

Reference Manual

Tektronix

PRELIMINARY

AM700

Programmer's Quick Reference

070-9034-00

First Printing: October 1994

Copyright © 1994 Tektronix, Inc. All rights reserved. Printed in the United States of America. Contents of this publication may not be reproduced in any form without the written permission of Tektronix, Inc. Specification and price change privileges are reserved.

This instrument, in whole or in part, may be protected by one or more U.S. patents or foreign patents or patent applications.

TEKTRONIX and TEK are registered trademarks.

For further information, contact Tektronix, Inc., Corporate Offices, P. O. Box 1000, Wilsonville, OR 97070-1000. Phone: (503) 627-7111; TLX: 192825; TWX: (910) 467-8708; Cable: TEKWSGT.

TRACe

TRACe:CATalog? returns a list of trace names for the application in use.

TRACe[:DATA]? <trace_name> returns the data values for the named trace.

TRACe:DATA:VALue? <trace_name>,<data_point>
{, <data_point>}

TRACe:POINTs? <trace_name>

TRIGger

START

STOP

UNIT

UNIT:FREQuency Hz|kHz|MHz

UNIT:IMPedance Ohm|kOhm

UNIT:RATio PCT|DB

UNIT:TIME HOUR|MINute|SECond

UNIT:VOLTage V|mV|dBu|dBv|dBFS

AM700 SCPI Commands

SYSTem:COMMunicate:SERial[1|2]:CONTRol:RTS
ON|RFR|IBFull

SYSTem:COMMunicate:SERial[1|2]:FEED <string>

SYSTem:COMMunicate:SERial[1|2]:BAUD <numeric_value>

SYSTem:COMMunicate:SERial[1|2]:BITS <numeric_value>

SYSTem:COMMunicate:SERial[1|2]:SBITs <numeric_value>

SYSTem:COMMunicate:SERial[1|2]:PACe XON|NONE

SYSTem:COMMunicate:SERial[1|2]:PARity[:TYPE]
EVEN|ODD|ZERO|ONE|NONE

SYSTem:COMMunicate:RELay[:STATe] <Boolean>

SYSTem:DATE <year>,<month>,<day>

SYSTem:ERRor?

SYSTem:STATe:SAVE:IMMediate

SYSTem:TIME <hour>,<minute>,<second>

SYSTem:VERSion?

Contents

Remote Operation

Control Protocol	1-1
Control Ports	1-2
GPIB Interface Information	1-2
IEEE 488.1 Interface Functions	1-2
Accessories for Remote Control	1-3
RS-232C Interface Information	1-3
Connector Information	1-4
Rear Panel Remote Control Connectors	1-5
I/O Connectors	1-5
GPIB	1-5
COM1, COM2	1-5
REMOTE Connector	1-6
SCPI Programming	1-8
IEEE 488.2 Commands	1-8
SCPI Command Subsystems Implemented in the AM700	1-10
Keyword Suffixes	1-13
Mnemonic Keyword	1-13
Compound Headers	1-14
AM700 Setup	1-16
Generator Controls	1-16
Signal Input Configuration	1-17
Application Control	1-18
File Management	1-18
Other Instrument Operations	1-19
Generator, Output, and Source Controls	1-20
GCONTRol Subsystem	1-20
OUTPut Commands	1-21
SOURCe Commands	1-21
AM700 SCPI Model	1-23
AM700 Signal Selection and Routing	1-23
AM700 Analog Signal Path	1-24
Route Subsystem	1-26
INPut Subsystem	1-28
IMPedance	1-30
SENSe Subsystem	1-30
CSTReam Subsystem	1-31
CALCulate Subsystem	1-32

AMEasure Subsystem	1-34
FFT and the AMEasure Subsystem	1-35
DISPlay Subsystem	1-35
INSTrument Subsystem	1-36
SYSTem Subsystem	1-37
STATus Subsystem	1-37
TRACe Subsystem	1-37
MMEMory Subsystem	1-38
AM700 File Structure	1-39
Functions	1-41
Running a Function	1-43
Timed Functions	1-44

AM700 SCPI Commands

AM700 SCPI Commands	2-1
User SCPI Commands	2-1

```

STATus:QUEStionable:INSTrument:MAP <numeric_value>,
<numeric_value>

STATus:QUEStionable:SOURce[:EVENT]?

STATus:QUEStionable:SOURce:CONDition?

STATus:QUEStionable:SOURce:ENABle <numeric_value>

STATus:QUEStionable:SOURce:PTRansition
<numeric_value>

STATus:QUEStionable:SOURce:NTRansition
<numeric_value>

STATus:QUEStionable:SOURce:MAP <numeric_value>,
<numeric_value>

STATus:QUEStionable:SOURce:SUMMary[:EVENT]?

STATus:QUEStionable:SOURce:SUMMary:CONDition?

STATus:QUEStionable:SOURce:SUMMary:ENABle
<numeric_value>

STATus:QUEStionable:SOURce:SUMMary:PTRansition
<numeric_value>

STATus:QUEStionable:SOURce:SUMMary:NTRansition
<numeric_value>

STATus:QUEStionable:SOURce:SUMMary:MAP
<numeric_value>,<numeric_value>

```

SYSTEM

```

SYSTem:COMMunicate:GPIB:ADDRESS <0-30>

SYSTem:COMMunicate:GPIB[:SELF]:MODE
TLISten|TONLy|OBUS

SYSTem:COMMunicate:GPIB:RDEvice:FEED <string>

```

AM700 SCPI Commands

STATus:QUEStionable:NTRansition <numeric_value>
STATus:QUEStionable:PTRansition <numeric_value>
STATus:QUEStionable:INPut[:EVENT]?
STATus:QUEStionable:INPut:CONDition?
STATus:QUEStionable:INPut:ENABle <numeric_value>
STATus:QUEStionable:INPut:PTRansition
<numeric_value>
STATus:QUEStionable:INPut:NTRansition
<numeric_value>
STATus:QUEStionable:INPut:MAP <numeric_value>,
<numeric_value>
STATus:QUEStionable:INPut:SUMMary[:EVENT]?
STATus:QUEStionable:INPut:SUMMary:CONDition?
STATus:QUEStionable:INPut:SUMMary:ENABle
<numeric_value>
STATus:QUEStionable:INPut:SUMMary:PTRansition
<numeric_value>
STATus:QUEStionable:INPut:SUMMary:NTRansition
<numeric_value>
STATus:QUEStionable:INPut:SUMMary:MAP
<numeric_value>,<numeric_value>
STATus:QUEStionable:INSTRument[:EVENT]?
STATus:QUEStionable:INSTRument:CONDition?
STATus:QUEStionable:INSTRument:ENABle
<numeric_value>
STATus:QUEStionable:INSTRument:PTRansition
<numeric_value>
STATus:QUEStionable:INSTRument:NTRansition
<numeric_value>

Illustrations

Figure 1-1: AM700 Rear Panel	1-5
Figure 1-2: GPIB Connector	1-6
Figure 1-3: COM1 and COM2 Serial Ports	1-6
Figure 1-4: Remote Contact-Closure Connector	1-7
Figure 1-5: Generator Control Model	1-20
Figure 1-6: AM700 SCPI Systems for Measurements	1-24
Figure 1-7: Analog Hardware SCPI Subsystem Commands ..	1-25
Figure 1-8: Signal Routing and Selection	1-26
Figure 1-9: FFT CALCulate Subsystem Block	1-33
Figure 1-10: Switching Between FFT and Multitone	1-34

Tables

Table 1-1: IEEE 488.1 Interface Functions Implemented in the AM700	1-2
Table 1-2: Serial Remote Control Protocol (serial printer support only)	1-4
Table 1-3: Remote Connector Pins	1-7
Table 1-4: IEEE 488.2 Status Reporting Commands	1-8
Table 1-5: Internal Operation Commands	1-9
Table 1-6: SCPI Subsystems Implemented in the AM700	1-10
Table 1-7: New Command Subsystems Implemented in the AM700	1-11
Table 1-8: SCPI Command Syntax Symbols	1-11
Table 1-9: Signals Available From the Generators	1-23

STATus:OPERation:ENABLE <numeric_value>
STATus:OPERation:NTRansition <NRF>
STATus:OPERation:PTRansition <NRF>
STATus:OPERation:INSTrument[:EVENT]?
STATus:OPERation:INSTrument:CONDition?
STATus:OPERation:INSTrument:ENABLE <numeric_value>
STATus:OPERation:INSTrument:PTRansition
<numeric_value>
STATus:OPERation:INSTrument:NTRansition
<numeric_value>
STATus:OPERation:INSTrument:MAP <numeric_value>,
<numeric_value>
STATus:OPERation:TRIGger[:EVENT]?
STATus:OPERation:TRIGger:CONDition?
STATus:OPERation:TRIGger:ENABLE <numeric_value>
STATus:OPERation:TRIGger:PTRansition <numeric_value>
STATus:OPERation:TRIGger:NTRansition <numeric_value>
STATus:OPERation:SYSTem[:EVENT]?
STATus:OPERation:SYSTem:CONDition?
STATus:OPERation:SYSTem:ENABLE <numeric_value>
STATus:OPERation:SYSTem:PTRansition <numeric_value>
STATus:OPERation:SYSTem:NTRansition <numeric_value>
STATus:OPERation:SYSTem:MAP <numeric_value>,
<numeric_value>
STATus:QUEStionable[:EVENT]?
STATus:QUEStionable:CONDition?
STATus:QUEStionable:ENable <numeric_value>

AM700 SCPI Commands

SOURce[1-8]:SWEep:COUNT <numeric_value>
SOURce[1-8]:SWEep:DIRection UP|DOWN
SOURce[1-8]:SWEep:DWELL <numeric_value>
SOURce[1-8]:SWEep:POINTs <numeric_value>
SOURce[1-8]:SWEep:SPACing LINear|LOGarithmic
SOURce[1-8]:SWEep:TIME <numeric_value>
SOURce[1-8]:SWEep:TIME:AUTO <Boolean>
SOURce[1-8]:USER:DATA:SCALing[:STATe] <Boolean>
SOURce[1-8]:USER:DATA:FILE:NAME <file_name>
SOURce[1-8]:USER:DATA:FILE:LNAME?
SOURce[1-8]:VOLTage:MODE CW|FIXed|SWEep|LIST
SOURce[1-8]:VOLTage:START <numeric_value>
SOURce[1-8]:VOLTage:STOP <numeric_value>
SOURce[1-8]:VOLTage[:LEVel] <numeric_value>
SOURce[1-8]:VOLTage[:LEVel][:IMMediate]
<numeric_value>
SOURce[1-8]:VOLTage[:LEVel][:IMMediate][:AMPLitude]
<numeric_value>

STATus

STATus:PRESet
STATus:QUEue[:NEXT]?
STATus:QUEue:ENABle <parameter>
STATus:OPERation[:EVENT]?
STATus:OPERation:CONDition?

Remote Operation

```

SOURCE:TBURst:PERiod <2 to 100000>
SOURCE:TBURst:WIDTh <1 to 100000>
SOURCE[1-8]:FUNction:SHApe SIMD
SOURCE:SIMD:VFRequency <numeric_value>
SOURCE:SIMD:IMFRequency <numeric_value>
SOURCE:SIMD:RATio <numeric_value>
SOURCE[1-8]:FUNction:SHApe CIMD
SOURCE:CIMD:TYPE CCIF
SOURCE:CIMD:CFRequency <numeric_value>
SOURCE:CIMD:RATio <numeric_value>
SOURCE:CIMD:SPACing <numeric_value>
SOURCE[1-8]:FUNction:SHApe SNOise
SOURCE:SNOise:BSASs:TYPE WHITE|PINK
SOURCE[1-8]:FUNction:SHApe PCHirp
SOURCE[1-8]:FUNction:SHApe POLarity
SOURCE[1-8]:FUNction:SHApe TPOLarity
SOURCE[1-8]:FUNction:SHApe USER
SOURCE[1-8]:FUNction:SHApe MTONE
SOURCE[1 - 8]:LIST:VOLTage
<volt_value1,{volt_value2,...volt_value128}>
SOURCE[1-8]:LIST:FREQuency
<freq1{,freq2,...freq128}>
SOURCE[1-8]:LIST:DWELl <time1{,time2,...time128}>
SOURCE[1-8]:MTONE:MODE LIST|FILE
SOURCE[1-8]:MTONE:DATA:FILE:NAME <file_name>
SOURCE[1-8]:MTONE:DATA:FILE:LNAME <file_name>
SOURCE[1-8]:STATe <Boolean>

```


SOURce5 Digital generator DSF 1
 SOURce6 Digital generator DSF 2

SOURce7 Digital Signal Processor A
 SOURce8 Digital Signal Processor B

SOURce[5-8]:DAUDio:AUDio:SRATe HIGH|MEDIum|LOW

SOURce[5,6]:DAUDio:AUDio:WSIZE <numeric_value>

SOURce[5,6]:DAUDio:CSTATUS:MODE PROFes-
sional|CONSUMER|NAUDio|FILE

SOURce[5,6]:DAUDio:CSTATUS:FILE:NAME <filename>

SOURce[5,6]:DAUDio:UBITs:MODE NULL|FILE

SOURce[5,6]:DAUDio:UBITs:FILE:NAME <filename>

SOURce[5,6]:DAUDio:UBITs:FILE:LNAME <filename>

SOURce[5-8]:DITHer:TYPE RECTangular|TRIangular
 Setting dither type for either A or B
 sets both to the same selection.

SOURce[1-8]:FOLLow[:STATe] <Boolean>

SOURce[1-8]:FREQuency <numeric_value>

SOURce[1-8]:FREQuency:MODE CW|FIXed|SWEep|LIST

SOURce[1-8]:FREQuency:START <numeric_value>

SOURce[1-8]:FREQuency:STOP <numeric_value>

SOURce[1-2]:DITHer:TYPE RECTangular|TRIangular

SOURce[1-8]:FUNCTion:SHAPE <signal_name>

SOURce:FUNCTion:SHAPE SINusoid

SOURce[5|6]:FUNCTion:SHAPE JSINE

SOURce[5|6]:JSINE:JFREquency <numeric_value>

SOURce[5|6]:JSINE:JAMPLitude <numeric_value>

SOURce[1-8]:FUNCTion:SHAPE TBURst

Remote Operation

Control Protocol

The AM700 uses SCPI-1994 as its primary control protocol. SCPI (Standard Commands for Programmable Instruments) is an emerging standard promoted by a consortium of test and measurement equipment vendors, including Tektronix. The intent of SCPI is to provide a consistent and standard command language for all test and measurement equipment. SCPI is layered on top of IEEE 488.2, and contains several 488.2 commands and status structures.

Remote control of the AM700 makes it possible to do most of the user interface operations available via the front panel, plus several things that come with a protocol such as SCPI. These other things are:

1. Immediately place an application in a certain state.
2. Query internal data points not directly accessible to a user at the front panel.
3. Place prompting messages on the screen and query for key presses for interactive procedures.
4. Set up timed functions.

The programming language for functions is Tcl. The Tcl commands permit a programmer to build conditional tests and create their own functions. The Tcl interpreter in the AM700 allows imbedding SCPI commands in a function program using the “scpi” Tcl command. The parser directs the SCPI commands appropriately. The Tcl command language is documented in *Part I: The Tcl Language* of the book *Tcl and the Tk Toolkit*, by John K. Ousterhout: Addison-Wesley Publishing Company, 1994.

Control Ports

The instrument may be controlled from the following sources:

1. The user operating the front panel keys, touch panel, and, for certain operations, an external keyboard.
2. The GPIB port; a standard IEEE-488.2 interface.

Though it is not advised, whenever it is not explicitly disabled as part of the protocol it is possible to control the instrument via both the front panel and the GPIB interface port at the same time.

GPIB Interface Information

In normal GPIB remote operation the AM700 is controlled via an external instrument controller. The only time the AM700 will attempt to take control of the GPIB is when a hardcopy is requested from the front panel with the print port configured to 'GPIB'.

IEEE 488.1 Interface Functions

The GPIB interface function set implemented in the AM700 GPIB interface and the capability level is given in Table 1-1.

Table 1-1: IEEE 488.1 Interface Functions Implemented in the AM700

Function	Implemented	Notes
Source Handshake	SH1	Complete capability
Acceptor Handshake	AH1	Complete capability
Talker	T6	Basic Talker, Serial Poll, Unaddr if MLA
Talker (extended)	TE0	No capability

SENSe[1-11]

Suffixes: Sense suffixes are:

SENSe1	Analog A High Res A/D
SENSe2	Analog A High BW A/D
SENSe3	Analog B High Res A/D
SENSe4	Analog B High BW A/D
SENSe5	DSF1 Digital Main
SENSe6	DSF2 Digital Main
SENSe7	A DSP Port
SENSe8	B DSP Port
SENSe9	A Digital Ref
SENSe10	B Digital Ref
SENSe11	Eye Pattern

SENSe[5-8]:DAUDio:AUDio:SRATe?

SENSe11:DAUDio:INTErface:BWIDth MEDium|HIGH

SENSe11:DAUDio:INTErface:JGAIN NORMAl|HIGH

SENSe5:DAUDio:INTErface:CORREction:EQualization?

SENSe[5,7]:DAUDio:INTErface:FREQuency?

SENSe5:DAUDio:INTErface:FREQuency:RATio?

SENSe11:DAUDio:INTErface:JSpectrum:EQualization
<Boolean>

SENSe5:DAUDio:INTErface:PHASe:DIFFerence?

SENSe5:DAUDio:INTErface:VOLTage:AC?

SOURce[1-8]

Suffixes:	SOURce1	Analog generator HR A
	SOURce2	Analog generator HR B
	SOURce3	Analog generator HB A
	SOURce4	Analog generator HB B

ROUTe [1 | 2 | 3]

Route suffixes are:

ROUTe1 Analog A input
 ROUTe2 Analog B input
 ROUTe3 Digital main input

ROUTe1:CLOSE 1|3

ROUTe2:CLOSE 2|4

ROUTe3:CLOSE 6|7|8

Close <numeric_values> are:

CLOSE 1 Analog A input connector
 CLOSE 2 Analog B input connector
 CLOSE 3 Analog generator channel A
 CLOSE 4 Analog generator channel B
 CLOSE 6 Digital front panel XLR
 CLOSE 7 Digital rear panel BNC
 CLOSE 8 Digital rear panel optical
 CLOSE 9 Digital Signal Processor

ROUTe1:CLOSE 1|3

ROUTe2:CLOSE 2|4

ROUTe3:CLOSE 6|7|8

ROUTe1:CLOSE? 1|3

ROUTe2:CLOSE? 2|4

ROUTe3:CLOSE? 6|7|8|9

ROUTe[1|2|3]:CLOSE:STATE?

ROUTe[1|2|3]:CLOSE <numeric_value>

ROUTe[1|2|3]:CLOSE:STATE?

Table 1-1: IEEE 488.1 Interface Functions Implemented in the AM700 (Cont.)

Function	Implemented	Notes
Listener	L4	Basic Listener, Unaddr if MTA
Listener (extended)	LE0	No capability
Service Request	SR1	Complete capability
Remote Local	RL0	No local lock out
Parallel Poll	PP0	No capability
Device Clear	DC1	Complete capability
Device Trigger	DT0	No capability
Controller	C0	No capability.
Electrical Interface	E2	Three-state bus drivers

Accessories for Remote Control

- GPIB Cable: double shielded in various lengths (user supplied)
- Preliminary Programmer Manual
- Preliminary Programmer's Quick Reference

RS-232C Interface Information

The COM1 and COM2 ports may be configured as serial RS-232C ports. These are DB-9 male connectors. At first release, these ports do not support remote control, and are used for hard copy output to serial printers. The Serial Interface parameters given in Table 1-2 are user selectable using menu control. If you are connecting the port to another terminal, you will need a null modem connector to make the appropriate connections.

Connector Information

Table 1-2: Serial Remote Control Protocol (serial printer support only)

Capabilities	Description
Control Ports	
COM1 and COM2	RS-232C
Connector	DB-9, male
Serial Interface	
RS-232C	
Baud Rate	User selectable: 300, 600, 1200, 2400, and 9600, 19,200, and 38,400.
Flow Control	XON/XOFF, CTS/RTS, and None.
Signal Bits	7 or 8
Stop Bits	1 or 2
Parity	Odd, Even, and None

```

PROGrama:CATalog?
PROGrama:RCATalog?
PROGrama:TCATalog?
PROGrama:EXPLicit:DEFine <programe>,<program_data>
PROGrama:EXPLicit:DELeTe <programe>
PROGrama:EXPLicit:LABel? <programe>
PROGrama:EXPLicit:STATe <programe>,RUN|STOP
PROGrama:EXPLicit:TIMed[:SET] <prog-
name>,'cron_string' {,'cron_string'}
PROGrama:EXPLicit:TIMed:CLEar <programe>
PROGrama:EXPLicit:TIMed:ADD <programe>,'cron_string'
{,'cron_string'}
PROGrama:EXPLicit:WAIT <programe>
PROGrama:SELeCted:DEFine <prog_data>
PROGrama:SELeCted:DELeTe:SELeCted
PROGrama:SELeCted:DELeTe:ALL
PROGrama[SELeCted]:LABel?
PROGrama:SELeCted:LNAME?
PROGrama:SELeCted:NAME <programe>
PROGrama:SELeCted:STATe RUN|STOP
PROGrama:SELeCted:TIMed[:SET] [<cron_string>]
{,<cron_string>}
PROGrama:SELeCted:TIMed:CLEar
PROGrama:SELeCted:TIMed:ADD
<cron_string>{,<cron_string>}
PROGrama:SELeCted:WAIT

```

OUTPut [1-4]

Suffixes: OUTPut1 Analog A
 OUTPut2 Analog B
 OUTPut3 Digital Main
 OUTPut4 DSP (no commands for DSP)

OUTPut:COMMon FLOat|GROund
 Setting either 1 or 2 changes both to
 the same setting.

OUTPut3:DAUDio:INTERface:CLOCK:PHASE:ADJust
 <numeric_value>

OUTPut3:DAUDio:INTERface:CLOCK:VFRequency
 <numeric_value>

OUTPut3:DAUDio:INTERface:CLOCK:FRequency:ADJust
 <numeric_value>

OUTPut3:DAUDio:INTERface:CLOCK:MODE Audio|DARS|VARI-
 able

OUTPut3:DAUDio:INTERface:VOLTage:BALanced
 <numeric_value>

OUTPut3:DAUDio:INTERface:VOLTage:UBALanced
 <numeric_value>

OUTPut3:FILTer[:LPASSs][:STATE] ON|OFF (This is the
 long cable simulator.)

OUTPut3:FILTer[:LPASSs]:TYPE?

OUTput:IMPedance 10|150|600

PROGram

The PROGram commands are used to access the Functions feature of the AM700. The pronames are the names of function files located in the ROM:/FUNCTION and NVRAM:/FUNCTION directories of the AM700. The data supplied in <program> must be in arbitrary block data format.

Rear Panel Remote Control Connectors

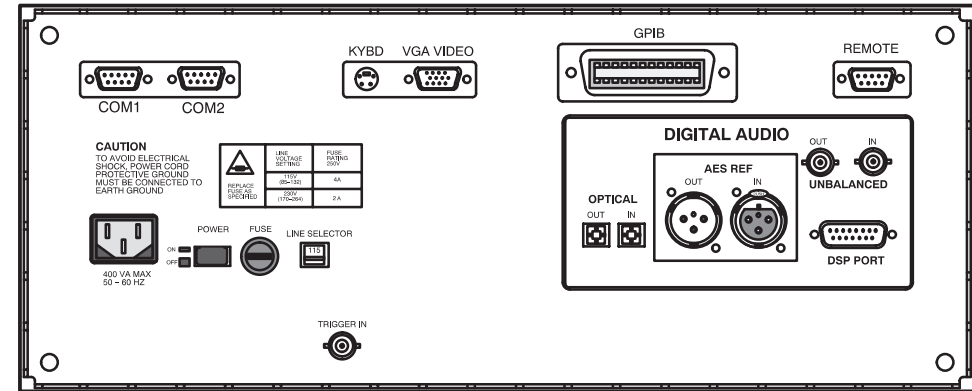


Figure 1-1: AM700 Rear Panel

I/O Connectors

GPIB. The GPIB connector (Figure 1-2) provides a remote control interface to the AM700. It is a standard IEEE 488 parallel GPIB connector. One- and two-meter, double-shielded GPIB cables are available as optional accessories.

COM1, COM2. These two male DB-9 connectors (Figure 1-3) provide the interface for RS-232C serial input/output. At first release these connectors only support RS-232C printer output.

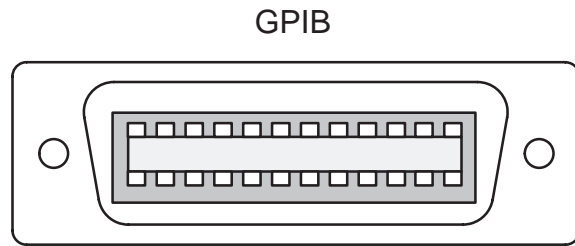


Figure 1-2: GPIB Connector

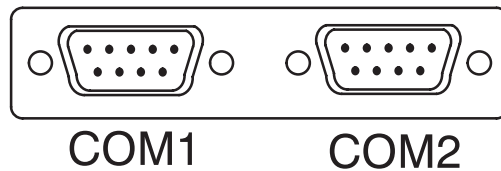


Figure 1-3: COM1 and COM2 Serial Ports

REMOTE Connector. This connector provides a user interface for contact-closure remote control. The AM700 may use this port to control external devices that are actuated via a contact-closure. The TTL-Level input may be used with external contact-closure relay or switch to initiate an action for the AM700. A possible use of this feature is to use a foot switch to start a function key operation or application in a production line. This type of operation leaves the operator's hands free to make connections to the equipment under test and operate the AM700 making the measurements without having to have immediate access to the front panel of the AM700. A function program with the appropriate commands is needed to make use of the remote connector for the operation just described.

```
INSTRument:LCATalog?
    Like CAT, but gives descriptive name

INSTRument:LCATalog:FULL?
    Like CAT:FULL, but gives descriptive
name and number.

INSTRument:LSElect <descriptive_name>

INSTRument:NSElect <app_number>

INSTRument[:SElect] <app_name>

INSTRument:STATE?
    Cannot be set, just returns: AUTO
```

MMEMory

```
MMEMory:CATalog?

MMEMory:CDIRectory '[device:/]directory_name'

MMEMory:CLOSe

MMEMory:COpy '[device:/{path_name/}source_file',
'[device:/{path_name/}destination_file'

MMEMory:DATA <filename>,<definite length block
data>

MMEMory:DElete '[device:/{path_name/}file_name'

MMEMory:FEED 'data handle'

MMEMory:NAME '[device:/{path_name/}file_name'

MMEMory:OPEN
```

```
HCOPY[:IMMEDIATE]
HCOPY:ITEM:ALL[:IMMEDIATE]
HCOPY:SDUMP[:IMMEDIATE]
```

INPut[1-3]

Suffixes: Input suffixes are:

```
INPut1    Analog_A_Unfiltered
INPut3    Analog_B_Unfiltered
```

```
INPut[1|3]:IMPedance <numeric_value>
INPut[1|3]:RANGe <numeric_value>
INPut[1|3]:RANGe:AUTO ON|OFF|ONCE
```

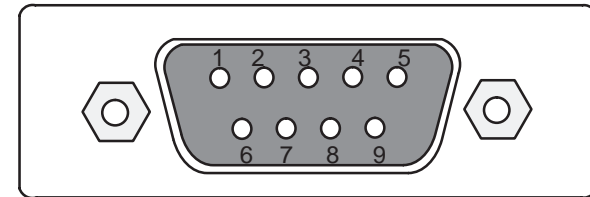
INSTRument

Application numbers and short names for use with NSEL and SEL commands.

```
NSEL 1    FFT
NSEL 2    ANALYZER
NSEL 3    MONITOR
NSEL 4    DIGITAL
NSEL 5    DIAGNOSTIC
NSEL 6    PANELCAL
```

```
INSTRument:CATalog?
    Gives list of strings which contain
<app_name>s
INSTRument:CATalog:FULL?
    List alternates <app_name> string,
<app_number>
```

REMOTE



DB-9 FEMALE REMOTE CONNECTOR

```
Pin 1      TTL-Level Input
Pins 2, 4, 6, 8  Ground
Pin 3 to Pin 9  Normally Open
Pin 5 to Pin 9  Normally Closed
Pin 7      +5 V Output @ 5 mA
```

Figure 1-4: Remote Contact-Closure Connector

Table 1-3: Remote Connector Pins

Remote Connector	
Control Type	Contact Closure
Connector	DB-9, female
Pin 1, TTL Level Input	0 to +5V, ≤ 100 mA. Input is protected. Use the +5 V output of pin 7 to drive this input via an external contact-closure relay. The state of the TTL input (HIGH or LOW) can be read and used as an external trigger input.
Pins 2, 4, 6, and 8	Ground
Pin 3 to Pin 9, Contact Closure	Normally open. Maximum voltage: 220 VDC; Maximum current: 2 A; Maximum power: 60 W. The state of the contact-closure relay (open or closed) is settable from a function key and using SCPI remote control.
Pin 5 to Pin 9	Reserved, normally-closed contact
Pin 7, +5 V Output	+5 V @ 5 mA. This output is provided to drive the TTL-Level input through an external contact-closure relay.

SCPI Programming

The AM700 uses SCPI-1993 as its primary control protocol. SCPI (Standard Commands for Programmable Instruments) is an emerging standard promoted by a consortium of test and measurement equipment vendors, including Tektronix. The intent of SCPI is to provide a consistent and standard command language for all test and measurement equipment. SCPI is layered on top of IEEE 488.2, and contains several 488.2 commands and status structures.

IEEE 488.2 Commands

Complete details of IEEE 488.2 commands and operation are found in IEEE STD 488.2-1992, *IEEE Standard Codes, Formats, Protocols, and Common Commands*. A brief summary of those implemented in the AM700 is given in the following tables.

Table 1-4: IEEE 488.2 Status Reporting Commands

Command	Name	Function
*CLS	Clear Status Command	Clears status data structures and forces the device to the Operation Completed Command Idle State and the Operation Complete Query Idle State.
*ESE NRf	Standard Event Status Enable Command	Sets the Standard Event Status Enable Register bits.
*ESE?	Standard Event Status Enable Query	Queries the contents of the Standard Event Status Enable Register.

FORMat

FORMat:PNAME STRING|CHARacter

Used with PROGRAM commands to set the <programe> type to be either string data or character data.)

GCONtrol

GCONtrol[1|2]:ANALog:LDISTortion ON|OFF

Suffixes: GCON1 Analog Generator A
GCON2 Analog Generator B

GCONtrol:ANALog:MODE HRESolution|HBW (high bandwidth)

GCONtrol:DIGital:MODE AES|DSP

GCONtrol:OUTput:STATE <Boolean>

HCOPY

HCOPY:ABORT

HCOPY:DESTination 'data handle'

'SYSTEM:COMMunicate:SERial1'
'SYSTEM:COMMunicate:SERial2'
'SYSTEM:COMMunicate:GPIB'
'UNIX'

HCOPY:DEVIce:COLor <Boolean>

Works for PostScript and TIFF output only.

HCOPY:DEVIce:LANGUage PCL|HPGL|POSTscript|ELQu-ality|INTERleaf


```

DISPlay[WINDow(1-4)]:TRACe[1|2]:STYLE:CSTATUS:SDIFFs
<Boolean>

DISPlay[:WINDow(1-4)]:TRACe[1|2]:STYLE:CSTATUS:SUBFr
ame A|B|BOTH

DISPlay[:WINDow(1-4)]:TRACe[1|2]:STYLE:PLOT:LINes
<Boolean>

DISPlay[:WINDow(1-4)]:TRACe[1|2]:STYLE:PLOT:POINTs
<Boolean>

DISPlay[:WINDow(1-4)]:TRACe[1|2]:X[:SCALE]:AUTO ONCE
DISPlay[:WINDow(1-4)]:TRACe[1|2]:X[:SCALE]:CENTer
<numeric_value>
DISPlay[:WINDow(1-4)]:TRACe[1|2]:X[:SCALE]:LEFT
<numeric_value>
DISPlay[:WINDow(1-4)]:TRACe[1|2]:X[:SCALE]:RIGHT
<numeric_value>
DISPlay[:WINDow(1-4)]:TRACe[1|2]:X[:SCALE]:SPACing
<LINear|LOGarithmic>

DISPlay[:WINDow(1-4)]:TRACe[1|2]:Y[:SCALE]:AUTO ONCE
DISPlay[:WINDow(1-4)]:TRACe[1|2]:Y[:SCALE]:BOTTom
<numeric_value>
DISPlay[:WINDow(1-4)]:TRACe[1|2]:Y[:SCALE]:CENTer
<numeric_value>
DISPlay[:WINDow(1-4)]:TRACe[1|2]:Y[:SCALE]:TOP
<numeric_value>
DISPlay[:WINDow(1-4)]:TRACe[1|2]:Y[:SCALE]:SPACing
<LINear|LOGarithmic>
    
```

Table 1-4: IEEE 488.2 Status Reporting Commands (Cont.)

Command	Name	Function
*ESR?	Standard Event Status Register Query	Queries the contents of the Standard Event Status Register. Reading the register clears it.
*SRE NRf	Service Request Enable Command	Sets the Service Request Enable Register bits.
*SRE?	Service Request Enable Query	Queries the Service Request Enable Register. Returns an NR1 that is the value of the service request enable register.
*STB?	Read Status Byte Query	Queries the status byte. Returns an NR1 that is the value of the status byte and the master summary message.

Table 1-5: Internal Operation Commands

Command	Name	Function
*IDN?	Identification Query	Queries the id of the AM700.
*RST	Reset Command	Does a AM700 reset.
*TST?	Self-Test Query	This is currently a "NO-OP" It is effect just tests the remote connectivity and does not change the operating state of the AM700.

The AM700 commands are based on SCPI VERSION 1994.0

SCPI Command Subsystems Implemented in the AM700

Many commands in the SCPI “Command Descriptions” are considered optional in that they depend on the capabilities of the SCPI instrument. Commands that are used will be implemented exactly as specified by using the SCPI defined commands. Certain commands, if used, require that other commands also be implemented. If a command is implemented where the instrument will not support all the SCPI alternative parameter values, a subset may be used. If the instrument does not support an alternative value of the complete set, it may generate an error on receipt.

Table 1-6: SCPI Subsystems Implemented in the AM700

Subsystem	Use in the AM700
CALCulate	Averaging and FFT parameters
CALIBration	Calibration start and Conversion Factors
DISPlay	Select displays and presentation styles for data and control the view windows and cursors
FORMat	Sets program names to either character or string data types.
HCOPY	Format and output screen dumps.
INPut	Select input parameters
INSTRument	Select Application
MMEMory	AM700 memory commands
OUTPut	Select output parameters
PROGram	Control the function programs
ROUTE	Select input routing
SENSE	Select input processing
SOURce	Select generator
STATus	Set and query the AM700 status registers
SYSTem	Control the communication parameters, hardcopy operations, and set the date/time

```
DISPlay:WINDow98:DISMiss:[ALL]
DISPlay:WINDow98:DISMiss:ONE
DISPlay:WINDow98:STATE?
DISPlay[:WINDow(1-4)]:TYPE <display_name>
DISPlay[:WINDow(1-4)]:FEED(1-10) <data_handle>
```

Default: Window 1 Feed is 'ame1'
 Window 2 Feed is 'ame2'
 Window 3 Feed is 'ame3'
 Window 4 Feed is 'ame4'
 Window 5 Feed is 'ame5'

```
DISPlay[:WINDow(1-4)]:CURSor[:POSition(1-2)]
<x-axis_value>
DISPlay[:WINDow]:CURSor:STATE <Boolean>
DISPlay[:WINDow(1-4)]:CURSor:CONStrain <Boolean>
DISPlay:WINDow60:STATE ON|OFF
DISPlay:WINDow99:STATE <Boolean>
DISPlay:WINDow99:TEXT[:DATA] <string>
DISPlay:WINDow99:TEXT:CLEAR
DISPlay[:WINDow(1-4)]:X[:AXIS]:JOIN <Boolean>
DISPlay[:WINDow(1-4)]:Y[:AXIS]:DUAL <Boolean>
DISPlay[:WINDow(1-4)]:Y[:AXIS]:JOIN <Boolean>
DISPlay[:WINDow(1-4)]:TRACe[1|2]:PERSistence
<Boolean>
```

Up to 10 traces may be supported but only 2 are presently used. TRAC1 is the top trace in a display and TRAC2 is the bottom (underneath) trace in a display.

```
DISPlay[WINDow(1-4)]:TRACe[1|2]:STYLE:CSTATUS:FORMAT
<char_data>
```

Suffixes: Sense suffixes are:

SENSe1	Analog A High Res A/D
SENSe2	Analog A High BW A/D
SENSe3	Analog B High Res A/D
SENSe4	Analog B High BW A/D
SENSe5	Digital SF1 Main
SENSe6	Digital SF2 Main
SENSe7	DSP Port A
SENSe8	DSP Port B
SENSe9	Digital Ref A
SENSe10	Digital Ref B

CSTReam[1|2]:FEED? <data_handle>

DISPlay

In some applications there may be more window numbers. Some of the following windows are application specific.

Windows 1–4 are the graphical displays.
 Window 5 is the Real Time display.
 Window 60 is the Generator Status display.
 Window 98 is the Notifier display.
 Window 99 is the dialog box display.

DISPlay:BRIGhtness <numeric_value> 0 is full dim; 1 is full bright.

DISPlay:ENABle ON|OFF

DISPlay[:WINDow(1-n|60|99)][[:STATe] <Boolean>

Table 1–6: SCPI Subsystems Implemented in the AM700 (Cont.)

Subsystem	Use in the AM700
TRACe	Query the displayed traces for name, number of points, and trace data
TRIGger	Start and stop the sweep
UNIT	Select units for input or output of certain parameters. Trace data does not follow units.

Table 1–7: New Command Subsystems Implemented in the AM700

Subsystem	Use in the AM700
AMEasure	Control measurement setups for Audio Analyzer, FFT Analyzer, Digital Interface Tester, and the Audio Monitor.
CMODE	Select operation modes within an application
CSTReam	Direct the two measurement channels into the CALC blocks.
GCONtrol	Control the Audio Generator

Table 1–8: SCPI Command Syntax Symbols

Symbols	Description
[]	Used to enclose one or more parameters that are optional when controlling the instrument. Omitting the optional element causes the default action to occur.
{ }	Use to enclose one or more parameters that may be included zero or more times.
?	Use to indicate a query by appending to the last keyword in a command. Not all commands have a query, and some commands are only queries.
	May be read as “or” and is used to separate alternative parameter options.
< >	Use to enclose a SCPI-defined parameter.
:	Used to separate elements of a SCPI command.

Table 1-8: SCPI Command Syntax Symbols (Cont.)

Symbols	Description
;	Used to separate SCPI commands in a command list.
,	Used to separate arguments in a SCPI argument list.

A query is formed by appending a question mark (?) to the last keyword in a command. Not all commands have a query form. Those that are events and do not set states do not. Some commands exist only as a query. A good example of that type of command is `SYSTEM:ERROR?` This command is used to find out if a system error has been spooled for reporting. Obviously, you can not set a system error.

To set the frequency value for a Continuous Wave signal send:

```
SOURce:FREQuency[:CW] 20000
or
sour:freq:cw 20000
```

Alternately, since CW is optional, the following command is also valid:

```
SOUR:FREQ 20000
or
sour:freq 20000
```

The query form of the `SOURce:FREQuency` command also has two valid forms, again with and without the optional CW:

```
SOUR:FREQ?
or
sour:freq?
```

```
'FWeight'    F Weighting
'LP15000'    15 kHz low pass
'HP400'      400 Hz high pass
```

```
CMODE:FFT FFT|MTONE
CMODE:FFT:MTONE[1|2]:AVECTOR?
CMODE:FFT:MTONE[1|2]:FILE:NAME 'file_name'
CMODE:FFT:MTONE[1|2]:FVECTOR?
CMODE:FFT:MTONE[1|2]:NTONES?
CMODE:FFT:MTONE[1|2]:RLENGTH?
CMODE:FFT:MTONE[1|2]:SOURCE GENERATOR|FILE
CMODE:FFT:MTONE[1|2]:SUFFIX <numerical_value>
CMODE:FFT:WINDOW:FILE:NAME <file_name>
CMODE:FFT:WINDOW:FILE:LNAME?
CMODE:DAUDIO:SRATE:SOURCE CSTATUS|CLOCK
CMODE:SOUND:STATE ON|OFF
CMODE:SOUND:SOURCE ANALOG|DIGITAL
CMODE:TRIGGER:LEVEL <numeric_value>
CMODE:TRIGGER:MODE AUTO|NORMAL
CMODE:TRIGGER:SLOPE RISING|FALLING
CMODE:TRIGGER:SOURCE CHANNEL1|CHANNEL2|EXTERNAL
```

CSTReam[1|2]

Suffixes: CSTReam1 is measurement channel 1;
CSTReam2 is measurement channel 2

```
CALibration:OUTPut:POWer:REFErence:RESistance
<numeric_value>
```

```
CALibration:VOLTage:FS <numeric_value>
```

CMODE

The CMODE:ENABLE commands control the display of the real-time measurement readout in the Real Time window. The CMODE:FFT commands are used to switch between FFT and Multitone modes in the FFT Analyzer. CMODE:SOUND commands are for turning the sound on or off to the headphones and selecting the source of the signal as the digital input or the analog input. CMODE:TRIGGER commands are used in the Audio Monitor to control the triggering features of that application.

```
CMODE:ENABLE:CHANnel[1|2] <Boolean>
```

```
CMODE:ENABLE:IMD <Boolean>
```

```
CMODE:ENABLE:STER <Boolean>
```

```
CMODE:ENABLE:THD <Boolean>
```

```
CMODE:ENABLE:WOW <Boolean>
```

```
CMODE:FILTer:STATE <Boolean>
```

```
CMODE:FILTer:TYPE[1|2] 'filter_name'
```

```
CMODE:ENABLE:CHANnel[1|2] ON|OFF
```

Suffixes:	TYPE1	Channel 1 Filter
	TYPE2	Channel 2 Filter

Parameters:	'NONE'	No filter type selected
	'CCIR468'	CCIR 468
	'CCIRARM'	CCIR ARM
	'AWeight'	A Weighting
	'BWeight'	B Weighting
	'CMessage'	C Message
	'CWeight'	C Weighting

Keyword Suffixes

When there are multiple choices for a keyword a suffix number is used to differentiate between the choices. A good example of this use is seen in the selection of signal sources from the AM700 generator. There are four analog signal source choices and four digital signal source choices so the SOURce command has a range of suffixes from 1 to 8. Omitting the suffix is defaulted to 1.

Suffixes:	SOURce1	Analog generator HR A
	SOURce2	Analog generator HR B
	SOURce3	Analog generator HB A
	SOURce4	Analog generator HB B
	SOURce5	Digital generator DSF 1
	SOURce6	Digital generator DSF 2
	SOURce7	Digital Signal Processor A
	SOURce8	Digital Signal Processor B

Mnemonic Keyword

The AM700 accept only the exact short form or the exact long form of a mnemonic keyword. Sending a keyword that is not either causes a command error to be generated.

SCPI has a rule for generating a mnemonic keyword. The long form mnemonic is generated from either a single word or a phrase. If a single word is used, then that word become the mnemonic. If a phrase is used, then the mnemonic is the first letter of each word and the entire last word.

Example:

```
relative velocity = RVELOCITY
```


AMEasure[1-5]:MODE?

AMEasure[1-5]:STATE <Boolean>

AMEasure[3-5]:MTONE:FUNCTION LEVEL|CROSstalk|LDIF-
ference|PDIFference|MDISortion

AMEasure[3-5]:MTONE:INPut CHANnel1|CHANnel2

Default: AME1:MODE FFT
AME2:MODE FFT
AME3:MODE MTONE
AME4:MODE MTONE
AME5:MODE MTONE

AMEasurement (Digital Interface Tester)

DIT has AME command support for four separate measurements.
They are:

Measurement 1: Bit Activity
Measurement 2: Channel Status
Measurement 3: Eye Diagram
Measurement 4: Jitter Spectrum

AMEasure:DAUDIO:CSTATUS[1-2]:DATA?

AMEasure:DAUDIO:PPJitter?

AMEasure[1-4]:STATE ON|OFF

AMEasurement (Audio Scope)

AMEasurement[1-2]:STATE ON|OFF

SOURCE[1-8]:FREQUENCY <numeric_value>
:MODE CW|FIXed|SWEep|LIST
:RESolution <numeric_value>
:START <numeric_value>
:STOP <numeric_value>

you can create a command with multiple elements reference to the same subnode.

Example: SOUR1:FREQ:MODE SWE;STAR 200;STOP 10000

This sets source 1 (Analog A high resolution generator) for a sweep signal starting at 200 Hz and ending at 10,000 Hz. MODE, START, and STOP are referenced to SOURCE1:FREQUENCY.

To specify the type of signal needed you need to send a SOURCE:FUNCTION:SHAPE command.

SOUR1:FUNC:SHAP SIN

to send a sine wave signal from Analog Generator A.

This still doesn't turn on the generator. Another command is needed for that.

GCON:OUTP:STAT ON

These can be all be placed in the same command as shown in the following example.

SOUR1:FREQ:MODE SWE;STAR 200;STOP
10000;;SOUR1:FUNC:SHAP SIN;;GCON:OUTP:STAT ON

Notice that the end of one element referenced back to the root is separated from the next root command by a ;; combination.

AM700 Setup

A systematic approach to setting up the AM700 for operation will help in keeping the task organized. A suggested approach is included here to illustrate one method. Most of the choices may be made in any order without concern when changing between applications. However, setting up a particular display must be done after the application is started as the displays setups are application specific.

Generator Controls

1. If you are going to use the AM700 generator to provide the test signal, make the generator choices first. This may include the following tasks.
 - a. Selecting the test signal (SOURCE:FUNCTION:SHAPE commands).
 - b. Enabling an output channel (OUTPUT commands).
 - c. Setting up a frequency and/or voltage sweep (SOURCE:FREQUENCY and SOURCE:SWEEP commands).
 - d. Making up a list used for sweeping or multitone signals (SOURCE:LIST, SOURCE:VOLT, and SOURCE:MULTITONE commands).
 - e. Starting and stopping a sweep signal (TRIGGER commands).
2. If the analog generator is used, the following choices must be made.
 - a. Selecting high bandwidth or high resolution mode for the generator (GCONTROL:ANALOG commands).
 - b. Setting the generator for low distortion mode or not (GCONTROL:ANALOG commands).

```
AMEasure:SETTled:SEPARation:RESolution
<numeric_value>
```

```
AMEasure:SETTled:SEPARation:TOLerance
<numeric_value>
```

```
AMEasure:SETTled:THD:RESolution <numeric_value>
```

```
AMEasure:SETTled:THD:TOLerance <numeric_value>
```

```
AMEasure:SETTled:THDN:RESolution <numeric_value>
```

```
AMEasure:SETTled:THDN:TOLerance <numeric_value>
```

```
AMEasure:SETTled:TIMEout <numeric_value>
```

```
AMEasure:SETTled:TYPE FLAT:EXponential
```

```
AMEasure:SETTled:VARIation:AMOUNT <numeric_value>
```

```
AMEasure:SETTled:VARIation:TYPE DEpendent|INDEpen-
dent|BOTH|EITHER
```

```
AMEasure[1-4]:SWEep:DIRection RISING|FALLing
```

```
AMEasure[1-4]:SWEep:FREQuency:DELta <numeric_value>
```

```
AMEasure[1-4]:SWEep:LEVel:DELta <numeric_value>
```

```
AMEasure[1-4]:SWEep:MODE FREQuency|LEVel
```

```
AMEasure[1-4]:SWEep:STATE <Boolean>
```

AMEasurement (FFT Analyzer)

FFT has AME command support for five separate measurements. They are:

Measurement 1: FFT on Measurement Channel 1

Measurement 2: FFT on Measurement Channel 2

Measurement 3: Configurable Multitone

Measurement 4: Configurable Multitone

Measurement 5: Configurable Multitone


```

AMEasure[1-4]:REGulation:FREQuency:MODE <char_data>
AMEasure[1-4]:REGulation:FREQuency:POINTs
<numeric_value>
AMEasure[1-4]:REGulation:FREQuency:UPPer
<numeric_value>
AMEasure[1-4]:REGulation:LEVel:LOWer <numeric_value>
AMEasure[1-4]:REGulation:LEVel:UPPer <numeric_value>
AMEasure:SETTled:COUNT <numeric_value>
AMEasure:SETTled:CROSstalk:RESolution
<numeric_value>
AMEasure:SETTled:CROSstalk:TOLerance <numeric_value>
AMEasure:SETTled:DELay <numeric_value>
AMEasure:SETTled:ENABle <Boolean>
AMEasure:SETTled:FREQuency:RESolution
<numeric_value>
AMEasure:SETTled:FREQuency:TOLerance <numeric_value>
AMEasure:SETTled:IMD:RESolution <numeric_value>
AMEasure:SETTled:IMD:TOLerance <numeric_value>
AMEasure:SETTled:LDIFFerence:RESolution
<numeric_value>
AMEasure:SETTled:LDIFFerence:TOLerance
<numeric_value>
AMEasure:SETTled:LEVel:MINimum <numeric_value>
AMEasure:SETTled:LEVel:RESolution <numeric_value>
AMEasure:SETTled:LEVel:TOLerance <numeric_value>
AMEasure:SETTled:PDIFFerence:RESolution
<numeric_value>
AMEasure:SETTled:PDIFFerence:TOLerance
<numeric_value>

```

- c. Deciding if Analog A and Analog B will have the same signals (B follow A) or different signals (SOURce:FOLLow commands).
 - d. Generator output impedance and floating or grounded audio common (OUTPut commands).
3. For the digital generator, the following are some of the choices that must be made.
- a. AES or DSP mode for generating the signals (GCONtrol:DIGital commands).
 - b. Sampling rate of the digital signal (SOURce:DAUDio commands).
 - c. Word size for the digital signal (SOURce:DAUDio commands).
 - d. Channel status — professional or consumer (SOURce:DAUDio commands).
 - e. Digital interface clock frequency (OUTPut:DAUDio commands).

Signal Input Configuration

1. The input signal routing must be determined.
 - a. For the digital signal inputs, the external choices are the front panel XLR connector, the rear panel BNC connector, or the rear panel optical connector; internally you may connect directly to the AES generator (ROUTE:CLOSe commands).
 - b. For the analog signal inputs, the routing is either the XLR external connectors or internal connections directly to the analog generators (ROUTE:CLOSe commands).
 - c. Additionally for the analog inputs, the gain setting must be set or AUTO enabled. The input impedance must also be set (INPut commands).

2. Finally, the input signal to the measurement channels must be assigned (CSTReam commands).

Application Control

1. Selecting the AM700 application to run depends on your measurement requirements. The choices are the FFT analyzer, the Audio Analyzer, the Digital Interface Tester, and the Audio Monitor (INSTrument commands).

After the application is selected, application specific choices must be made. These include:

- a. Selecting measurements and assigning views to the measurements (AMEasure commands).
 - b. Selecting the other application and measurement view specific choices. These include: averaging, window type, cursors, number of measurement views, number of traces in a measurement view, and scaling (CALCulate and DISPlay:WINDow commands).
2. Selecting the measurements that are enabled in the Real-Time measurement view (CMODE commands).

File Management

1. Navigating the AM700 filing system is needed when you need to copy files to and from the disk drive. This is needed when function programs (SCPI programs written to perform specific task) are used (MMEMory commands).
2. Running the function programs and setting up timed functions are other operating choices that must be done. From within a function, other functions may be called to perform many automated measurements (PROGram commands).

AM700 SCPI Commands

User SCPI Commands

AMEasurement (Audio Analyzer)

```
AMEasure[1-4]:STATe <Boolean>
AMEasure[1-4]:HISTory <numeric_value>
AMEasure[1-5]:MODE <char_data>
AMEasure[1-4]:REFerence:SET <event>
AMEasure[1-4]:REFerence:STATe <Boolean>
AMEasure[1-4]:XY:TRACkgen <Boolean>
AMEasure[1-4]:XY:X:INPut <char_data>
AMEasure[1-4]:XY:X:FUNCTion <char_data>
AMEasure[1-4]:XY:Y:INPut <char_data>
AMEasure[1-4]:XY:Y:FUNCTion <char_data>
AMEasure[1-4]:REGulation:COUNT <numeric_value>
AMEasure[1-4]:REGulation:ERRor <numeric_value>
AMEasure[1-4]:REGulation:FUNCTion <char_data>
AMEasure[1-4]:REGulation:INPut <char_data>
AMEasure[1-4]:REGulation:TARGeT <numeric_data>
AMEasure[1-4]:REGulation:GENerator <char_data>
AMEasure[1-4]:REGulation:FREQuency:LOWer
<numeric_value>
```

Other Instrument Operations

The AM700 has supporting features that are also controllable via remote control. These features are ones that are not accessed often. They include:

1. Printing screen displays and files (HCOPY commands).
2. Calibration selections (CALibration commands).
3. Setting communications parameters for GPIB and RS232, setting the time for the internal clock, and checking for system errors (SYSTEM commands).
4. Setting AM700 registers for status reporting (STATUS and GPIB interface commands).

Generator, Output, and Source Controls

AM700 SCPI Commands

GCONtrol Subsystem

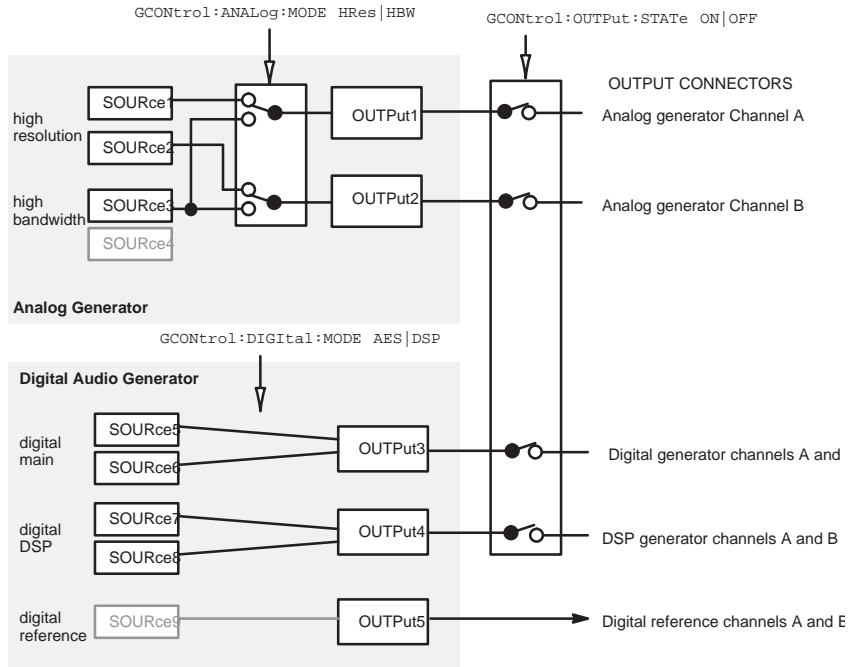


Figure 1-5: Generator Control Model

Generator Mode Commands The following commands control the generator signal source. The first sets either the high resolution generator or the high bandwidth generator for the analog signal. The second determines whether the digital mode will be from the digital main generator (AES) or the digital signal processor (DSP), and third turns the generator on or off.

```
usage: panel beep [<count>]
       panel led {<led name>|all}
       {on|off}
       panel led {<led name>|all} duty
       <brightness>
```

The <led name> is one of: analyzer, average, clear_menu, configure, copy, cursor, digital, display, expand, fft, filter, freeze, function, gen_control, gen_onoff, help, limits, menu, monitor, move, other, rescale, sound, storage, sweep_run, user.

Command: rm <file> ...

Usage: Removes one or more named files¹. Error reports are not fully implemented for removing files.

Command: scpi

Usage: Specifies that the command following is a SCPI command. The Tcl parser passes it immediately to the SCPI command interpreter.

¹ Further file access (and 'rm' for that matter) can be accessed through the 'scpi' command and the MMEMory subsystem.

```
GCON:ANAL:MODE HRES|HBW
GCON:DIG:MODE AES|DSP
GCON:OUTP:STAT ON|OFF
```

OUTPut Commands

OUTPut commands are used to set the output impedance, state, and common state for the analog outputs. For the digital signals, these commands control many of the interface functions: clock frequency and phase, voltage, jitter, and filtering.

Analog Output Commands. The OUTPut analog commands are the following:

```
OUTP:IMP 10|150|600
OUTP:COMM FLO|GRO
```

Digital Output CommandsThe OUTPut digital commands for the digital interface include the following:

```
OUTP:DAUD:INT:CLOC:PHAS:ADJ <-64 to +63>
OUTP:DAUD:INT:VOLT:BAL <voltage_value>
OUTP:DAUD:JITT:STAT ON|OFF
```

One command is common for both the digital and analog generators. That is the command that enables or disables a single generator channel.

```
OUTP:STAT ON|OFF
```

SOURCE Commands

Commands in the source subsystem control the signal selections from the AM700 generators. The commands are divided between those used to control the digital generator and those used to control the analog generator and suffixes are used to provide the necessary identification.

Suffixes:	1	Analog generator HR channel A
	2	Analog generator HR channel B
	3	Analog generator HB channel A
	4	Analog generator HB channel B
	5	Digital generator DSF1
	6	Digital generator DSF2
	7	Digital Ref A
	8	Digital Ref B

The generator output for Analog high resolution may be different signals on each channel. For high bandwidth, both channels output either the same signal or silence.

Setting the type of signal output wanted from the generator is done using commands of the following syntax:

```
SOURCE:FUNCTION:SHAPE SINusoid|JSINE|
TBURst|CIMD|SIMD|PRNoise|PCHirp|POLarity|
TPolarity|MTONE|USER
```

where each of the shape parameters is a different signal type available from the generator. Not all signals are available from all the different sources. Table 1–9 provides the information about what is available from each generator.

```
condition <conditionNr> wait
is_clear

condition <conditionNr> wait
change

condition <conditionNr> wait
change is_set

condition <conditionNr> wait
change is_clear
```

The <conditionNr> is one of those documented. Things like ‘sweeping’ which is number 103. The usage string should be self-explanatory.

The <value> argument sets the condition to that value. A nonzero value means the condition is set, and a zero value means it is clear.

Additionally, the conditions numbered 9100 through 9107 are user-modifiable conditions. These are the only ones that the ‘set’, ‘clear’, or <value> arguments can be used for. The user can use these to signal between functions concurrently or consecutively running.

Command: event

Usage: Usage string returned from Tcl:

```
usage: event <eventNr> trigger
event <eventNr> wait
```

The <eventNr> is one of those documented.

The user-modifiable events (on which ‘trigger’ can be used) are numbered 9200 through 9207.

Command: panel

Usage: Usage string returned from Tcl:

Usage: Delay for the specified seconds and optional tenths and hundredths of seconds

Command: `notifier <message> <buttons>`
`[notifier|warning|error] [wait]`

Usage: Displays the message in a notifier on the screen, requiring one of the buttons to be pressed. The `<buttons>` argument is a semicolon-separated list of button names. There are different kinds of notifiers, though 'notifier' 'warning' and 'error' look alike to the Tcl parser.

If 'wait' is specified, the command returns only after a button has been pressed, and the return value of the command is the number of the button that was pressed (the numbering starts with 1).

If the "disp:wind98:dism:one" or "disp:wind98:dism:all" (commands are documented in Section 3 of the AM700 Programmers Manual) command is given, causing a notifier to pop down, the return value from a waiting notifier command is -1.

Command: `condition`

Usage: Usage string returned from Tcl:

```
usage: condition <conditionNr> set
       condition <conditionNr> clear
       condition <conditionNr> <value>

       condition <conditionNr> is_set
       condition <conditionNr> is_clear
       condition <conditionNr>

       condition <conditionNr> wait
is_set
```

Table 1-9: Signals Available From the Generators

Signal	Analog High Resolution Generator	Analog High Bandwidth Generator	Digital AES	Digital DSP
sine	Yes	Yes	Yes	Yes
tone burst	Yes	Yes	Yes	Yes
smpte imd	Yes	No	Yes	Yes
ccif imd	Yes	No	Yes	Yes
shaped noise	Yes	No	Yes	Yes
chirp	Yes	Yes	Yes	Yes
polarity	Yes	No	Yes	Yes
tek polarity	Yes	Yes	Yes	Yes
user	Yes	Yes	Yes	Yes
multitone	Yes	Yes	Yes	Yes
jittered sine	No	No	Yes	No

AM700 SCPI Model

The AM700 functional areas are divided as to task to perform. In a like manner SCPI commands are organized to command the task required in each functional area. The AM700 is logically divided between its hardware functions needed to acquire signals for processing and the display of the results and its software functions that do the signal processing needed to produce the measurement results. The measurement results in most cases, consist of a display trace of the acquired data after processing and numerical values of key features of a measurement. Examples of display numerical value are the amplitude at a frequency, the frequency at a cursor location, the THD + N, etc.

AM700 Signal Selection and Routing

Signals to acquire must be selected and routed to the CALCulate blocks for measurement. Since the AM700 has multiple input choice for analog and digital signals, the routing commands take on

suffixes and channel_list numbers to provide the control needed. Figure 1-8 illustrates the selection paths available in the AM700.

An illustration (see Figure 1-6) of how the AM700 is modeled for SCPI control is a useful start in determining which SCPI System and Subsystem commands are used to control the AM700 functions. After that connection is made, the actual SCPI commands required to perform a measurement task may be developed.

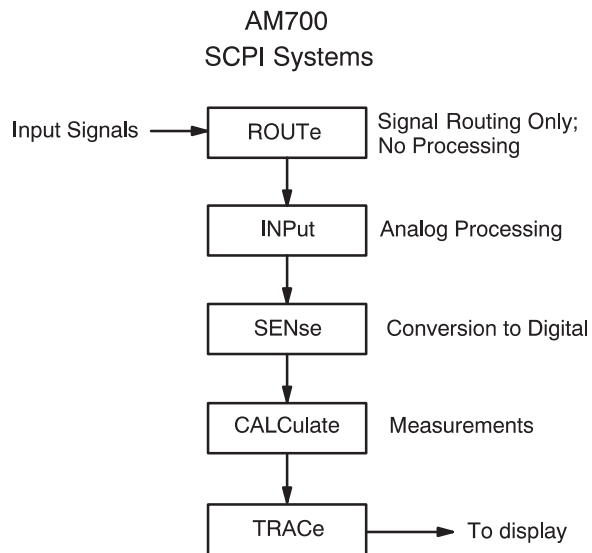


Figure 1-6: AM700 SCPI Systems for Measurements

AM700 Analog Signal Path

In the AM700, the hardware of the analog signal path is organized in functional blocks. These blocks are assigned SCPI System commands related to the functions to be done as shown in Figure 1-7. Signal Routing is shown in Figure 1-8.

pair of numbers separated by a hyphen (2-5) specifying 2, 5 and all the numbers in between.

Example: `PROG:TIM:SET '30 * 1,15 5 *',
'45 17 19 5 4'`

This set the times for the timed function at every half past the hour, on the 1 and 15 of the month May for every day of the week that applies and 1745, 19 May, Thursday.

Use the `PROG:TIM:ADD` command to add new times to the same named function and use the `PROG:TIM:CLE` command to clear all the times for a named function. If you want other timed functions to run, change the named function and use the `PROG:TIM:SET` command to set a new cron_string of set of cron_strings for the newly named command.

The `PROG:EXPL` commands are used to run a explicitly named program. This command is used when you want to set the cron_string for a specific named function. See the Programmer Manual for further details on the `PROG` commands and their use in running functions.

Example: `PROG:EXPL:TIM <prog_name>,<cron_string>`

Tcl Programming Changes

Here are some additions, changes, and limitations of the AM700 Tcl environment as it is used for Functions.

Added commands:

Command: `sleep <seconds> [<tenths>
[<hundredths>]]`

- Use the `PROG:STAT RUN` command to start the named function. Send `PROG:STAT STOP` to stop the running function if it is not self terminating.

Timed Functions

A timed function may be set to run at certain times by setting a time for it to start using the `PROG:NAME` commands.

The `Prog:Name <prog_name>` command names the file that a following `TIMed:SET` command associates with the `cron_string` or strings that are given.

```
PROG:NAME 'function_name string'
```

```
PROG:TIM[:SET] {'cron_string'[, 'cron_string']}
```

```
PROG:TIMed:ADD 'cron_string'[, 'cron_string']
```

Each line in a crontab file consists of five fields, separated by spaces or tabs, as follows:

- Minutes field, in the range 0 through 59.
- Hours field, in the range 0 through 23.
- Day of the month, in the range 1 through 31.
- Month of the year, in the range 1 through 12.
- Day of the week, in the range 0 through 6.
- Sunday is day 0.

A cron string is `MIN HOUR DAY_of_MONTH MONTH and DAY_of_WEEK` as in `'30 15 15 5 1'`. An `*` in a field means to do it on all occurrences. A field may also be a comma separated list specifying multiple occurrences for the timed program to run.

Example `'0 0,12 * *'` specifies a time of midnight and noon every day of the month and every day of the week. A field may also be a

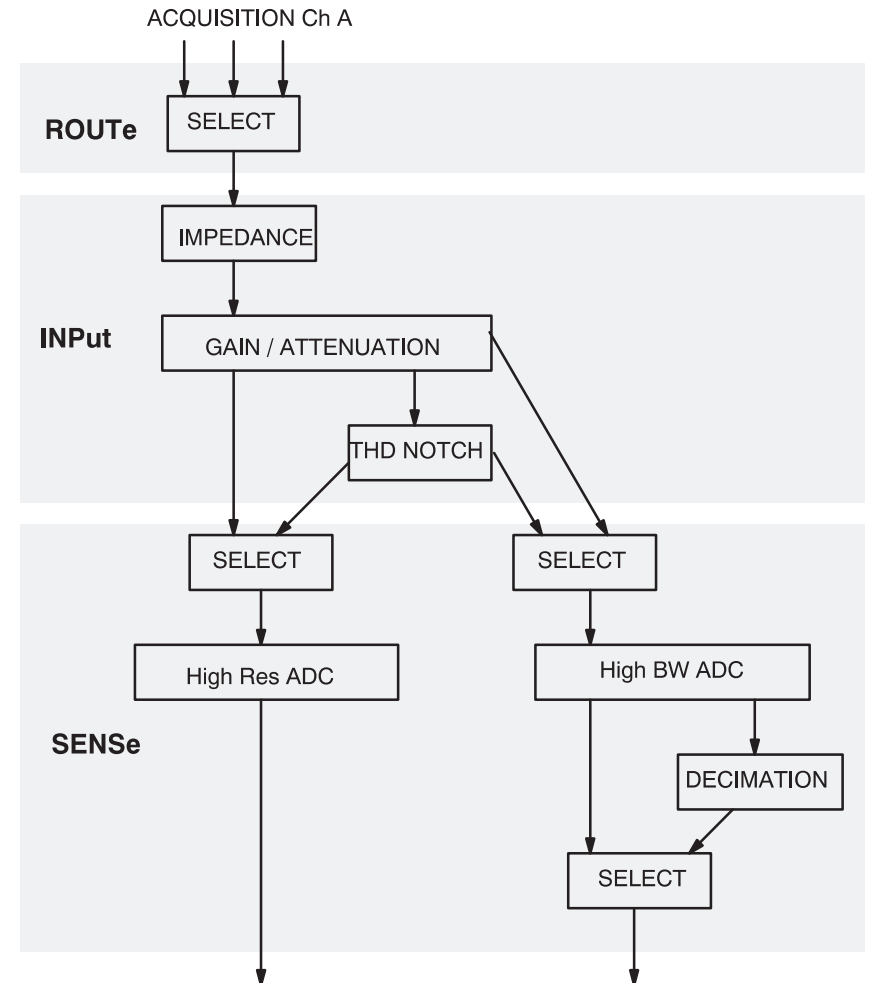


Figure 1-7: Analog Hardware SCPI Subsystem Commands

Some of the hardware choices are user selectable depending on the measurement application in use. Examples of these choices are the channel to be acquired and whether the measurement will in High Resolution Mode or in High BW Mode. Other of the hardware choices are made as required by the measurement application that is selected by the user.

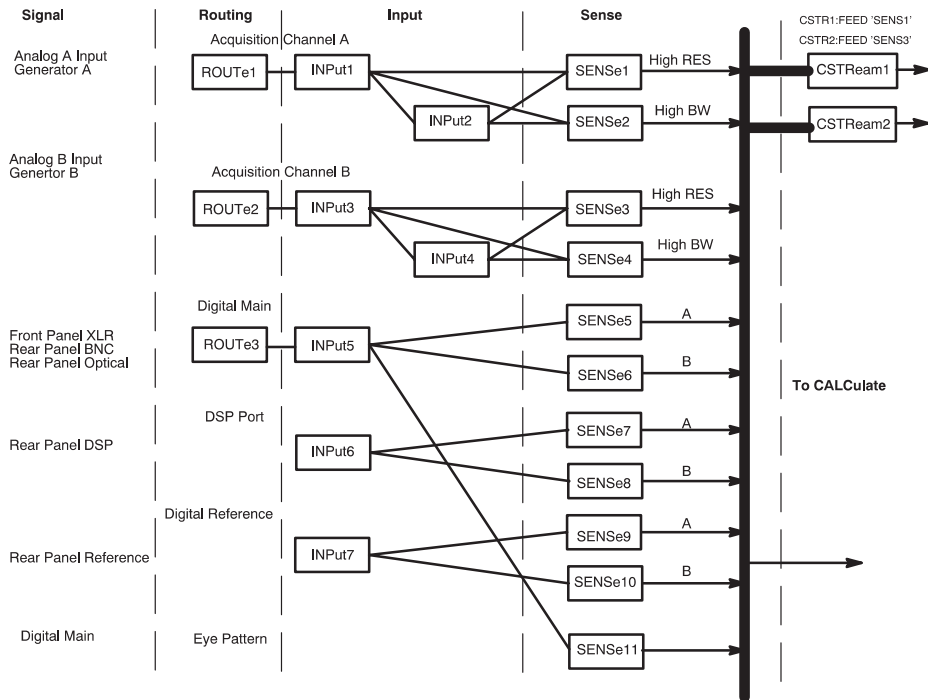


Figure 1-8: Signal Routing and Selection

Route Subsystem

The ROUTE Subsystem commands are used for switching signals in the input. No processing is done. To use the ROUTE commands, selectable routes must have a numerical assignments.

In the AM700 the route and close suffix numbers are as follows:

```

generated.
#DSF2 follows DSF1 by default.
scpi SOUR5:FUNC:SHAP CIMD
#
#Make sure the digital input path is the
#front panel XLR connector
scpi ROUT3:CLOS 6
#
#Turn on the generator signal.
scpi GCON:OUTP:STAT ON
#
#Start the Audio Monitor application
scpi INST:SEL MON
#
#Display two measurement windows; view 1 is
displayed by default.
scpi DISP:WIND2:STAT ON
#
#The Function Control Panel remains displayed
unless #it is cleared by the front panel
control
puts stdout "Press Clear Menu to see the
application display"
    
```

Running a Function

From the front panel, stored functions may be recalled and ran using menu choices called up by the Function button. Using the SCPI commands requires the following steps.

1. Send `PROG:CAT?` to get a comma separated list of the available functions. The names are recovered both from the `ROM:/FUNCTION` and the `NVRAM:/FUNCTION` directories. To avoid possible confusion, you should not have the same named functions in both directories.
2. Use the `PROG:NAME` command to select the function to run.

- If double quote marks are used in a SCPI command, they must be escaped by a backslash (“string\”) in order to pass them to SCPI using the Tcl scpi command.
- Precede all SCPI commands with the string “scpi”.

The Tcl parser looks first to see if it is a Tcl command that it knows about; if so, it runs it as a Tcl command. If not, the parser looks for an *, a ?, or a : in the command lines to see if it is a SCPI command. Putting the letters “scpi” in front of the SCPI commands immediately tells the Tcl parser that it is a SCPI command and passes it to the SCPI parser.

The example function program shown uses some of the commands that may be needed to change application, signals, and displays. Some control changes may not be required, because the default setting is correct for the function. You may use the *RST command to set everything to its default state or the state of controls may be explicitly set to avoid the possibility that the control is not in the default setting.

The following program uses the AM700 digital generator to supply the signal to the AM700 FFT application. The front panel XLR digital output connector must be connected to the front panel XLR digital input connector to complete the setup. In the sample function, the first line labels the function and the second line provides some visual feedback to the front panel in the Function Output window that appears in the Function control panel display.

```
#label: Digital Generator CCIF IMD
puts stdout "Digital CCIF IMD"
#
#Set the AM700 to its default states.
scpi *RST
#
#Select the digital generator for measurement
#CSTR1 has DSF1 and CSTR2 has DSF2
scpi CSTR1:FEED 'sens5'
scpi CSTR2:FEED 'sens6'
#
#Select the CCIF IMD test signal to be
```

ROUTE suffixes:

ROUTE1	Analog channel 'A'
ROUTE2	Analog channel 'B'
ROUTE3	AES/EBU digital audio

CLOSE <channel_list> indices:

CLOSE1	Analog input channel 'A'
CLOSE2	Analog input channel 'B'
CLOSE3	Analog generator channel 'A'
CLOSE4	Analog generator channel 'B'
CLOSE6	Balanced (FP) digital audio
CLOSE7	Unbalanced (RP) digital audio
CLOSE8	Optical (RP) digital audio
CLOSE9	AES (Internal) digital audio

The ROUTE Subsystem command has the following syntax:

Syntax: ROUTE[1|2|3]:CLOSE <channel_list>
or
ROUTE[1|2|3]:CLOSE:STATE?

Only certain closures go with a given route: The acceptable routes and closures are:

```
rout1:clos 1|3
rout2:clos 2|4
rout3:clos 6|7|8|9
```

Example: The Close command allows specific individual channels to be closed or queried. To select the generator as the signal source for the Analog A input, the command is:

ROUT1:CLOS 3

The `ROUT2[1|2|3]:close:state?` query returns the `<channel_list>` number for the designated route. If all the specified channels cannot be closed (trying to close two channels at the same time if that is not allowed) an execution error is reported. See the ROUTe Subsystem Command description in the *AM700 Programmer's Reference Manual* for complete information on the route command as used in the AM700.

INPut Subsystem

Suffixes: Suffixes are:

INP1	Analog acquisition channel 'A'
INP2	Analog acquisition channel 'B'

The Input Subsystem commands set the conditions prior to sampling. Here the choices of input impedance and input range are made. The input choices are also user selectable from within some measurement applications. The choices of selecting the filtered output are not a measurement choice that will be made as needed by the application. Selections for the notch filter range and use are not selectable via remote control. The syntax for the Input commands is as follows:

Syntax: `INPut:RANGe 0.0870 to 173.616`
`INPut:IMPedance 150|600|200000`

Range. The AM700 treats attenuation and gain as joined parameters that define the input range setting. A range window has a dynamic range of approximately 100 dB, but the location of the range window may be positioned within overall dynamic range of approximately +44 to -122 dBu. The numeric value in the Range command is the level at which the input signal will clip (the maximum signal level for the range window). The default range setting is AUTO. In AUTO, the input signal is examined and the

`MMEM:CAT?`

This returns a comma separated list of the directories and files in the directory and the number of bytes they contain.

Copying a file from a floppy disk into the nvram function file is done using the `MMEMemory:COPY` command as follows:

`MMEM:COPY`
`'dos:/filename', 'nvram:/function'`

This copies the dos file into the function directory with the same name as on the dos disk.

Functions

Writing a Function

Functions may be created externally to the AM700 using Tcl programming and loaded via the floppy disk drive into the file system of the AM700. Front panel menus for loading files into the internal file system are called up by the Storage button. Functions may also be loaded through SCPI commands using the PROGram subsystem commands.

Sample Function Program. A function program may be very simple or it may contain many steps. A running function program is interpreted by the Tcl parser. Some Tcl formatting information is required. A few points are mentioned here to show why they are used in the sample. Full information on Tcl programming is found in Part 1 of *Tcl and the Tk Toolkit*: Addison-Wesley Publishing Company, 1994.

- Tcl comments are preceded by a # symbol as the first non-blank character in the line. The # symbol appearing in the line at other locations is treated as any other character.

Note that this syntax places some restrictions on the <file_name> (for example, commas are not allowed).

The AM700 allows an optional mass storage unit specifier (a logical directory name) with any file name given to the MMEMory commands. The syntax of the file name is:

```
<[device:]{/path_name/path_name/}<file_name>
```

```
'dos:/directory/filename '  
'nvram:/directory/filename '  
'rom:/directory/filename '
```

The device and path portions are optional and not needed if the file name is in the current working directory. “device” can be one of “rom”, “nvram”, or “dos”. Legal dos file names are permitted. That is a file name of eight characters maximum length followed by a file extension up to three characters in length. Filenames are not case sensitive and must be single quoted in the SCPI commands. An example is:

```
MMEM:DEL 'nvram:/function/usrtone1.ton'
```

Moving Between Directories. Changing directories is done using the MMEMemory:CDIRectory command as follows:

```
MMEM:CDIR 'dos:/'  
MMEM:CDIR 'nvram:/function'  
MMEM:CDIR 'rom:/function'  
  
MMEM:CDIR '...'
```

To list the files in a directory use the MMEMemory:CATalog? query. First change directory to the one you are interested in, then cat the directory.

```
MMEM:CDIR 'nvram:/function' or  
MMEM:CDIR 'dos:/'
```

range window is positioned to produce a useful measurement on the applied signal. Setting the range to a specified value turns off AUTO.

Syntax: INPUT:RANGE:AUTO <Boolean>|ONCE
INPut:RANGe <numeric_value>

Example: The first of the two following examples turns autoranging on; the second sets the signal clipping level at +35 dBu.

```
INPut:RANGe:AUTO ON  
INPut:RANGe 43
```

Range: The numerical_value for range is 0.0870 to 173.616 volts in 12, 6 dB steps:

```
-18.9911 0.0870  
-12.9911 0.1736  
-6.9911 0.3464  
-0.9911 0.6911  
5.0089 1.3789  
11.0089 2.7512  
17.0089 5.4893  
23.0089 10.9527  
29.0089 21.8535  
35.0089 43.6034  
41.0089 87.0003  
47.0103 173.6160
```

Setting the value for a number that is in between the step settings defaults the range to the next highest valid range step. For example, if you set it for 2 volts, the actual setting will be 2.7512 for clipping at approximately 11 dBu.

Suffixes: INPut1 Analog acquisition channel 'A'
 INPut3 Analog acquisition channel 'B'

The Input Subsystem commands set the conditions prior to sampling. In the AM700. Here the choices of input impedance and input range are made. The input choices are user selectable from within a measurement application. The syntax for the Input command is as follows:

Syntax: INP[1|3]:RANG <numeric_value>:IMP
 150|600|200000

IMPedance. Another choice in the Input subsystem is the input impedance setting. The Input command in the AM700 uses suffixes to designate the input that has the impedance setting applied. Syntax for the Impedance command is as follows:

Syntax: INPut:IMPedance <numeric_value>

Example: This command sets the input impedance for Analog A to 600 ohms.

```
INP1:IMP 600
```

Range: The input impedance choices are: 150, 600, and 200000 ohms with 200000 being the power on default.

SENSE Subsystem

The Sense block samples the incoming data from input and performs pre-calculate signal conditioning. Sense must deal with analog signal input and digital signal input.

Suffixes: SENSE1 Analog 'A' high resolution
 SENSE2 Analog 'A' high bandwidth
 SENSE3 Analog 'B' high resolution
 SENSE4 Analog 'B' high bandwidth

NOTE

Diskettes used in the disk drive must be purchased DOS formatted or formatted for 1.44 Megs using a DOS machine.

Once past the device specification, the name looks pretty much like a UNIX file name. Slashes separate the path-name components.

Mass MEMory provides mass storage capabilities for the AM700. Mass storage is either internal or external and the AM700 supports both.

The CLOSe, FEED, NAME, and OPEN commands are used to stream data from anywhere in the data flow into a file; this is particularly useful for saving HCOPY output.

Mass storage media may be formatted in one of a number of standard formats. The AM700 does not support the SCPI mass storage unit specifier <msus>.

AM700 File Structure

The upper level of the AM700 files comprise ROM, NVRAM, and DOS logical directories. Under those, other directories or files may exist. Certain directories are accessible by the user for storage use or information. Certain other directories are accessible by AM700 operating system firmware only.

File Names. The <file_name> parameter in the MMEMory subsystem is a string. The contents of the string are dependent on the needs of the format of the mass storage media. In particular, the file name may contain characters for specifying sub-directories (that is, \ for DOS, / for UNIX). File names may be absolute, rooted, or relative. Absolute file names use the complete name with device and total path to the file. Rooted file names use the path within a designated device and may be used after changing directory to that device. Relative file names are assumed to be in the current working directory.

data held in a trace. Different traces are selected by trace name; not suffixes. Trace commands are also application specific and will change with the Instrument selection.

A useful query in the Trace subsystem is one that returns the names of the defined traces in the running application.

Query: TRACe:CATalog?
or
trac:cat?

If there are no named traces, a single empty string is returned. When multiple traces are defined, the names are returned in a comma separated list of trace_name strings. A listing of the possible trace names by application is provided in the *TRACe* subsystem commands in Section 3 of the *AM700 Programmer Manual*.

MMEMory Subsystem

The Mass Memory subsystem behavior in the AM700 is very similar to to the documented SCPI behavior. One major divergence from standard SCPI is the absence of ‘msus’ (mass storage unit specifier) support in the AM700. The AM700 allows an optional mass storage unit specifier with any filename given to the MMEMory commands. The syntax of the file name is:

```
<[device:] {/path_name/path_name/} <file_name>
```

The device portion is optional. “device” can be one of “rom”, “nvram”, or “dos”.

SENSe5	Subframe ‘A’ digital audio
SENSe6	Subframe ‘B’ digital audio
SENSe7	DSP port ‘A’
SENSe8	DSP port ‘B’
SENSe9	Digital reference ‘A’
SENSe10	Digital reference ‘B’

A few SENSe commands related to the digital audio signal are:

Syntax: SENSe[5-8]:DAUDio:AUDio:SRATe?
Returns the sample rate of the incoming signal.

LOW is 32 kHz
MEDIum is 44.1 kHz
HIGH is 48 kHz

SENSe[5-8]:DAUDio:INTerface:BWIDth
LOW|MEDIum|HIGH

SENSe[5-8]:DAUDio:INTerface:JGAin
NORMal|HIGH

CSTReam Subsystem

The sense choices fed to the calculate blocks by a CSTReam:FEED command are fixed by application and measurement. You may query the CSTReam:FEED to determine the source of the feed.

Example: CSTReam1:FEED?
CSTReam2:FEED?’

The SENSE that is feeding the CSTReam is returned.

CALCulate Subsystem

A number of independent subsystems comprise the CALCulate subsystem. Each of the subsystems is a sub-block of the CALCulate block. Data flows through the sub-blocks in serial fashion. The CALCulate block has more than one instance of some of the sub-blocks. The same named sub-blocks are differentiated by a numeric suffix. The syntax of the CALCulate subsystem commands are application dependent. One example of the signal flow in show in Figure 1-9. It illustrates the FFT CALCulate subsystem. CALC blocks for the other applications are fixed by application and measurement.

Suffixes: CALCulate1 Trace1 Zoom
 CALCulate2 Trace2 Zoom
 CALCulate3 Trace1 Average
 CALCulate4 Trace2 Average

FFT Analyzer SCPI Commands. In the FFT Analyzer, the calculate blocks are separated by channels and function. Calculate1 is used for the fft zoom functions of channel 1 and is fed by the CSTReam1 feed. Calculate2 is also used for the fft zoom functions, but for channel 2. It is fed by the CSTReam2 feed where the feeds are one of the SENSE signals. Calculate3 and Calculate4 control the averaging commands of the FFT Analyzer. As with calc1 and calc2 there are two channels, and calc3 controls the averaging on channel 1 while calc controls the averaging on channel 2.

Range :	1	FFT analyzer
	2	Audio analyzer
	3	Monitor
	4	Digital Interface Tester
	5	Diagnostics
	6	PanelCal

SYSTEM Subsystem

SYSTEM commands are used to communicate with operating system of the AM700. Status, ID, errors, and other required interface type operations related to running the AM700 remotely are included in the SYSTEM commands. A useful query of the system commands is:

```
SYST:ERR?
```

This query returns error messages from the error message stack until all messages have been read. If no error exists, it returns "0, NO ERROR."

Also found in the system commands are the ones used for setting the communications parameters for RS-232 communications (baud rate, data bits, and parity) and the GPIB address.

STATUS Subsystem

The STATUS Subsystem allows you to set the AM700 status registers and query the state of operation of the AM700 to determine if an event is happening or has happened. Refer to the *AM700 Programmer Manual* for further information.

TRACe Subsystem

The trace subsystem contains measurement results for display and remote query. Trace queries are used to return the measurement

subsystem are given in Section 2 of this manual, with full definitions given in the *AM700 Programmer Manual*.

Windows 1 - n, specify the view windows in application. There are four graphical display windows plus the real time view in Audio Analyzer and two in the FFT Analyzer. Window60 is the Generator Status display, Window98 is for Notifiers, and Window99 is the user text window.

INSTRUMENT Subsystem

Usage: Commands in this subsystem are used to switch between AM700 applications either by name or number. Several query commands are used to return the names and application numbers. These are the several versions of catalog queries. The catalog queries, INST:CAT, INST:CAT:FULL, INST:LCAT, and INST:LCAT:FULL all return comma separated list of the application name strings.

Syntax: INST:SEL <app_name>

Syntax: INST:NSEL <app_number>

Syntax: INST:LSEL <app_descriptive_name>

Query: INST:NSEL? returns the application number.

Range:

FFT	FFT analyzer
Analyzer	Audio analyzer
Monitor	Monitor
Digital	Digital Interface Tester
Diagnostics	Diagnostics
PanelCal	Touch Panel Calibration

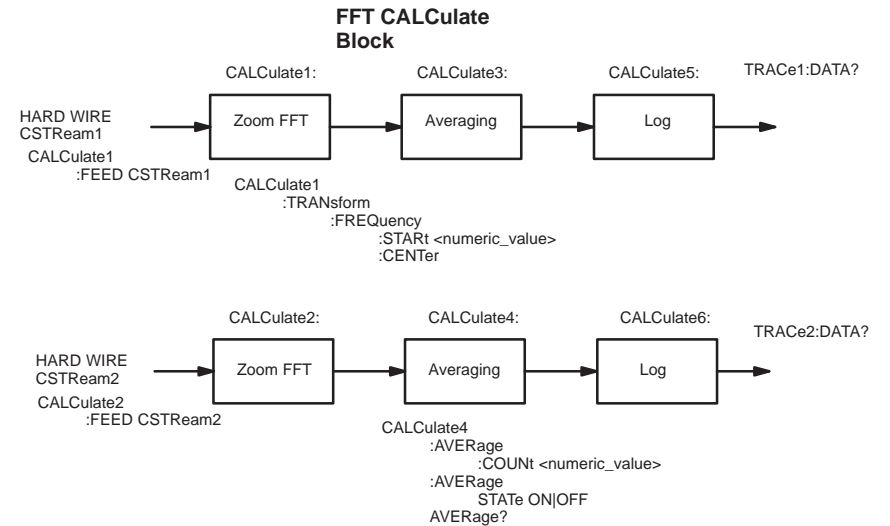


Figure 1-9: FFT CALCulate Subsystem Block

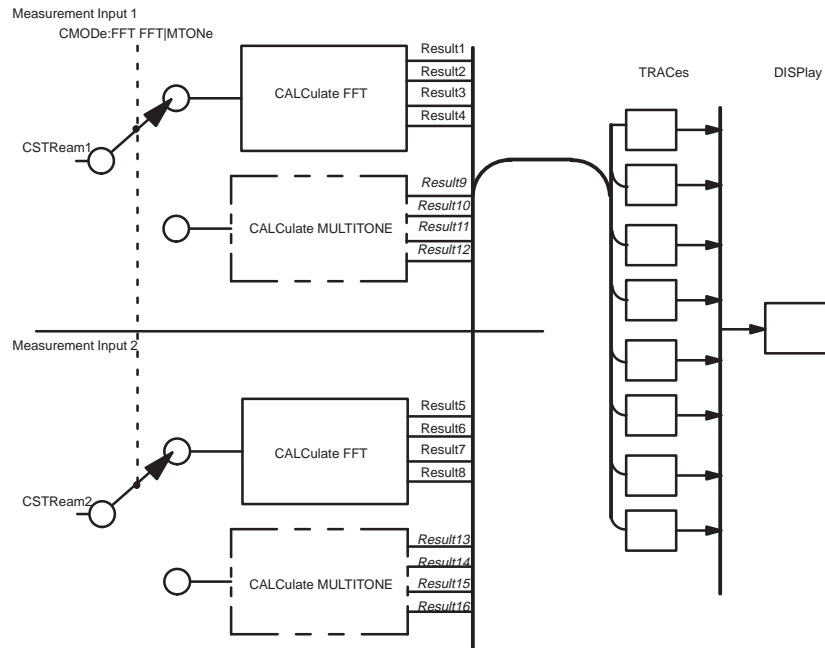


Figure 1-10: Switching Between FFT and Multitone

The FFT Analyzer application also contains the Multitone measurement calculate blocks. A special SCPI command switches the FFT Analyzer between the two functions as shown in Figure 1-10. The calculate block results are given different trace names.

AMEasure Subsystem

The AMEasure subsystem provide commands that quickly set up the AM700 to perform selected measurements and to control some parameter settings of applications. The commands are application specific. The majority of the AMEasure commands are used for the Audio Analyzer application. These commands include those used to control settling time, command regulation mode, and select the measurements to be done by the Audio Analyzer.

Suffixes:

AME1	MEAS 1
AME2	MEAS 2
AME3	MEAS 3
AME4	MEAS 4

FFT and the AMEasure Subsystem. FFT supports 5 separate measurements. They are:

- AMEasure1: FFT on channel 1
- AMEasure2: FFT on channel 2
- AMEasure3: Configurable Multitone
- AMEasure4: Configurable Multitone
- AMEasure5: Configurable Multitone

These measurements ‘assignments’ are described by the the AME:MODE command. FFT implements AME:MODE, with the following initial values for each suffix combination:

- AME1:MODE FFT
- AME2:MODE FFT
- AME3:MODE MTONE
- AME4:MODE MTONE
- AME5:MODE MTONE

DISPlay Subsystem

The display commands control the number and type of displays that are available for a running application. The adjustments available in the DISPlay subsystem include controlling display brightness, turning on and off view windows (graphs, text, and generator control panel), enabling and positioning cursors, entering text in a text view window (dialog box), assigning X- and Y-axis scales, and switching the graphical plotting methods (point-to-point, linear or logarithmic). The complete set of commands for the DISPlay