUT Dev;"CSS:RFLVL -60"
110   OUTPUT Dev;"CSS:FOCC:PCI 1"
120   OUTPUT Dev;"CSS:START"
130   OUTPUT Dev;"CSS:CALL:TYPE 1"
140   OUTPUT Dev;"CSS:CALL:CHAN 5"
150   OUTPUT Dev;"CSS:CALL:SLOT 2"
160   OUTPUT Dev;"CSS:FDTC:ENABLE:SIGNAL 1"
170   OUTPUT Dev;"CSS:FDTC:SIGNAL:PITCH 0;CADENCE 1"
180   OUTPUT Dev;"CSS:FDTC:ENABLE:CALLING:NUM 1"
190   OUTPUT Dev;"CSS:FDTC:CALLING:NUM '316/522-4981'"
200   OUTPUT Dev;"CSS:FDTC:CALLING:TYPE 0"
210   OUTPUT Dev;"CSS:FDTC:CALLING:PLANID 0"
220   OUTPUT Dev;"CSS:FDTC:CALLING:PI 0"
230   OUTPUT Dev;"CSS:FDTC:CALLING:SI 0"
240   OUTPUT Dev;"CSS:FDTC:ENABLE:DMAC 0;TA 1;DTX 0;DIC 0"
250   OUTPUT Dev;"CSS:FDTC:TA 2"
260   Srq_flag=0
270   REPEAT
280     OUTPUT Dev;"CSS:CALL:PROC:REG"
290     WAIT 1
300   UNTIL Srq_flag=1
310   PRINT "REGISTRATION COMPLETED"
320   OUTPUT Dev;"CSS:CALL:PROC:PAGE"
330   Srq_flag=0
340   REPEAT
350     WAIT .5
360   UNTIL Srq_flag=1
370   PRINT "PAGE RESPONSE RECEIVED"
380   OUTPUT Dev;"CSS:CALL:PROC:ASSIGN"
390   WAIT 2
400   PRINT "CALL ASSIGN TO DIGITAL TRAFFIC CHANNEL
410   REPEAT
420     OUTPUT Dev;"CSS:FDTC:FACCH:PLC"
430     WAIT .5
440     OUTPUT Dev;"RDTC:FACCH:MSG?"
450     ENTER Dev;Comm$
460   UNTIL Comm$="PLC ACK"
```

(The GPIB example program continues on the next page.)

```
470  PRINT "PLC ACK RECEIVED"
480  REPEAT
490    OUTPUT Dev;"CSS:FDTC:FACCH:ALERT"
500    WAIT .5
510    OUTPUT Dev;"RDTC:FACCH:MSG?"
520    ENTER Dev;Comm$
530  UNTIL Comm$="MOBILE ACK"
540  PRINT "PHONE ALERTED"
550  REPEAT
560    WAIT .5
570    OUTPUT Dev;"RDTC:FACCH:MSG?"
580    ENTER Dev;Comm$
590  UNTIL Comm$="CONNECT"
600  PRINT "CALL COMPLETED"
610  STOP
620  Srq_flag=1
630  S=SPOLL(Dev)
640  ENABLE INTR 7;2
650  RETURN
660  END
```

APPENDICES

APPENDIX A - PREDEFINED MACROS AND CONSTANTS

A-1 PREDEFINED MACROS

The following macros are predefined in the FM/AM-1600S and FM/AM-1600CSA:

```
*DMC "Chirp_1", BEGIN
    SOUND 1600,50
    DELAY 100
    SOUND 1600,50
END
```

```
*DMC "Chirp_2", BEGIN
   SOUND 1300,30
   SOUND 1500,30
   SOUND 1700,30
   SOUND 1500,30
   SOUND 1300,30
END
```

A-2 PREDEFINED CONSTANTS

The following colors are predefined constants in the FM/AM-1600S and FM/AM-1600CSA:

WHITE	YELLOW	MAGENTA
RED	CYAN	GREEN
BLUE	DARK_GRAY	LIGHT_GRAY
BROWN	DARK_MAGENTA	DARK_RED
DARK_CYAN	DARK_GREEN	DARK_BLUE
BLACK		

The following Front Panel Keys are predefined constants and are equal to the corresponding keycodes (see Appendix B):

DEL	CE	RCL
STOR	SETUP	SGL_STEP
AUTO	GO	STOP
F_RIGHT	F_LEFT	RIGHT
LEFT	F_DOWN	F_UP
DOWN	UP	BACK_ARROW
ESC	ENTER	F1
F2	F3	F4
F5	F6	

APPENDIX B - FRONT PANEL KEYS AND KEYCODES

Table B-1 lists the FM/AM-1600S Front Panel Keys and the corresponding keycodes. Press the SHIFT Key to toggle the Front Panel Keyboard to the alphabetic characters printed in blue on the Front Panel. Press the SHIFT Key again to toggle the Front Panel Keyboard to the numeric values.

Key	Keycode	Key	Keycode	Key	Keycode
F1	1152	+/-	1025	K	75
F2	2176	4	2056	L	76
F3	4224	5	2052	M	77
F4	8320	6	2050	N	78
F5	16512	M/μ	2049	O	79
F6	32896	1	4104	P	80
RF GEN	1088	2	4100	Q	81
RCVR	2112	3	4098	R	82
DPLX	4160	K/m	4097	S	83
AF GEN	8256	.	8208	T	84
SCOPE/ANLZ	16448	*	8200	U	85
MTRS	32832	0	8196	V	86
AUTO	1056	#	8194	W	87
SGL STEP	2080	ENTER	8193	X	88
GO	4128	A	65	Y	89
STOP	8224	B	66	Z	90
SETUP	16400	C	67	FIELD SELECT ←	257
STORE	32784	D	68	FIELD SELECT ↑	513
RCL	32776	E	69	FIELD SELECT ↓	258
CE	16388	F	70	FIELD SELECT →	514
DEL	32772	G	71	DATA SCROLL ↑	264
7	1032	H	72	DATA SCROLL ↓	260
8	1028	I	73	DATA SCROLL ←	520
9	1026	J	74	DATA SCROLL →	516

Table B-1 Front Panel Keys and Keycodes

APPENDIX C - ABBREVIATIONS

A	
A	Ampere
ac	Alternating Current
AF	Audio Frequency
AGC	Automatic Gain Control
AM	Amplitude Modulation
AMPS	Advanced Mobile Phone Service
ANLZ	Analyzer
ANSI	American National Standards Institute
ANT	Antenna
ASCII	American National Standard Code for Information Interchange
B	
BFO	Beat Frequency Oscillator
BER	Bit Error Rate
bps	Bits per second
C	
CCH	Control Channel
ccw	Counterclockwise
CRT	Cathode Ray Tube
cw	Clockwise
C Wt	C-Weight
D	
dB	Decibels
dBm	Decibels relative to 1 milliwatt
dc	Direct Current
DCS	Digital Coded Squelch
Demod Audio	Demodulated Audio
DMM	Digital Multimeter
DSAT	Digital Supervisory Audio Tone
DST	Digital Signalling Tone
DTMF	Dual Tone Multi-Frequency
E	
ESD	Electrostatic discharge
ESN	Electronic Serial Number
ETACS	Enhanced Total Access Communications System
Ext	External
EXT MOD	External Modulation
F	
FACCH	Fast Associated Control Channel
FDTC	Forward Digital Traffic Channel
FM	Frequency Modulation
FOCC	Forward Control Channel
Freq	Frequency
Func Gen	Function Generator
FVC	Forward Voice Channel

	G
Gen	Generator
GHz	Gigahertz
GND	Ground
GPIB	General Purpose Interface Bus
	H
Hz	Hertz
Hex	Hexadecimal
	I
Id	Identification
IEEE	Institute of Electrical and Electronic Engineers
IMTS	Improved Mobile Telephone Service
I/O	Input/Output
	K
kbps	kilobits per second
kHz	kilohertz
kΩ	kilo ohm
	L
LSB	Lower Sideband
Lvl	Level
	M
MHz	Megahertz
MIN	Mobile Identification Number
MOD	Modulation
MODL	Modulation Level
MΩ	Mega ohm
ms	Milliseconds
msg	message
MTS	Mobile Telephone Service
mV	Millivolts
mW	Milliwatts
	N
N/A	Not applicable
NAMPS	Narrow Band Advanced Mobile Phone Service
NVRAM	Non-Volatile Random Access Memory
	O
Opn	Operation
	P
PM	Phase Modulation
PWR	Power

R	
RAM	Random Access Memory
RBW	Resolution Bandwidth
RCI	Remote Command Interpreter
RCL	Recall
RCV	Receive
Rcvr	Receiver
RDTC	Reverse Digital Traffic Channel
RECC	Reverse Control Channel
Res	Resolution
RF	Radio Frequency
RF Pwr	RF Power
Rgtr	Register
RMS	Root Mean Square
RVC	Reverse Voice Channel
S	
SACCH	Slow Associated Control Channel
SAT	Supervisory Audio Tone
SCC	Supervisory Audio Tone Color Code
sec	Second
SIN	Sine
SQU	Square
SSB	Single Side Band
SSD	Shared Secret Data
Std	Standard
T	
TDMA	Time Division Multiple Access
TERM	Terminal
TRI	Triangle
Tx	Transmit
U	
USB	Upper Sideband
UUT	Unit Under Test
V	
V	Volt
VCHAN	Voice Channel
Vrms	Voltage Root Mean Square
VSELP	Vector Sum Excited Linear Prediction
W	
W	Watt
X	
Xmtr	Transmitter

INDEX

D

E

F

F (continued)

G

H

V

W

X