GP-IB
OPERATION MANUAL
ERROR DETECTOR
MP1702A/MP1756A
MP1609A/MP1651A

APRIL 1993
VER. II

ANRITSU CORPORATION

MAR.
1998

CERTIFICATION

ANRITSU CORPORATION certifies that this instrument has been thoroughly tested and inspected, and found to meet published specifications prior to shipping.
Anritsu further certifies that its calibration measurements are based on the Japanese Electrotechnical Laboratory and Radio Research Laboratory standards.

WARRANTY

All parts of this product are warranted by Anritsu Corporation of Japan against defects in material or workmanship for a period of one year from the date of delivery.
In the event of a defect occurring during the warranty period, Anritsu Corporation will repair or replace this product within a reasonable period of time after notification, free-of-charge, provided that: it is returned to Anritsu; has not been misused; has not been damaged by an act of God; and that the user has followed the instructions in the operation manual.

Any unauthorized modification, repair, or attempt to repair, will render this warranty void.

This warranty is effective only for the original purchaser of this product and is not transferable if it is resold.

ALL OTHER EXPRESSED WARRANTIES ARE DISCLAIMED AND ALL IMPLIED WARRANTIES FOR THIS PRODUCT, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED IN DURATION TO A PERIOD OF ONE YEAR FROM THE DATE OF DELIVERY. IN NO EVENT SHALL ANRITSU CORPORATION BE LIABLE TO THE CUSTOMER FOR ANY DAMAGES, INCLUDING LOST PROFITS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES, ARISING OUT OF THE USE OR INABILITY TO USE THIS PRODUCT.

All requests for repair or replacement under this warranty must be made as soon as possible after the defect has been noticed and must be directed to Anritsu Corporation or its representative in your area.

WARNING

NO OPERATOR SERVICEABLE PARTS INSIDE.
REFER SERVICING TO QUALIFIED PERSONNEL.

CAUTION

FOR CONTINUED FIRE PROTECTION REPLACE
ONLY WITH SPECIFIED TYPE AND RATED FUSE.

TABLE OF CONTENTS

SECTION 1
GENERAL

The MP1702A/MP1756A/MP1609A/MP1651A comes equipped with the GP-IB (General Purpose Interface Bus based on IEEE Std. 488.2) as a standard feature. However, the common commands (headers defined in IEEE Std. 488.2 which have an *) are not supported; equivalent control commands are provided in place of these commands.

For details, refer to the relationship between the common commands and the MP1702A/MP1756A/MP1609A/MP1651A commands shown in SECTION 6 GP-IB CONTROL COMMANDS, paragraph 6.2. Also, the MP1702A/MP1756A/MP1609A/MP1651A uses only the necessary standard event status byte functions.

The programmable patterns of the MP1702A/MP1756A/MP1609A/MP1651A can be set via GP-IB, as can the output interface parameters and output parameters. In addition, information on various set patterns parameters can be saved to or read from a 3.5-inch floppy disk. Furthermore, a DMA function enables high-speed transfer of large amounts of pattern data.

In addition to device functions, the MP1702A/MP1756A/MP1609A/MP1651A provides switch-selectable controller functions. The controller functions that can be implemented with the MP1702A/MP1756A/MP1609A/MP1651A are, however, limited to the pattern tracking functions (see paragraph 2.2) that transfer measurement patterns to ANRITSU's Pulse Pattern Generators.

Note: Take care of the following when reading "SECTION 15 SAMPLE PROGRAMS" in this manual.

There is no description about the MP1756A/MP1755A in the sample programs, but the GP-IB commands for the MP1756A/MP1755A can be used commonly with the MP1702A/MP1701B. However, when using the MP1755A, internal clock frequency setting is ineffective.

SECTION 2
GP-IB ADDRESS SETTING

2.1 Setting Device Addresses

When using the GP-IB, addresses must be set for each device connected to the bus. To set the GP-IB address of the MP1702A/MP1756A/MP1609A/MP1651A, use the GP-IB ADDRESS switch located on the rear panel.

Addresses 0 to 30 can be set. The five switches are individually weighted so that [5] has a weight of 16, [4] has a weight of 8, [3] has a weight of 4, [2] has a weight of 2, and [1] has a weight of 1. To set the address to 11, for example,

$$11 = 8 + 2 + 1$$

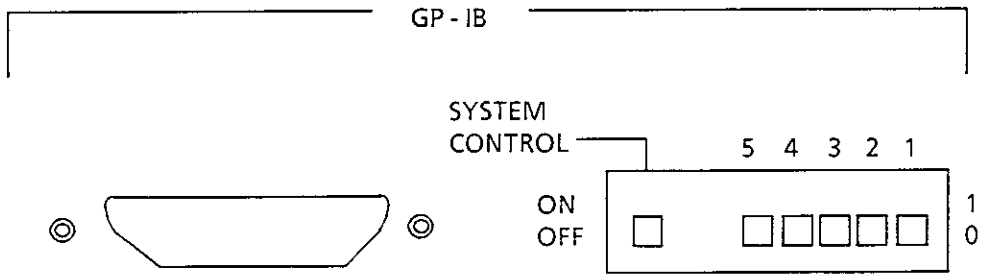
so, set the switches as follows

[5] = 0, [4] = 1, [3] = 0, [2] = 1, and [1] = 1

However, if all switches are set to ON (i.e., 31), the address is assumed to be 0.

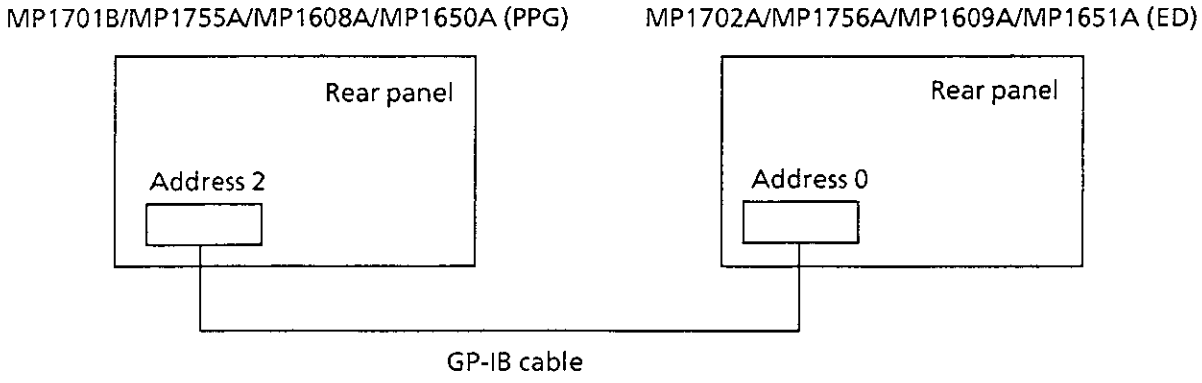
Notes: 1. The MP1702A/MP1756A/MP1609A/MP1651A continuously checks the GP-IB ADDRESS settings as long as the power is on. Therefore, changes in addresses are always accepted.

2. When controlling the MP1702A/MP1756A/MP1609A/MP1651A externally, set the SYSTEM CONTROL on the GP-IB ADDRESS switch to OFF (0).



2.2 Pattern Tracking (System Controller) with MP1701B/MP1755A/MP1608A/ MP1650A (PPG)

Step	Procedure
1	Connect the Pulse Pattern Generator (PPG) as shown below.
2	Set SYSTEM CONTROL switch on the MP1702A/MP1756A/MP1609A/MP1651A's GP-IB ADDRESS to ON (1).
3	Set the address of the MP1701B/MP1755A/MP1608A/MP1650A (PPG) to the MP1702A/MP1756A/MP1609A/MP1651A address plus 2. (If the MP1702A/MP1756A/MP1609A/MP1651A address is 0, for example, set the PPG address to "2".)
4	Turn on the power of the MP1702A/MP1756A/MP1609A/MP1651A to enable pattern tracking.



SECTION 3
GP-IB CABLE CONNECTION

Connect the GP-IB cable to the GP-IB connector "GP-IB" located on the rear panel. When connecting the GP-IB cable, fasten the fixing screws to the cable connector, then attach the cable connector to the GP-IB connector of the MP1702A/MP1756A/MP1609A/MP1651A.

Although a GP-IB system can accommodate up to 15 devices, cable lengths are limited as follows:

(Total cable length) $\leqq$ 2 m $\times$ (number of devices)

However, the maximum total length of all cables is 20 m.

The next page shows the GP-IB connection diagram of the MP1702A/MP1756A/MP1609A/MP1651A.

- Pin connections

Pin No.	Signal name	Pin No.	Signal name
1	DIO1	13	DIO5
2	DIO2	14	DIO6
3	DIO3	15	DIO7
4	DIO4	16	DIO8
5	EOI	17	REN
6	DAV	18	Signal ground
7	NRFD	19	Signal ground
8	NDAC	20	Signal ground
9	IFC	21	Signal ground
10	SRQ	22	Signal ground
11	ATN	23	Signal ground
12	Shield	24	Signal ground

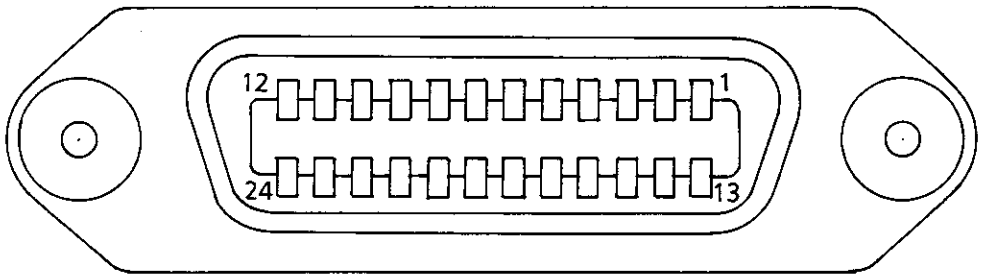


Fig. 3-1 GP-IB Connector

SECTION 4
INTERFACE FUNCTION

The GP-IB interface functions of the MP1702A/MP1756A/MP1609A/MP1651A are listed in Table 4-1.

Table 4-1 GP-IB Interface Functions

Function	Symbol	Explanation
Source handshake	SH1	All source handshake functions provided
Acceptor handshake	AH1	All acceptor handshake functions provided
Talker	T6	Basic talker functions provided Serial polling function provided Talk only function not provided Clear designated talker by listener function provided
Extended talker	TE0	Extended talker function not provided
Listener	L4	Basic listener functions provided Listen only mode function not provided Clear designated listener by talker function provided
Extender listener	LE0	Extended listener function not provided
Service request	SR1	All service request functions provided
Remote/local	RL1	All remote/local functions provided
Parallel polling	PP0	Parallel polling function not provided
Device clear	DC1	All device clear functions provided
Device trigger	DT1	Device trigger functions provided
Controller	C1 C2 C3 C27	System controller functions provided IFC send function provided REN send function provided Interface message function provided
Bus electrical condition	E2	Tri-state driver

SECTION 5
GP-IB COMMAND FORMAT

5.1 COMMAND FORMAT

The MP1702A/MP1756A/MP1609A/MP1651A GP-IB command format consists of a header field that indicates the command type, and a numeric data field that indicates spaces and values.

Use at least one space after the header field to separate it from the numeric data field.

In the data request message format, the header field, which shows the command type, is followed by [?].

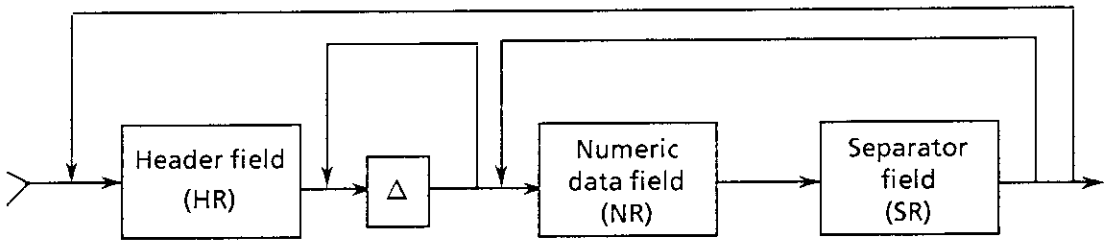
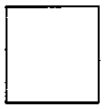


Fig. 5-1 Basic Command Format

5.2 Explanation of Symbols and Description

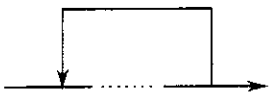
The method using the figure shown in Fig. 5-1 and the format figures shown in paragraph 5.3 are described below:

: Indicates data element.

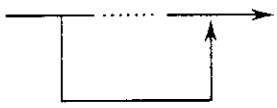


: Indicates the data element of one character.

: Indicates connection of data elements.



: Indicates that data elements in this area can be repeated.



: Indicates that this area can be omitted.

5.3 Header Field (HR)

HR2 headers consist of three alphabetic characters (A to Z, a to z).

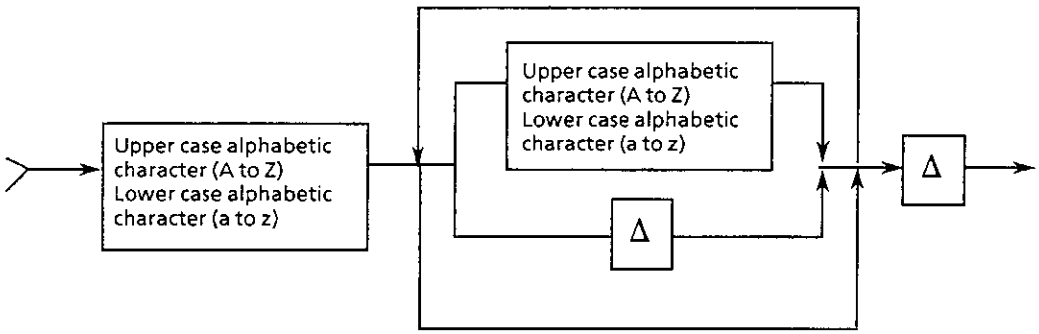


Fig. 5-2 HR2 Format

5.4 Numeric Data Field (NR)

The numeric data field has four formats depending on the comamnd: NR1 format (integer type), NR2 format (real number type), Hexadecimal format (HEX type), Binary format (BIN type).

5.4.1 NR1 format (integer type)

NR1 format is used to represent integers. It consists of a character sign (+, −) followed by a string of integers (0 to 9). The + sign can be omitted, or replaced by a space. Leading spaces can also be inserted to align data, etc.

Examples:

- General representation +1234, − 567, +0
- + sign omitted +1234 → 1234
- + sign replaced +1234 → Δ1234
- Spaces inserted +1234 → ΔΔ+1234
  −567 → ΔΔΔ−567

(Δ : Space)

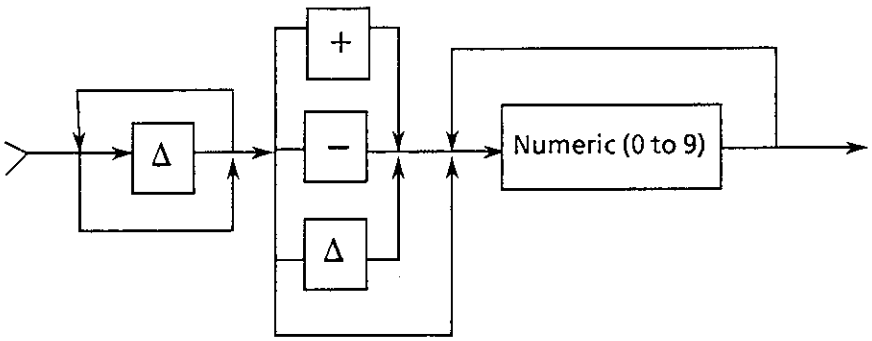


Fig. 5-3 NR1 Format

5.4.2 NR2 format (real number type)

NR2 format is used to represent real numbers.

It consists of a character sign (+, −) followed by a string of integers (0 to 9), a decimal point (.), and another string of integers (0 to 9).

The + sign can be omitted or replaced and leading spaces can be inserted to align data, as for the NR1 format. When the decimal point is preceded, or followed by 0, the 0 can be omitted.

Examples:

- General representation +1.23, −45.6, −0.12, +34.0, +0.0
- + sign omitted +1.23 → 1.23
- + sign replaced +1.23 → Δ1.23
- Space insertion +1.23 → ΔΔ+1.23
  −45.6 → Δ−45.6
- High-order 0 omitted −0.12 → −.12
- Low-order 0 omitted +34.0 → +34
- +0.0 omitted +0.0 → +0. or +.0

(Δ : Space)

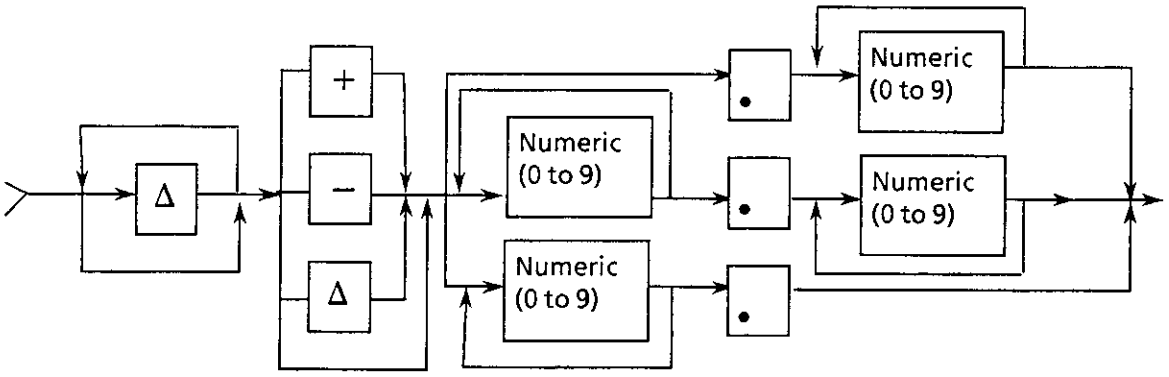


Fig. 5-4 NR2 Format

5.4.3 Hexadecimal format (HEX type)

Hexadecimal format consists of the identifier #H followed a string of integers (0 to 9) and alphabetic characters (A to F). In hexadecimal format, a space can be inserted between the identifier and data. High-order zeros can be omitted.

Examples:

- General representation #H1234,#H00FF,#H0000
- Spaces inserted #H1234 → #HΔ1234
  #H00AF → #HΔΔΔ00AF
- Zeros omitted #H00FF → #HFF
  #H0000 → #H0

(Δ : Space)

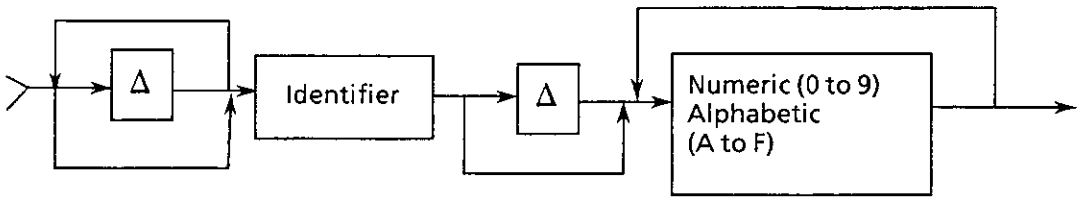


Fig. 5-5 Hexadecimal Format

5.4.4 Binary format (BIN type)

Binary format consists of the identifier #B followed by a string of binary digits (0 and 1). In binary format, a space can be inserted between the identifier and data.

When the most significant bit is 0, it can be omitted.

Examples:

- General representation #B11011011,#B00100100,
  #B00000000
- Space insertion #B11011011 → #BΔ11011011
  #B00100100 → #BΔΔΔ00100100
- 0 omission #B00100100 → #B100100
  #B00000000 → #B0

(Δ : Space)

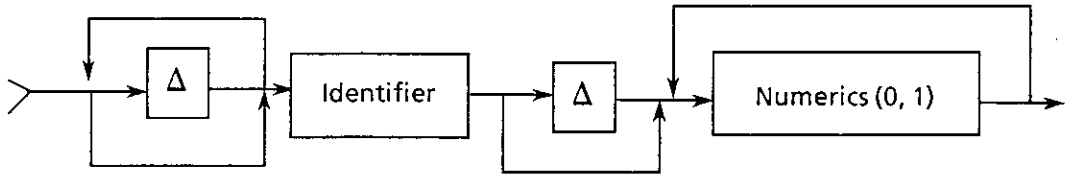


Fig. 5-6 Binary Format

5.5 Command Separator

There are four command separators:

SR1 format (semicolon ;) SR1 format (comma ,) SR2 format (line feed LF), SR3 format (END message).

5.5.1 SR1 format (semicolon ;)

The SR1 format (semicolon ;) is used to separate consecutive commands.

Example:

ABC123;DEF45.6 is decoded as two commands, "ABC123" and "DEF45.6".

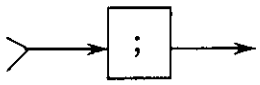


Fig. 5-7 SR1 (Semicolon ;) Format

5.5.2 SR1 format (comma ,)

The SR1 format (comma ,) is used to separate data sent after a header filed.

Example:

ABC123,45.6 is decoded as two data, "123" and "45.6" following the header "ABC".

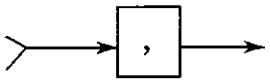


Fig. 5-8 SR1 (Comma ,) Format

5.5.3 SR2 format (line feed LF)

The SR2 format is used to indicate the end of a command sequence . The line feed (LF) code can be used as the SR2 format code. The carriage return (CR) code can also be inserted before the line feed code.

When the MP1702A/MP1756A/MP1609A/MP1651A is designated to be a listener, it continues to function as a listener until the SR2 or SR3 format code, described next, is received. When the SR2 format or SR3 format code is received, reception terminates and the received command is decoded.

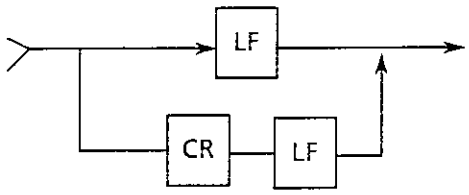


Fig. 5-9 SR2 Format

5.5.4 SR3 format (END message)

The SR3 format is used to indicate the end of a command sequence, as does the SR2 format.

An END message (EOI line used) can be used as the SR3 format code. (The SR3 and SR2 formats can be used together.)

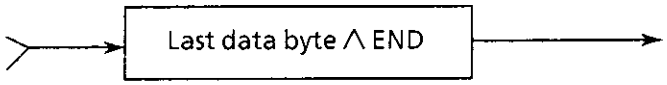


Fig. 5-10 SR3 Format

5.6 Definition of Space

The MP1702A/MP1756A/MP1609A/MP1651A treats the ASCII codes 00H to 09H and 0BH to 20H as blank spaces.

SECTION 6
GP-IB MESSAGES

6.1 GP-IB Message List

Table 6-1 lists the control messages and data request messages. The messages that can be used depend upon the operation mode. Also, some messages are restricted by conditions that have already been set.

Table 6-1 Outline of GP-IB Commands (1/5)

Function	Control message			Data request message	No.
	Header field	Space	Numeric data field	Header field	
Page number	ADR		NR1 format	ADR?	⑯
Preset (all bits on all pages)	ALL		NR1 format	———	㉑
Alarms monitor	ALM		NR1 format	ALM?	㊹
Measurement result buffer clear	BCL		———	———	㉜
Pattern bit	BIT		NR1 format Hexadecimal format	BIT?	⑱
Intermediate data calculation	CAL		NR1 format	CAL?	㊸
Clock loss state	———		———	CLI?	㉓
Clock loss processing	CLS		NR1 format	CLS?	㊾
Clock input phase (delay)	CPA		NR1 format	CPA?	③
Clock input polarity	CPL		NR1 format	CPL?	⑥
Clock input termination voltage	CTM		NR1 format	CTM?	④
Intermediate result display	CUR		NR1 format	CUR?	㉗
Display light selection	DGT		NR1 format	DGT?	㊷

Table 6-1 Outline of GP-IB Commands (2/5)

Function	Control message			Data request message	No.
	Header field	Space	Numeric data field	Header field	
Measurement data length	DLN		NR1 format	DLN?	⑮
Delay status	———		———	DLY?	⑤
Display mode	DMS		NR1 format	DMS?	㉖
One-second data print threshold selection	DOT		NR1 format	DOT?	㊶
Data input threshold voltage	DTH		NR2 format	DTH?	①
Data input termination voltage	DTM		NR1 format	DTM?	②
Error count measurement results	———		———	EC?	㉞
Extended event status register	———		———	EER?	㊾
Extended event status enable register	EES		NR1 format	EES?	㊽
Error-free internal measurement results	———		———	EFI?	㊱
Error interval measurement results	———		———	EI?	㉟
Error performance data print selection	EPF		NR1 format	EPF?	㉘
Error ratio measurement results	———		———	ER?	㉝
Error detection status	———		———	ERS?	㉕
Standard event status enable register	ESE		NR1 format	ESE?	㊻
Standard event status register	———		———	ESR?	㊼
Error performance threshold	ETH		NR1 format	ETH?	㉛

Table 6-1 Outline of GP-IB Commands (3/5)

Function	Control message			Data request message	No.
	Header field	Space	Numeric data field	Header field	
Error detection mode selection	ETY		NR1 format	ETY?	⑭
File No./directory mode switching	FIL		NR1 format	FIL?	㊲
Measurement frame length	FLN		NR1 format	FLN?	⑰
Data print format selection	FMT		NR1 format	FMT?	㊻
Frame/word length selection	FRM		NR1 format	FRM?	⑫
One-second clock frequency measurement result	———		———	FRQ?	㊼
Frame patterns synchronization	FSY		NR1 format	FSY?	⑪
Intermediate measurement result output	———		———	IMD?	㊶
Initialization	INI		———	———	⑳
Intermediate data print selection	ITM		NR1 format	ITM?	㊾
Measurement interval time selection	ITV		NR1 format	ITV?	㊺
Measurement pattern logic	LGC		NR1 format	LGC?	⑧
Floppy disk access status	———		———	MAC?	㊶
Memory function switching	MEM		NR1 format	MEM?	㊱
Measurement mode	MOD		NR1 format	MOD?	㉘
Error detection	MON		NR1 format	MON?	㊺
Measurement pattern mark ratio	MRK		NR1 format	MRK?	⑩
Mask route length selection	MSK		NR1 format	MSK?	㊻

Table 6-1 Outline of GP-IB Commands (4/5)

Function	Control message			Data request message	No.
	Header field	Space	Numeric data field	Header field	
Measurement status	———		———	MSR?	㉛
Measurement result GP-IB output select status	———		———	MTY?	㊸
One-second data print selection	OSC		NR1 format	OSC?	㊵
Page number	PAG		NR1 format	PAG?	⑯
Measurement time setting	PRD		NR1 format	PRD?	㉟
Print function	PRN		NR1 format	PRN?	㊷
Manual printing	PSA		———	———	㊸
Preset (all bits on 1 page)	PST		NR1 format	———	㉒
Paper saving function	PSV		NR1 format	PSV?	㊷
Measurement pattern	PTN		NR1 format	PTN?	⑨
Data recall	RCL		NR1 format	———	㊳
Number of pattern data output bytes	RED		NR1 format	———	⑳
Data resave	RSV		NR1 format	———	㊵
Internal timer setting	RTM		NR1 format	RTM?	㉞
Data save	SAV		NR1 format	———	㊴
Number of mask route channels	SCH		Binary format	SCH?	㊼
Number of mark ratio AND bit shifts	SFT		NR1 format	SFT?	㊽
Sync loss state	———		———	SLI?	㉔

Table 6-1 Outline of GP-IB Commands (5/5)

Function	Control message			Data request message	No.
	Header field	Space	Numeric data field	Header field	
Sync loss processing	SLS		NR1 format	SLS?	⑤⓪
Automatic phase threshold search	SRH		NR1 format	SRH?	⑦
Service request enable register	SRQ		NR1 format	SRQ?	⑥④
Start measurement	STA		———	———	②⑨
Status byte register			———	STB?	⑥⑤
Stop measurement	STO		———	———	③⓪
Automatic synchronization	SYN		NR1 format	SYN?	③②
Threshold EI/EFI data print selection	THR		NR1 format	THR?	⑤⑦
Real-time/measurement-time display switching	TIM		NR1 format	TIM?	③③
Word length	WLN		NR1 format	WLN?	⑭
Number of words	WNB		NR1 format	WNB?	⑬
Number of pattern data input bytes	WRT		NR1 format	———	⑲

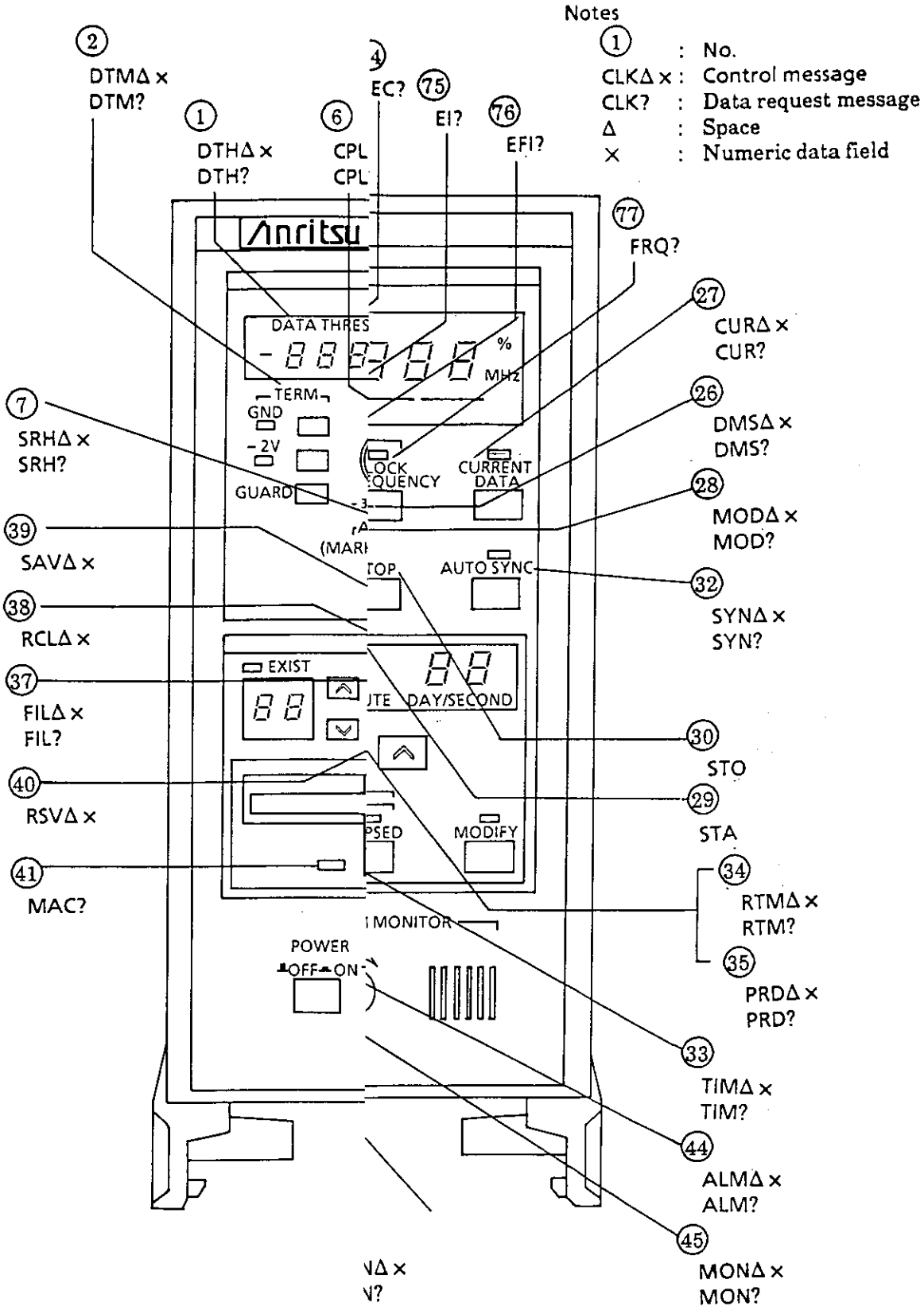


Fig. 6-1 Control Message and Data Request Message (Front Panel)

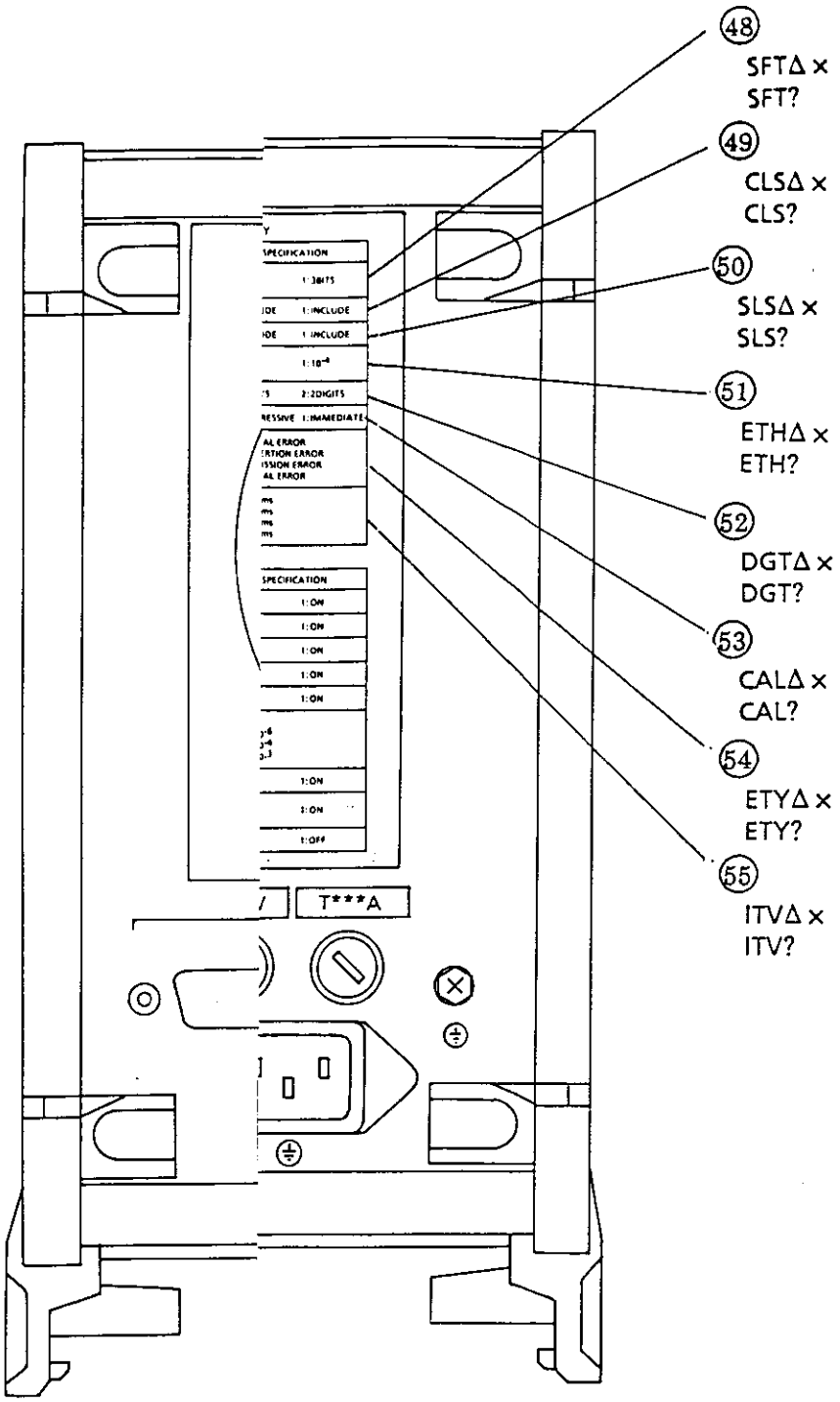


Fig. 6-2 Control Message and Data Request Message (Rear Panel)

6.2 Correspondence with Common Commands

The relationship between the common commands defined by IEEE 488.2 and the MP1702A/ MP1756A/MP1609A/MP1651A commands is shown in Table 6-2.

The common commands in the table below are defined as essential commands by IEEE 488.2. For a detailed description of the common commands, refer to IEEE 488.2.

Table 6-2 Correspondence with Common Commands

Common Command	Corresponding Command	No.	Item on IEEE 488.2 standard
*IDN?	———	———	10.14
*RST	———	———	10.32
*TST?	———	———	10.38
*OPC	———	———	10.18
*OPC?	———	———	10.19
*WAI	———	———	10.39
*CLS	———	———	10.3
*ESE	ESE	(66)	10.10
*ESE?	ESE?	(66)	10.11
*ESR?	ESR?	(67)	10.12
*PSC	———	———	10.25
*PSC?	———	———	10.26
*SRE	SRQ	(64)	10.34
*SRE?	SRQ?	(64)	10.35
*STB?	STB?	(65)	10.36

SECTION 7
DEVICE CLEAR

When the MP1702A/MP1756A/MP1609A and MP1651A receive the device clear command, they are re-initialized to the power-on state.

If the device clear command is received during a measurement, the measurement is aborted and all measurement data are lost.

At the same time, the rear-panel functions are also re-initialized according to the DIP switch settings. (As the mode is switched from REMOTE to LOCAL, the rear-panel functions are also re-initialized according to the DIP switch settings.)

Furthermore, if the device clear command is received while the floppy disk is being accessed then that, access is aborted.

SECTION 8
DEVICE TRIGGER

When the device trigger command is received, the MP1702A/MP1756A/MP1609A/MP1651A operate as though they had received an STA command.

SECTION 9
MESSAGE BUFFER LENGTH

The maximum message length for data accepted by the MP1702A/MP1756A/MP1609A/MP1651A is 256 bytes (256 characters). Messages longer than 256 bytes are not input to the MP1702A/MP1609A/ MP1651A. Therefore, the maximum length of a single GP-IB command line is 256 characters including CR/LF codes.

DTMΔ0 ; PTNΔ0 ;·····; MEMΔ0 (CR/LF)

256 characters or less

Also note that the maximum length of messages output by the MP1702A/MP1756A/MP1609A/ MP1651A is 512 bytes.

SECTION 10
CONTROL MESSAGE AND DATA REQUEST MESSAGE

The following examples use HP-BASIC language of the Hewlett-Packard HP9000 series computer.

10.1 INPUT Section

The following pages describe the GP-IB messages of the INPUT section. Δ indicates a space.

① Data input threshold voltage (Data threshold)

Set value	Control message	Data request message	Output message
−3.000 V ⋮ +1.750 V	DTHΔ−3.000 ⋮ DTHΔ+1.750	DTH?	DTH Δ−3.000 ⋮ DTHΔΔ1.750
The resolution is 0.002 V.			

Restrictions

Control message — In the following cases, this message is invalid and a buzzer rings:

- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message — None

Examples

Control message

OUTPUT 700; "DTHΔ−3.0"

The data input threshold voltage becomes −3.0 V.

Data request mesage

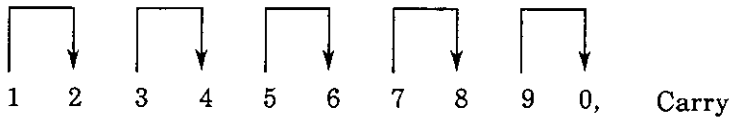
When the data input threshold voltage is −3.0 V

```
OUTPUT 700; "DTH?"
ENTER 700; B$
PRINT B$
```

↓

DTHΔ−3.000 (CR/LF) is output.

Note: If the third decimal place of the number representing the threshold voltage is not a multiple of 0.002 V, then the number is rounded up to the nearest effective value as shown below.



② Data input termination voltage (Data termination)

Set value	Control message	Data request message	Output message
GND	DTMΔ0	DTM?	DTMΔ0
−2 V	DTMΔ1		DTMΔ1

Restrictions

Control message — In the following cases, this message is invalid and a buzzer rings:

- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message — None

Examples

Control message

OUTPUT 700; "DTMΔ0"

Sets the data input termination voltage to GND.

Data request message

When the data input termination voltage is GND

```
OUTPUT 700; "DTM?"
ENTER 700;B$
PRINT B$
```

↓

DTMΔ0 (CR/LF) is output.

③ Clock input phase (Clock phase adjust)

MP1702A/MP1609A				
Set value	Control message	Data request message	Output message	Setting Resolution
+500 ps ⋮ −500 ps	CPAΔ500 ⋮ CPAΔ−500	CPA?	CPAΔΔΔ 500 ⋮ CPAΔΔ−500	1 ps

MP1651A				
Set value	Control message	Data request message	Output message	Setting Resolution
+1000 ps ⋮ −1000 ps	CPAΔ1000 ⋮ CPAΔ−1000	CPA?	CPAΔΔ 1000 ⋮ CPAΔ−1000	2 ps

Restrictions

Control message — In the following cases, this message is invalid and a buzzer rings:

- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message — None

Examples

Control message

OUTPUT 700; "CPAΔ500"

Sets the clock input phase to 500 ps.

Data request message

When the clock input phase is 500 ps

```
OUTPUT 700; "CPA?"
ENTER 700;B$
PRINT B$
```

↓

CPAΔΔΔ500 (CR/LF) is output.

④ Clock input termination voltage (Clock termination)

Set value	Control message	Data request message	Output message
GND	CTMΔ0	CTM?	CTMΔ0
−2 V	CTMΔ1		CTMΔ1

Restrictions

Control message — In the following cases, this message is invalid and a buzzer rings:

- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message — None

Examples

Control message

```
OUTPUT 700; "CTMΔ0"
```

Sets the clock input termination voltage to GND.

Data request message — When the clock input termination voltage is GND

```
OUTPUT 700; "CTM?"
ENTER 700;B$
PRINT B$
```

↓

CTMΔ0 (CR/LF) is output.

⑤ Delay status (Delay unlock?)

Set value	Control message	Data request message	Output message
READY	———	DLY?	DLYΔ0
BUSY			DLYΔ1

Restrictions

Data request message None

Examples

Data request message When the clock-input-phase servo circuit is in a READY state

```
OUTPUT 700; "DLY?"
ENTER 700;B$
PRINT B$
```

↓

DLYΔ0 (CR/LF) is output.

Note: READY means that the clock-input-phase servo circuit is not in operation. BUSY means that the clock-input-phase servo circuit is in operation.

⑥ Clock input polarity (Clock polarity)

Set value	Control message	Data request message	Output message
CLK	CPLΔ0	CPL?	CPLΔ0
$\overline{\text{CLK}}$	CPLΔ1		CPLΔ1

Restrictions

Control message — In the following cases, this message is invalid and a buzzer rings:

- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message — None

Examples

Control message

```
OUTPUT 700; "CPLΔ0"
```

Sets the clock input polarity to CLK.

Data request message — When the clock input polarity is CLK

```
OUTPUT 700; "CPL?"
ENTER 700;B$
PRINT B$
        ↓
```

CPLΔ0 (CR/LF) is output.

⑦ Automatic phase threshold search (Augo search)

Set value	Control message	Data request message	Output message
Auto search OFF	SRHΔ0	SRH?	SRHΔ0
Auto search ON	SRHΔ1		SRHΔ1
When search cannot be performed.	———		SRHΔ2

Restrictions

Control message

In the following cases, this message is invalid and a buzzer rings:

- When the AUTO SYNC function is OFF
- When the measurement pattern mark ratio is 0/8 or 8/8
- When the mark ratio of the pattern data is less than 1/8 or greater than 7/8 while the measurement pattern PRGM mode
- When the clock is lost
- When floppy disk is being accessed

Data request message

None

Examples

Control message

```
OUTPUT 700; "SRHΔ0"
```

Switches the AUTO SEARCH function to OFF.

Data request message

When the AUTO SEARCH function is OFF

```
OUTPUT 700; "SRH?"
ENTER 700;B$
PRINT B$
  ↓
```

SRHΔ0 (CR/LF) is output.

10.2 PATTERN Section

The following pages describe the GP-IB messages in the PATTERN section.

Δ indicates a space.

⑧ Output pattern logic (Logic mode)

Set value	Control message	Data request message	Output message
Positive logic (POSITIVE)	LGCΔ0	LGC?	LGCΔ0
Negative logic (NEGATIVE)	LGCΔ1		LGCΔ1

Restrictions

Control message — In the following cases, this message is invalid and the buzzer rings:

- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message — None

Examples

Control message

```
OUTPUT 700; "LGCΔ0"
```

Sets positive Output pattern logic (POSITIVE).

Data request message — When output pattern logic is positive logic (POSITIVE)

```
OUTPUT 700; "LGC?"
ENTER 700;B$
PRINT B$
```

↓

LGCΔ0 (CR/LF) is output.

Note: When the output pattern is a PRBS pattern and the output pattern logic is changed, the output pattern mark ratio is automatically adjusted.

Positive logic	0/8	1/8	1/4	1/2
Negative logic	8/8	7/8	3/4	$\overline{1/2}$

⑨ Output pattern (Pattern mode)

Set value	Control message	Data request message	Output message
PRGM. WORD	PTNΔ0	PTN?	PTNΔ0
PRGM. DATA	PTNΔ1		PTNΔ1
PRBS 2^7-1	PTNΔ2		PTNΔ2
PRBS 2^9-1	PTNΔ3		PTNΔ3
PRBS 2^{11}-1	PTNΔ5		PTNΔ5
PRBS 2^{15}-1	PTNΔ6		PTNΔ6
PRBS 2^{20}-1	PTNΔ7		PTNΔ7
PRBS 2^{23}-1	PTNΔ8		PTNΔ8
PRBS 2^{31}-1	PTNΔ9		PTNΔ9

Restrictions

Control message

In the following cases, the message is invalid and the buzzer rings:

- When PTNΔ4 is set
- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message

None

Examples

Control message

```
OUTPUT 700; "PTNΔ0"
```

Sets the PRGM.WORD word pattern.

Data request message

When output pattern is PRGM.WORD

```
OUTPUT 700; "PTN?"
ENTER 700;B$
PRINT B$
```

↓

PTNΔ0 (CR/LF) is output.

Note: PRGM: Programmable pattern, PRBS: Pseudorandom pattern

When the output pattern is a PRBS pattern, the previous output pattern mark ratio and settings are used.

⑩ Output pattern mark ratio (Mark ratio mode)

Set value	Control message	Data request message	Output message
Positive logic : 0/8 Negative logic : 8/8	MRKΔ0	MRK?	MRKΔ0
Positive logic : 1/8 Negative logic : 7/8	MRKΔ1		MRKΔ1
Positive logic : 1/4 Negative logic : 3/4	MRKΔ2		MRKΔ2
Positive logic : 1/2 Negative logic : 1/2	MRKΔ3		MRKΔ3

Restrictions

Control message

In the following cases, this message is invalid and the buzzer rings:

- When output pattern is in PRGM. (WORD or DATA) mode
- When AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message

In the following case, this request message is invalid and ERR is output:

- when the output pattern is PRGM. (WORD or DATA) mode

Examples

Control message

When output pattern logic is positive (POSITIVE) and the output pattern is a PRGM pattern.

```
OUTPUT 700; "MRKΔ0"
```

Sets the output pattern mark ratio to 0/8.

Data request message

When output pattern mark ratio is 0/8

```
OUTPUT 700; "MRK?"
ENTER 700;B$
PRINT B$
```

↓

MRKΔ0 (CR/LF) is output.

When output pattern is in PRGM (WORD or DATA) mode

```
OUTPUT 700; "MRK?"
ENTER 700;B$
PRINT B$
```

↓

ERR (CR/LF) is output.

Note: When the output pattern is a PRBS pattern, the previous output pattern mark ratio and settings are used.

⑪ Frame pattern sync (Frame sync)

Set value	Control message	Data request message	Output message
OFF	FSYΔ0	FSY?	FSYΔ0
ON	FSYΔ1		FSYΔ1

Restrictions

Control message

In the following cases, this message is invalid and a buzzer rings:

- When measurement pattern is the PRBS mode
- When the measurement data length is less than 1024
- When the measurement bit length (number of words × word length) is less than 1024
- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message

In the following cases, the message is invalid and an error (ERR) is output.

- When the measurement pattern is a PRBS pattern

Examples

Control message

OUTPUT 700; "FSYΔ0"

Switches the frame pattern sync function OFF

Data request message

When the frame pattern sync function is OFF

```
OUTPUT 700; "FSY?"
ENTER 700;B$
PRINT B$
        ↓
FSYΔ0 (CR/LF) is output.
```

When the measurement pattern is a PRBS pattern

```
OUTPUT 700; "FSY?"
ENTER 700;B$
PRINT B$
        ↓
ERR (CR/LF) is output.
```

⑫ Frame/word length selection (Frame/word length)

Set value	Control message	Data request message	Output message
WORD (normal mode)	FRMΔ0	FRM?	FRMΔ0
FRAME (frame length mode)	FRMΔ1		FRMΔ1

Restrictions

Control message

In the following cases, this message is invalid and a buzzer rings:

- When the frame pattern sync function is OFF
- When the measurement pattern is a PRBS pattern
- When the measurement data length is less than 1024
- When the measurement bit length (number of words × word length) is less than 1024
- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message

In the following cases, the message is invalid and error (ERR) is output

- When the measurement pattern is a PRBS pattern

Examples

Control message

OUTPUT 700; "FRMΔ0"

Switches the frame/word length selection to WORD (normal mode)

Data request message

When the frame/word length selection is WORD (normal mode)

```
OUTPUT 700; "FRM?"
ENTER 700;B$
PRINT B$
```

↓

FRMΔ0 (CR/LF) is output.

When the measurement pattern is a PRBS pattern

```
OUTPUT 700; "FRM?"
ENTER 700;B$
PRINT B$
```

↓

ERRΔ0 (CR/LF) is output.

⑬ Number of words (Number of words)

Set value	Control message	Data request message	Output message
1 ⋮ 32768	WNBΔ 1 ⋮ WNBΔ 32768	WNB?	WNBΔΔΔΔΔ1 ⋮ WNBΔ32768
However, there are some restrictions on the step value.			

Restrictions

Control message

In the following cases, this message is invalid and a buzzer rings:

- When the output pattern is not a WORD pattern
- When FRAME of FRAME/WORD LENGTH is selected
- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message

In the following cases, this request message is invalid and ERR is output:

- When the output pattern is not aWORD pattern
- When FRAME of FRAME/WORD LENGTH is selected

Examples

Control message

When the output pattern is a WORD pattern

OUTPUT 700; "WNBΔ1"

Sets the number of words to 1.

Data request message

When the number of words is 1

OUTPUT 700; "WNB?"
ENTER 700;B$
PRINT B$
↓
WNBΔΔΔΔΔ1 (CR/LF) is output.

When the output pattern is not a WORD pattern

OUTPUT 700; "WNB?"
ENTER 700;B$
PRINT B$
↓
ERR (CR/LF) is output.

Note: The step value for the number of words depens on the word length.
(See Table 10-1.)

Note (Cont.): When the desired number of words cannot be set, an optimal number derived by comparing the displayed value to the input value is set.

Example: For a word length of 2

When 2100 is input as the number of words, t is changed to 2048 and displayed.

If 2100 is input again, the number of words changes to 2112.

Table 10-1 Relationship between Word Length and Number of Words

Word length	Number of words			
	Step width	Range	Step width	Range
2	1 step	1 to 2048	64 step	2112 to 32768
3	1 step	1 to 1365	128 step	1408 to 32768
4	1 step	1 to 1024	32 step	1056 to 32768
5	1 step	1 to 819	128 step	896 to 32768
6	1 step	1 to 682	64 step	704 to 32768
7	1 step	1 to 585	128 step	640 to 32768
8	1 step	1 to 512	16 step	528 to 32768
9	1 step	1 to 455	128 step	512 to 32768
10	1 step	1 to 409	64 step	448 to 32768
11	1 step	1 to 372	128 step	384 to 32768
12	1 step	1 to 341	32 step	352 to 32768
13	1 step	1 to 315	128 step	384 to 32768
14	1 step	1 to 292	64 step	320 to 32768
15	1 step	1 to 273	128 step	384 to 32768
16	1 step	1 to 256	8 step	264 to 32768

⑭ Word length (Word length)

Set value	Control message	Data request message	Output message
2 ⋮ 16	WLNΔ 2 ⋮ WLNΔ16	WLN?	WLN ΔΔ 2 ⋮ WLN Δ16
The set value is every step in the table above.			

Restrictions

Control message

In the following cases, this message is invalid and a buzzer rings:

- When the output pattern is not a WORD pattern
- When FRAME of FRAME/WORD LENGTH is not selected
- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message

In the following cases, this request message is invalid and ERR is output:

- When the output pattern is not a WORD pattern
- When FRAME of FRAME/WORD LENGTH is selected

Examples

Control message

When the output pattern is WORD pattern

OUTPUT 700; "WLNΔ2"

Sets the word length to 2.

Data request message

When the word length is 2

```
OUTPUT 700; "WLN?"
ENTER 700;B$
PRINT B$
```

↓

WLNΔΔ2 (CR/LF) is output.

When the output pattern is not a WORD pattern

```
OUTPUT 700; "WLN?"
ENTER 700;B$
PRINT B$
```

↓

ERR (CR/LF) is utput.

Note: If the number of words is not compatible with the word length setting, it is automatically optimized. (See Table 10-1 on the preceding page.)

⑮ Data length (Data length)

Set value	Control message	Data request message	Output message
2 ⋮ 524288	DLNΔ2 ⋮ DLNΔ524288	DLN?	DLNΔΔΔΔΔΔ2 ⋮ DLNΔ524288
However, there are some restrictions on the step value in the table above.			

Restrictions

Control message

In the following cases, this message is invalid and a buzzer rings:

- When the output pattern is not a DATA pattern
- When FRAME of FRAME/WORD LENGTH is selected
- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message

In the following cases, this request message is invalid and ERR is output:

- When the output pattern is not a DATA pattern
- When FRAME of FRAME/WORD LENGTH is selected

Examples

Control message

When the output pattern is a DATA pattern

```
OUTPUT 700; "DLNΔ2"
```

Sets the data length to 2.

Data request message

When data length is 2

```
OUTPUT 700; "DLN?"
ENTER 700;B$
PRINT B$
```

↓

DLNΔΔΔΔΔΔ2 (CR/LF) is output.

When the output pattern is not a DATA pattern

```
OUTPUT 700; "DLN?"
ENTER 700;B$
PRINT B$
```

↓

ERR (CR/LF) is output.

Note: Depending on the data length, data is increment in 1 step or 128 steps. (See Table 10-2.)

When the input data length is invalid, an optimal value is automatically set by comparing the currently displayed value to the input value.

When an invalid data length,

Example:

When 4100 is input, it is changed to 4096 and displayed.

If 4100 is input again, the data length is changed to 4224 and displayed.

Table 10-2 Numerical Relationship of Data Length

Data length	Step width	
2 to 4096	1 step	524288/128 = 4096
4224 to 524288	128 steps	128N (N = 33 to 4096)

⑯ Page number (Page)

Set value	Control message	Data request message	Output message
1 ⋮ 32768	PAGΔ1 ⋮ PAGΔ32768	PAG?	PAG ΔΔΔΔΔ 1 ⋮ PAGΔ32768
1 ⋮ 32768	ADRΔ1 ⋮ ADRΔ32768	ADR?	ADR ΔΔΔΔΔ 1 ⋮ ADRΔ32768

Restrictions

Control message

In the following cases, this message is invalid and a buzzer rings:

- When the output pattern is a PRBS pattern
- When AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message

In the following case, this message is invalid and ERR is output:

- When the output pattern is a PRBS pattern

Examples

Control message

When the output pattern is a PRGM pattern

```
OUTPUT 700; "PAGΔ1"
```

Sets the page number to 1.

```
OUTPUT 700; "ADRΔ1"
```

Sets page number to 1.

Data request message

When the page number is 1

```
OUTPUT 700; "PAG?"
ENTER 700;B$
PRINT B$
```

↓

PAGΔΔΔΔΔ1 (CR/LF) is output.

```
OUTPUT 700; "ADR?"
ENTER 700;B$
PRINT B$
```

↓

ADRΔΔΔΔΔ1 (CR/LF) is output.

When the output pattern is a PRBS pattern

```
OUTPUT 700; "PAG?"
ENTER 700;B$
PRINT B$
```

↓

ERR (CR/LF) is output.

Note: There are two comands for setting page number PAG and ADR.

They have the same function. The maximum page number depends on the number of words, word length, and data length.

The page number may not exceed the maximum allowed page number.

Example

When the data length = 32 and the displayed page numbere = 1, the maximum page number is 2.

If PAGΔ3 is input, the displayed page number is changed to page 2.

Table 10-3 Relationship between Number of Page and WORD/DATA Pattern

Output pattern	Page variation range
WORD	1 to maximum number of words, in 1 word steps
DATA	1 to value not more than (data length ÷ 16), in single steps (When remainder is 0, up to value of quotient, and when there is a remainder, up to value of quotient + 1, 1 step width) Data length — Page number 2 to 16 — 1 17 to 32 — 2 33 to 48 — 3 49 to .....

⑰ Measurement frame length (Frame length)

Set value	Control message	Data request message	Output message
4 ⋮ 32	FLNΔ4 ⋮ FLNΔ32	FLN?	FLNΔΔ4 ⋮ FLNΔ32
Step values are set in multiples of 4.			

Restrictions

Control message

In the following cases, this message is invalid and a buzzer rings:

- When the frame/word length selection is WORD
- When the measurement pattern is a PRBS pattern
- When the frame length is not a multiple of 4
- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message

In the following cases, the message is invalid and error (ERR) is output:

- When the frame/word length selection is WORD (normal mode)
- When the measurement pattern is a PRBS pattern

Examples

Control message

```
OUTPUT 700; "FLNΔ4"
```

Sets the frame length to 4.

Data request message

When the frame length is 4

```
OUTPUT 700; "FLN?"
ENTER 700;B$
PRINT B$
        ↓
```

FLNΔΔ4 (CR/LF) is output.

When the frame/word length selection is WORD (normal mode)

```
OUTPUT 700; "FLN?"
ENTER 700;B$
PRINT B$
        ↓
```

ERR (CR/LF) is output.

⑱ Pattern bit (Pattern bit)

Set value	Control message		Data request essage
0 : : 65535	BITΔ0 : : BITΔ65535	BITΔ0#H0000 : : BITΔ#HFFFF	BIT?
Output message image			
The current page number and the contents of the bits up to 8 pages are output in the following format: PAG Δ*****;BITΔ#H****,#H****,#H****,#H****,#H****,#H****,#H****,#H****			

Restrictions

Control message

In the following cases, this message is invalid and a buzzer rings:

- When the output pattern is a PRBS pattern
- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message

In the following case, this request message is invalid and ERR is output:

When the output pattern is a PRBS pattern

Examples

Control message

When the output pattern is a PRGM pattern and the bit pattern of 3 pages from the currently set page is set.

OUTPUT 700; "BITΔ10,20,30"

OUTPUT 700; "BITΔ#HFFFF,#H1000,#H2000"

Bit patterns on consecutive pages can be set by separating data from each other with a comma (,).

When setting the page number and pattern bit of 4 pages from the current page.

OUTPUT 700; "PAGΔ10;BITΔ10,20,30,40"

OUTPUT 700; "PAGΔ10;BITΔ#HFFFF,#H1000,#H2000,#H3000"

OUTPUT 700; "ADRΔ10;BITΔ10,20,30,40"

OUTPUT 700; "ADRΔ10;BITΔ#HFFFF,#H1000,#H2000,#H3000"

Data request message When the page number is 1 and the maximum page number is 29, data appears in the following format.

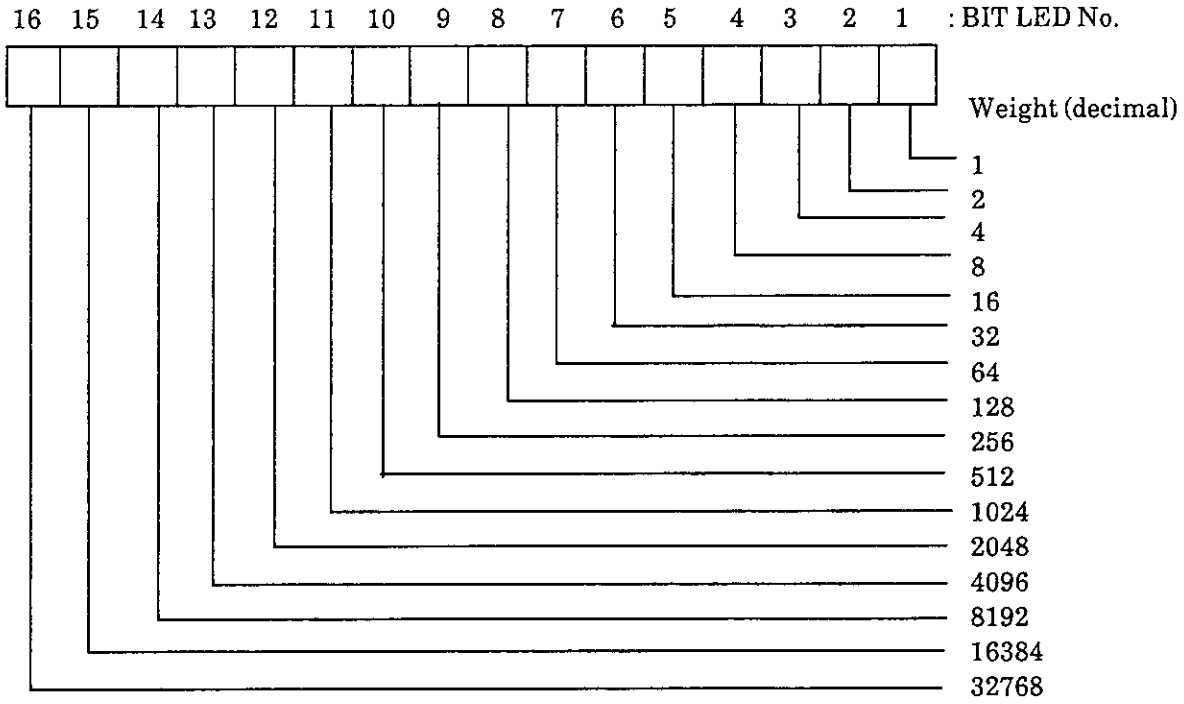
```
OUTPUT 700; "BIT?"
FOR I=1 to 4
  ENTER 700;B$
  PRINT B$
NEXT I "
      ↓
PAGΔΔΔΔΔ1;BITΔ#H0000,#H0000,#H0000,#H0000,#H0000,#H0000,#H0000,#H0000
PAGΔΔΔΔΔ9;BITΔ#H0000,#H0000,#H0000,#H0000,#H0000,#H0000,#H0000,#H0000
PAGΔΔΔΔ17;BITΔ#H0000,#H0000,#H0000,#H0000,#H0000,#H0000,#H0000,#H0000
PAGΔΔΔΔ25;BITΔ#H0000,#H0000,#H0000,#H0000,#H0000
```

When output pattern is a PRBS pattern

```
OUTPUT 700;"BIT?"
ENTER 700;B$
PRINT B$
    ↓
ERR (CR/LF) is output.
```

Set value and bit weight

The relationship between the actual bit pattern and data values are shown below:



BitΔ1	:	Only BIT LED No. 1 is set.
BitΔ16	:	Only BIT LED No. 5 is set.
BitΔ10	:	Only BIT LED's Nos. 2 and 4 are set.
BitΔ#H000F	:	Only BIT LED's Nos. 1 to 4 are set. (Decimal: 15)
BitΔ#H00F0	:	Only BIT LED's Nos. 5 to 8 are set. (Decimal: 240)
BitΔ#H1000	:	Only BIT LED No. 13 is set. (Decimal: 4096)
BitΔ#HA000	:	Only BIT LED's Nos. 14 and 16 are set. (Decimal 40960)

Note: Pattern bits are set for the page whose number is given in the NR field.

Bit patterns for consecutive pages can be separated with commas (,). However, when the output pattern is a WORD pattern, the number of pattern bits must not exceed the set word-length bits; when the output pattern is a DATA pattern, they are restricted to up to number of bits of quotient [data length /16; when remainder is 0, up to all bits.)

⑲ Number of pattern data input bytes (Pattern data write)

Set value		Control message	Data request message
Number of pattern transfer bytes	Pattern input start address		
1 : : 65536	0 : : 32767	WRTΔ1, 0 : : WRTΔ65536,32767	None
Use "WRTΔ number of pattern transfer bytes, pattern input start address" format			

Restrictions

Control message

In the following cases, this message is invalid and a buzzer rings:

- When (the number of pattern transfer bytes + pattern input start address × 2) >65536
- When the output pattern is a PRBS pattern
- When AUTO SEARCH function is ON
- When floppy disk is being accessed

Examples

Control message

When output pattern is a PRGM pattern

```
DIM B (9)
READ B (*)
DATA 1,2,4,8,16,32,64,128,256,512
OUTPUT 700; "WRTΔ20,0"
OUTPUT 700 USING "W";B(*)
```

↓

Sets the pattern data corresponding to the DATA statement for the number of pages 1 to 10.

Note: The MP1702A/MP1756A/MP1609A/MP1651A dfines the number of pattern data bytes to be transferred by DMA as well as the start address. In addition, it specifies the DMA mode and the internal RAM area storage address.

The relationship between the pattern input start address and the actual page number is:

(pattern input start address + 1) = actual page number

The DMA transfer mode is released at the end of pattern data transfer.

However, the following as exception processing is sometimes performed.

- When all pattern data has not been sent even though the number of pattern input bytes have been set, the DMA mode is released and bit 0 (command error bit) of the GP-IB status byte is set to 1 by the time-out function. Time-out occurs after 60 seconds.
- If a new control command is sent while a number of pattern input bytes is set, the new command is assumed to be the pattern data set value.

Pattern data transfer by DMA is also described in SECTION 12.

⑳ Number of pattern data output bytes (Pattern data read)

Set value		Control message	Data request message
Number of pattern transfer bytes	Pattern output start address		
1 : : 65536	0 : : 32767	REDΔ1, 0 : : REDΔ65536,32767	None
Use"REDΔ number of pattern transfer bytes, pattern output start address" format			

Restrictions

Control message

In the following cases, this message is invalid and a buzzer rings:

- When (number of pattern transfer bytes + pattern output start address × 2) >65536
- When output pattern is a PRBS pattern
- When AUTO SEARCH function is ON
- When floppy disk is being accessed

Examples

Control message

When output pattern is a PRGM pattern

```
DIM B (9)
OUTPUT 700; "REDΔ20,0"
ENTER 700 USING "W";B(*)
PRINT B (*)
```

↓

Pattern data from pages 1 to 10 are output.

Note: The number of pattern data bytes to be transferred by DMA and the output start address are defined in the NR fields. The MP1702A/MP1756A/MP1609A/MP1651A also defines the RAM storage area addresses and DMA mode switching.

The relationship between the pattern output start addresa and the page number is

(pattern output start address + 1) = Page number

The DMA transfer mode is released after completion of pattern data transfer or when the following exception occurs.

- When pattern data is not received at all, the DMA mode is released and bit 0 of the GP-IB status byte (the command error bit) is set to 1 by time-out function. Time-out occurs after 60 seconds.

Pattern data transfer by DMA is also described in SECTION 12.

㉑ Preset (all bits on all pages to 0 or 1)

Set value	Control message	Data request message	Output message
Clear all bits on all pages	ALLΔ0	None	———
Set all bits on all pages	ALLΔ1		

Restrictions

Control message

In the following cases, this message is invalid and a buzzer rings:

- When the output pattern is a PRBS pattern
- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Examples

Control message

When output pattern is a PRGM pattern

OUTPUT 700; "ALLΔ0"

Sets all bits on all pages to 0.

Note: There are 512k bits of pattern data on all pages.

㉒ Preset (all bits on a page to 0 or 1)

Set value	Control message	Data request message	Output message
Clear all bits on 1 page	PSTΔ0	None	———
Set all bits on 1 page	PSTΔ1		

Restrictions

Control message — In the following cases, this message is invalid and a buzzer rings:

- When the output pattern is a PRBS pattern
- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Examples

Control message — When the output pattern is aPRGM pattern

OUTPUT 700; "PSTΔ0"

Sets all bits on the displayed page to 0.

Note: All bits on a single screen represent all the bits on one page.

10.3 MEASUREMENT Section

The following pages describe the GP-IB messages of the MEASUREMENT section.

Δ indicates a space.

㉓ Clock loss state (Clock loss intervals?)

Set value	Control message	Data request message	Output message
When clock is not lost	———	CLI?	CLIΔ0
When clock is lost			CLIΔ1

Restrictions

Data request message　None

Examples

Data request message　When the clock is not lost

```
OUTPUT 700; "CLI?"
ENTER 700;B$
PRINT B$
```

↓

CLIΔ0 (CR/LF) is output.

㉔ Sync loss state (Sync loss intervals?)

Set value	Control message	Data request message	Output message
When sync is not lost	——	SLI?	SLIΔ0
When sync is lost			SLIΔ1

Restrictions

Data request message　None

Examples

Data request message　When sync is not lost

```
OUTPUT 700; "SLI?"
ENTER 700;B$
PRINT B$
```

↓

SLIΔ0 (CR/LF) is output.

㉕ Error detection (Errors?)

Set value	Control message	Data request message	Output message
When no erros are detected	———	ERS?	ERSΔ0
When errors are detected			ERSΔ1

Restrictions

Data request message　None

Examples

Data request message　When no errors are detected

```
OUTPUT 700; "ERS?"
ENTER 700;B$
PRINT B$
```

↓

ERSΔ0 (CR/LF) is output.

㉖ Display mode (Display or measurement)

Set value	Control message	Data request message	Output message
ERROR RATIO	DMSΔ0	DMS?	DMSΔ0
ERROR COUNT	DMSΔ1		DMSΔ1
ERROR INTERVAL	DMSΔ2		DMSΔ2
ERROR FREE INTERVAL	DMSΔ3		DMSΔ3
CLOCK FREQUENCY	DMSΔ4		DMSΔ4

Restrictions

Control message — In the following cases, this message is invalid and a buzzer rings:

- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message — None

Examples

Control message

OUTPUT 700; "DMSΔ0"

Switches the display mode to ERROR RATIO.

Data request message

When the display mode is ERROR RATIO

```
OUTPUT 700; "DMS?"
ENTER 700;B$
PRINT B$
```

↓

DMSΔ0 (CR/LF) is output.

㉗ Intermediate result display function (Current data)

Set value	Control message	Data request message	Output message
OFF	CURΔ0	CUR?	CURΔ0
ON	CURΔ1		CURΔ1

Restrictions

Control message — In the following cases, this message is invalid and a buzzer rings:

- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message — None

Examples

Control message

```
OUTPUT 700; "CURΔ0"
```

Switches the intermediate result display function OFF.

Data request message — When the intermediate result display function is OFF

```
OUTPUT 700; "CUR?"
ENTER 700;B$
PRINT B$
```

↓

CURΔ0 (CR/LF) is output.

㉘ Measurement mode (Measurement mode)

Set value	Control message	Data request message	Output message
REPEAT	MODΔ0	MOD?	MODΔ0
SINGLE	MODΔ1		MODΔ1
UNTIMED	MODΔ2		MODΔ2

Restrictions

Control message — In the following cases, this message is invalid and a buzzer rings:

- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message — None

Examples

Control message

OUTPUT 700; "MODΔ0"

Switches the measurement mode to REPEAT.

Data request message

When the measurement mode is REPEAT

```
OUTPUT 700; "MOD?"
ENTER 700;B$
PRINT B$
```

↓

MODΔ0 (CR/LF) is output.

㉙ Start measurement (Start)

Set value	Control message	Data request message	Output message
Restart (when measurement is in progress)	STA	None	——
Start (when measurement is not in progress)			

Restrictions

Control message

In the following cases, the message is invalid and a buzzer rings:

- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Examples

Control message

OUTPUT 700; "STA"

This command causes measurements to be aborted and restarted if it is issued while a measurement is in progress.

If issued while measurements are not in progress, measurement is initiated.

㉚ Stop measurement (Stop)

Set value	Control message	Data request message	Output message
Measurement stopped or terminated	STO	None	———

Restrictions

Control message

In the following cases, this message is invalid and a buzzer rings:

- When measurements are not in progress
- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Examples

Control message

OUTPUT 700; "STO"

Stops or terminates measurement.

㉛ Measurement status (Measurements in progress?)

Set value	Control message	Data request message	Output message
Measurements not in progress	———	MSR?	MSRΔ0
Measurements in progress			MSRΔ1

Restrictions

Data request message　None

Examples

Data request message　When measurement is not in progress

```
OUTPUT 700; "MSR?"
ENTER 700;B$
PRINT B$
```

↓

MSRΔ0 (CR/LF) is output.

㉜ Automatic sync function (Auto sync)

Set value	Control message	Data request message	Output message
OFF	SYNΔ0	SYN?	SYNΔ0
ON	SYNΔ1		SYNΔ1

Restrictions

Control message — In the following cases, this message is invalid and a buzzer rings:

- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message — None

Examples

Control message

OUTPUT 700; "SYNΔ0?"

Switches the automatic sync function OFF.

Data request message

When the automatic sync function is OFF

```
OUTPUT700;"SYN?"
ENTER 700;B$
PRINT B$
```

↓

SYNΔ0 (CR/LF) is output.

㉝ Selecting real- or measurement-times for display (Real time or measurement time)

Set value	Control message	Data request message	Output message
Y.M.D of internal timer	TIMΔ0	TIM?	TIMΔ0
H.M.S of internal timer	TIMΔ1		TIMΔ1
PERIOD (measurement time)	TIMΔ2		TIMΔ2
TIMED (remaining measurement time)	TIMΔ3		TIMΔ3
ELAPSED (elapsed measurement time)	TIMΔ4		TIMΔ4

Restrictions

Control message

In the following cases, this message is invalid and a buzzer rings:

- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message

None

Examples

Control message

```
OUTPUT 700; "TIMΔ0"
```

Switches the display parameter to Y.M.D display (year, month, and day of the internal timer).

Data request message

When the display parameter is Y.M.D display (year, month, and day of the internal timer).

```
OUTPUT 700; "TIM?"
ENTER 700;B$
PRINT B$
  ↓
```

TIMΔ0 (CR/LF) is output.

㉞ Internal timer setting (Rear time setting)

Setting format

RTMΔNRI, NRI, NRI, NRI, NRI, NRI

Function		Range
Second	:	0 to 59
Minute	:	0 to 59
Hour	:	0 to 23
Day	:	1 to 31 (1 to 29 for leap year)
Month	:	1 to 12
Year	:	0 to 99 (Gregorian calendar year)

Control message	Data request message	Output message
RTMΔ00, 01,01,00,00,00 : : RTMΔ99,12,31,23,59,59	RTM?	RTMΔ00, 01,01,00,00,00 : : RTMΔ99,12,31,23,59,59

Restrictions

Control message — In the following cases, this message is invalid and a buzzer rings:

- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message — None

Examples

Control message

```
OUTPUT 700; "RTMΔ90, 03, 10, 00, 10, 29"
```

Sets the internal timer to 00:10:20 on March 10, 1990 from which the clock starts counting.

Data request message — When the internal timer is 00:10:29 on March 10, 1990

```
OUTPUT 700; "RTM?"
ENTER 700;B$
PRINT B$
```

↓

RTMΔ90, 03, 10, 00, 10, 29 (CR/LF) is output.

Note: The NR field cannot be omitted.

㉟ Measurement time setting (Measurement period)

Setting format

PRDΔNRI, NRI, NRI, NRI

Function		Range
Second	:	0 to 59
Minute	:	0 to 59
Hour	:	0 to 23
Day	:	1 to 99

Control message	Data request message	Output message
PRDΔ00, 00,00,00 : RTMΔ99,23,59,59	PRD?	PRDΔ00, 00,00,00 : PRDΔ99,23,59,59

Restrictions

Control message — In the following cases, this message is invalid and a buzzer rings:

- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message — None

Examples

Control message

OUTPUT 700; "PRDΔ00, 00, 10, 29"

Sets the measurement time to 10 minutes 29 seconds.

Data request message

When the measurement time is 10 minutes 29 seconds

```
OUTPUT 700; "PRD?"
ENTER 700;B$
PRINT B$
```

↓

PRDΔ00,00,10,29 (CR/LF) is output.

Note: The NR field cannot be omitted.

10.4 MEMORY Section

The GP-IB commands of the MEMORY section are described on the following pages.

Δ indicates a space.

㊱ Memory function switching (Memory mode ptn/other)

Set value	Control message	Data request message	Output message
PTN mode	MEMΔ0	MEM?	MEMΔ0
OTH mode	MEMΔ1		MEMΔ1

Restrictions

Control message — In the following cases, this message is invalid and a buzzer rings:

- When MEMORY display is in error display mode
- When AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message — None

Examples

Control message

```
OUTPUT 700; "MEMΔ0"
```

Switches the memory function to the PTN mode.

Data request message

When memory function is in PTN mode

```
OUTPUT 700; "MEM?"
ENTER 700;B$
PRINT B$
```

↓

MEMΔ0 (CR/LF) is output.

Note: "PTN mode" means the PATTERN mode, in which only pattern data is processed.
"OTH mode" means the OTHERS mode, in which other data is processed.

㊲ File No./directory mode switching (File no./directory mode)

Set value	Control message	Data request message	Output message
FILE No.	FILΔ0	FIL?	FILΔ0
DIR	FILΔ1		FILΔ1

Restrictions

Control message

In the following cases, this message is invalid and a buzzer rings:

- When the MEMORY display is in the error display mode
- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Data request message

None

Examples

Control message

```
OUTPUT 700; "FILΔ0"
```

Switches the memory function to the FILE No. mode.

```
OUTPUT 700; "FILΔ1"
```

Switches the memory function to the DIR mode.

After the DIR mode is entered, the inserted floppy disk is accessed and the directory information is retrieved.

Data request message

When memory function is in FILE No. mode

```
OUTPUT 700; "FIL?"
ENTER 700;B$
PRINT B$
```

↓

FILΔ0(CR/LF) is output.

When memory function is in DIR mode

```
OUTPUT 700; "FIL?"
LOOP
  ENTER 700;B$
  PRINT B$
EXIT IF B$="FILΔ1"
END LOOP
```

Directory information followed by FILΔ1 (CR/LF) is output in the format shown on the next page.

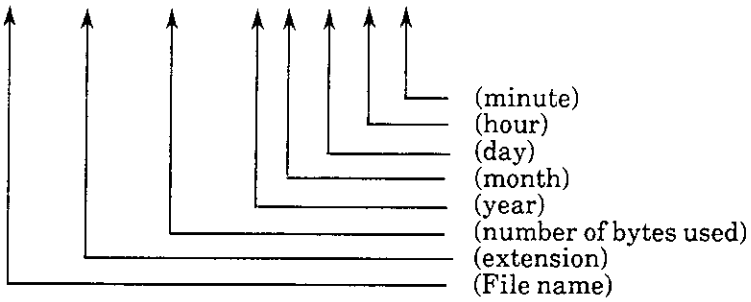
Note: The MP1702A/MP1756A/MP1609A/MP1651A does not automatically detect inserted floppy disk.

Therefore, after inserting a floppy disk, enter the directory mode to store directory information for the inserted floppy disk to RAM.

Directory information and FILΔ1 (CR/LF) output format

1. Existing directory information (output for all files)

   ********, ***, ********, **-**-**, **:** (CR/LF)



   * marks above indicate the file name, extension (file type), number of bytes used, date, and time.

   Each item is fixed as follows:

File name:	8 digits
Extension (file type):	3 digits
Number of bytes used:	8 digits
Date:	2 digits each
Time:	2 digits each

2. Setting state

   FILΔ1 (CR/LF + EOI)

Example: For actual output example

T01ΔΔΔΔΔ,PTN, ΔΔΔ65640, 89-03-27,13:45

T99ΔΔΔΔΔ,OTH,ΔΔΔΔΔ108, 89-03-27,09:01

FILΔ1

㊳ Data recall (Data recall)

File name	Control message	Data request message	Output message
0 : : 99	RCLΔ0 : : RCLΔ99	None	———

Restrictions

Control message — In the following cases, this message is invalid and a buzzer rings:

- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Examples

Control message — OUTPUT 700; "RCLΔ0"

Reads data for the specified file name from floppy disk and displays it by the set memory function.

Note: When the specified file does not exist, an error messages is displayed and the buzzer rings.

This error message is cleared when a data recall, data save, or data resave command is entered.

㊴ Data save (Data save)

File name	Control message	Data request message	Output message
0 : : 99	SAVΔ0 : : SAVΔ99	None	————

Restrictions

Control message — In the following cases, this message is invalid and a buzzer rings:

- When in directory (DIR) mode
- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Examples

Control message — When memory system is in FILE No. mode

OUTPUT 700; "SAVΔ0"

Stores the data in the specified file name under the specified memory conditions .

Note: If the specified file name already exists, an error message is displayed and the buzzer rings.

This error message is cleared when a data recall, data save, or data resave command is entered.

⑩ Data resave (Data resave)

File name	Control message	Data request message	Output message
0 ⋮ 99	RSVΔ0 ⋮ RSVΔ99	None	———

Restrictions

Control message — In the following cases, this message is invalid and a buzzer rings:

- When the AUTO SEARCH function is ON
- When floppy disk is being accessed

Examples

Control message — OUTPUT 700; "RSVΔ0"

Overwrites the specified file name with new data and memory conditions.

Note: When the specified file does not exist, an error message is displayed and the buzzer rings.

This error message is cleared when a data recall, data save, or data resave command is entered.

㊶ Floppy disk access status (Memory access condition?)

Setting value	Control message	Data request message	Output message
Not being accessed	———	MAC?	MACΔ0
Access in progress			MACΔ1

Restrictions

Data request message　None

Examples

Data request message　When floppy disk is not being accessed

```
OUTPUT 700; "MAC?"
ENTER 700;B$
PRINT B$
```

↓

MACΔ0 (CR/LF) is output.

10.5 PRINTER/ALARM MONITOR Section

The following pages describe the GP-IB messages for the PRINTER/ALARM MONITOR section.

Δ indicates a space.

㊷ Printer function (Printer on/off)

Set value	Control message	Data request message	Output message
OFF	PRNΔ0	PRN?	PRNΔ0
ON	PRNΔ1		PRNΔ1

Restrictions

Control message — In the following case, this message is invalid and a buzzer rings:

- When floppy disk is being accessed

Data request message — None

Examples

Control message

```
OUTPUT 700; "PRNΔ0"
```

Switches the built-in printer function OFF.

Data request message — When the built-in printer function is OFF

```
OUTPUT 700; "PRN?"
ENTER 700;B$
PRINT B$
```

↓

PRNΔ0 (CR/LF) is output.

㊸ Manual print (Printer start)

Set value	Control message	Data request message	Output message
Print start	PSA	None	———

Restrictions

Control message

In the following cases, this message is invalid and a buzzer rings:

- When the printer function is OFF
- When measurement is not in progress
- During the first one-second interval after the start of measurement
- When floppy disk is being accessed

Examples

Control message

OUTPUT 700; "PSA"

Outputs intermediate measurement results to the built-in printer.

㊹ Alarm monitor function (Alarm monitor on/off)

Set value	Control message	Data request message	Output message
OFF	ALMΔ0	ALM?	ALMΔ0
ON	ALMΔ1		ALMΔ1

Restrictions

Control message — In the following case, this message is invalid and a buzzer rings:

- When floppy disk is being accessed

Data request message — None

Examples

Control message

```
OUTPUT 700; "ALMΔ0"
```

Switches the alarm monitor function OFF.

Data request message — When the alarm montior function is OFF

```
OUTPUT 700; "ALM?"
ENTER 700;B$
PRINT B$
```

↓

ALMΔ0 (CR/LF) is output.

㊺ Error detect function (Errors monitor on/off)

Set value	Control message	Data request message	Output message
OFF	MONΔ0	MON?	MONΔ0
ON	MONΔ1		MONΔ1

Restrictions

Control message — In the following case, this message is invalid and a buzzer rings:

- When floppy disk is being accessed

Data request message — None

Examples

Control message

```
OUTPUT 700; "MONΔ0"
```

Switches the error detect function OFF.

Data request message — When the error detect function is OFF

```
OUTPUT 700; "MON?"
ENTER 700;B$
PRINT B$
```

↓

MONΔ0 (CR/LF) is output.

10.6 MEASURE CH MASK SELECT Section

The following pages describe the GP-IB messages for the MEASURE CH MASK SELECT section.

Δ indicates a space.

㊻ Mask route length selection (Mask route length select)

Set value	Control message	Data request message	Output message
8 route length	MSKΔ0	MSK?	MSKΔ0
16-route length (for using 1ch to 8ch)	MSKΔ1		MSKΔ1
16-route length (for using 9ch to 16ch)	MSKΔ2		MSKΔ2

Restrictions

Control message — In the following case, this message is invalid and a buzzer rings:

- When floppy disk is being accessed

Data request message — None

Examples

Control message

OUTPUT 700; "MSKΔ0"

Switches the mask route length to the 8-route length.

Data request message

When the mask route length is 8-route long

```
OUTPUT 700; "MSK?"
ENTER 700;B$
PRINT B$
```

↓

MSKΔ0 (CR/LF) is output.

㊼ Number of mask route channels (Measure ch mask select)

When the mask route length is 8-routes long or l16-routes long (1ch to 8ch)			
Set value	**Control message**	**Data request message**	**Output message**
1 to 8ch (OFF) 1 ch only (ON) : : 2 to 8ch (ON) 1 to 8ch (ON)	SCHΔ#B00000000 SCHΔ#B00000001 : : SCHΔ#B11111110 SCHΔ#B11111111	SCH?	SCHΔ#B00000000 SCHΔ#B00000001 : : SCHΔ#B11111110 SCHΔ#B11111111
Each error counter is either masked or not masked depending on the value of each bit in the NR field. 0: OFF (not masked) 1: ON (masked)			

When the mask route length is 16 routes long (9ch to 16ch)			
Set value	**Control message**	**Data request message**	**Output message**
9 to 16ch (OFF) 9ch only (ON) : : 10 to 16ch (ON) 9 to 16ch (ON)	SCHΔ#B00000000 SCHΔ#B00000001 : : SCHΔ#B11111110 SCHΔ#B11111111	SCH?	SCHΔ#B00000000 SCHΔ#B00000001 : : SCHΔ#B11111110 SCHΔ#B11111111
Each error counter is either masked or not masked depending on the value of each bit in the NR field. 0: OFF (not masked) 1: ON (masked)			

Restrictions

Control message
- When floppy disk is being accessed

Data request message
None

Examples

Control message

When the mask route length is 8-routes long

OUTPUT 700; "SCHΔ#B00000001"

Switches only one channel of the error counter ON (masked).

Data request message

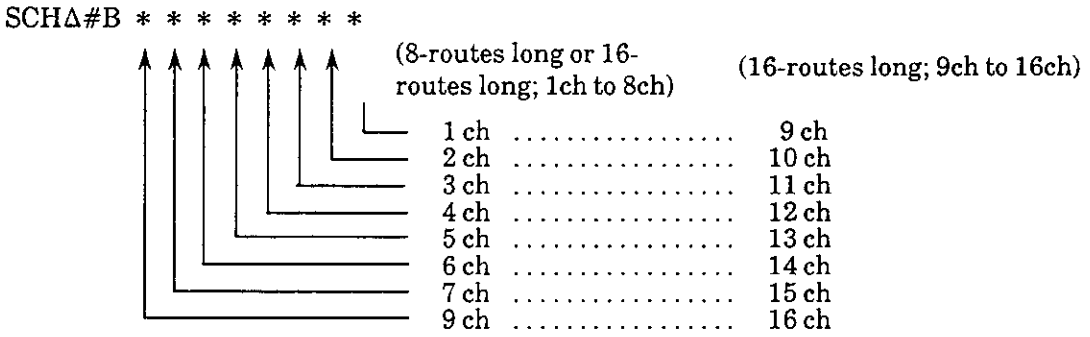
When the mask route length is 8-routes long and channels 1 and 5 of the error counter are ON (masked)

```
OUTPUT 700; "SCH?"
ENTER 700;B$
PRINT B$
    ↓
    MSKΔ0;SCHΔ#B00010001 (CR/LF) is output.
```

When the mask route length is 16-routes long and channels 1, 5 and 15 of the error counter are ON (masked)

```
OUTPUT 700; "SCH?"
FOR I = 1 TO 2
    ENTER 700;B$
    PRINT B$
NEXT I
    ↓
    MSKΔ1;SCHΔ#B00010001 (CR/LF)
    MSKΔ2;SCHΔ#B01000000 (CR/LF + EOI) are output.
```

Note:



- When setting multiple conditions, combine the above channels.
- Before setting the number of mask route channels, be sure to select the mask route length.
- The LED display of mask route channels indicates the setting that correspond to the selected mask route length.

10.7 FUNCTION Section

The following pages described the GP-IB messages for the FUNCTION section.

Δ indicates a space.

㊽ Number of mark ratio AND bit shifts (Mark ratio bit shift)

Set value	Control message	Data request message	Output message
1-bit shift	SFTΔ0	SFT?	SFTΔ0
3-bit shift	SFTΔ1		SFTΔ1

Restrictions

Control message

In the following cases, this message is invalid and a buzzer rings:

When the output pattern is a PRGM. (WORD or DATA) pattern

When floppy disk is being accessed.

Data request message

In the following case, this request is invalid and ERR is output:

When the output pattern is a PRGM. (WORD or DATA) pattern

Examples

Control message

When the output pattern is a PRBS pattern

OUTPUT 700; "SFTΔ0"

Sets the number of mark ratio AND bit shifts to a 1 bit shift.

Data request message

When number of mark ratio AND bit shifts is 1 bit shift

```
OUTPUT 700; "SFT?"
ENTER 700;B$
PRINT B$
```

↓

SFTΔ0 (CR/LF) is output.

When the output pattern is a PRGM. (WORD or DATA) pattern

```
OUTPUT 700; "SFT?"
ENTER 700;B$
PRINT B$
```

↓

ERR (CR/LF) is output.

㊾ Clock loss processing function (Clock loss processing option)

Set value	Control message	Data request message	Output message
EXCLUDE	CLSΔ0	CLS?	CLSΔ0
INCLUDE	CLSΔ1		CLSΔ1

Restrictions

Control message — In the following case, this message is invalid and a buzzer rings:
- When floppy disk is being accessed

Data request message — None

Eamples

Control message

```
OUTPUT 700; "CLSΔ0"
```

Excludes clock loss processing from calculations.

Data request message — When clock loss processing is excluded from calculations

```
OUTPUT 700; "CLS?"
ENTER 700;B$
PRINT B$
```

↓

CLSΔ0 (CR/LF) is output.

Note: When the EXCLUDE option is set, clock loss processing is excluded from calculations.
When the INCLUDE option is set, clock loss processing is included in calculations.

⑤⓪ Sync loss processing function (Sync loss processing option)

Set value	Control message	Data request message	Output message
EXCLUDE	SLSΔ0	SLS?	SLSΔ0
INCLUDE	SLSΔ1		SLSΔ1

Restrictions

Control message — In the following case, this message is invalid and a buzzer rings:
- When floppy disk is being accessed

Data request message — None

Eamples

Control message

```
OUTPUT 700; "SLSΔ0"
```

Excludes sync loss processing from calculations.

Data request message — When sync loss processing is excluded from calculations

```
OUTPUT 700; "SLS?"
ENTER 700;B$
PRINT B$
```

↓

SLSΔ0 (CR/LF) is output.

Note: When the EXCLUDE option is set, sync loss processing is excluded from calculations.
When the INCLUDE option is set, sync loss processing is included in calculations.

⑤① Error performance threshold selection (Error performance threshold)

Set value	Control message	Data request message	Output message
10E – 3	ETHΔ0	ETH?	ETHΔ0
10E – 4	ETHΔ1		ETHΔ1

Restrictions

Control message: In the following case, this message is invalid and a buzzer rings:
- When floppy disk is being accessed

Data request message: None

Eamples

Control message:

OUTPUT 700; "ETHΔ0"

Switches the error performance measurement threashold to 10E-3.

Data request message: When the error performance measurement threashold is 10E-3

```
OUTPUT 700; "ETH?"
ENTER 700;B$
PRINT B$
```

↓

ETHΔ0 (CR/LF) is output.

⑤② Display digit selection (Mantissa display digit)

Set value	Control message	Data request message	Output message
5-digit display	DGTΔ0	DGT?	DGTΔ0
2-digit display	DGTΔ1		DGTΔ1

Restrictions

Control message — In the following case, this message is invalid and a buzzer rings:
- When floppy disk is being accessed

Data request message — None

Eamples

Control message

```
OUTPUT 700; "DGTΔ0"
```

Switches to 5-digit display.

Data request message — When a 5-digit display is used

```
OUTPUT 700; "DGT?"
ENTER 700;B$
PRINT B$
```

↓

DGTΔ0 (CR/LF) is output.

⑤③ Intermediate data calculation (Current data calculation)

Set value	Control message	Data request message	Output message
Cumulative calculation data	CALΔ0	CAL?	CALΔ0
Current calculation data only	CALΔ1		CALΔ1

Restrictions

Control message — In the following case, this message is invalid and a buzzer rings:
- When floppy disk is being accessed

Data request message — None

Eamples

Control message — OUTPUT 700; "CALΔ0"

Switches calculation of intermediate data to cumulative calculation data.

Data request message — When intermediate data calculation is made for cumulative calculation data

```
OUTPUT 700; "CAL?"
ENTER 700;B$
PRINT B$
```

↓

CALΔ0 (CR/LF) is output.

⑭ Error detection mode selection (Error type)

Set value	Control message	Data request message	Output message
Total error	ETYΔ0	ETY?	ETYΔ0
Insertion error	ETYΔ1		ETYΔ1
Omission error	ETYΔ2		ETYΔ2

Restrictions

Control message — In the following case, this message is invalid and a buzzer rings:
- When floppy disk is being accessed

Data request message — None

Eamples

Control message

```
OUTPUT 700; "ETYΔ0"
```

Switches the error detection mode to TOTAL ERROR.

Data request message — When the error detection mode is TOTAL ERROR

```
OUTPUT 700; "ETY?"
ENTER 700;B$
PRINT B$
```

↓

ETYΔ0 (CR/LF) is output.

⑮ Measurement interval time selection (Interval time)

Set value	Control message	Data request message	Output message
100 ms	ITVΔ0	ITV?	ITVΔ0
200 ms	ITVΔ1		ITVΔ1
500 ms	ITVΔ2		ITVΔ2

Restrictions

Control message — In the following case, this message is invalid and a buzzer rings:
- When floppy disk is being accessed

Data request message — None

Eamples

Control message

```
OUTPUT 700; "ITVΔ0"
```

Switches the measurement interval time to 100 ms.

Data request message — When the measurement interval time is 100 ms

```
OUTPUT 700; "ITV?"
ENTER 700;B$
PRINT B$
```

↓

ITVΔ0 (CR/LF) is output.

⑯ Data print format selection (Output data format)

Set value	Control message	Data request message	Output message
Standard format	FMTΔ0	FMT?	FMTΔ0
Abridged (short) format	FMTΔ1		FMTΔ1

Restrictions

Control message: In the following case, this message is invalid and a buzzer rings:
- When floppy disk is being accessed

Data request message: None

Eamples

Control message:

```
OUTPUT 700; "FMTΔ0"
```

Sswitches the data print format to standard format.

Data request message: When the data print format is standard format

```
OUTPUT 700; "FMT?"
ENTER 700;B$
PRINT B$
```

↓

FMTΔ0 (CR/LF) is output.

⑤⑦ Threshold EI/EFI data print selection (Threshold EI/EFI data output)

Set value	Control message	Data request message	Output message
OFF	THRΔ0	THR?	THRΔ0
ON	THRΔ1		THRΔ1

Restrictions

Control message: In the following case, this message is invalid and a buzzer rings:
- When floppy disk is being accessed

Data request message: None

Eamples

Control message:

```
OUTPUT 700; "THRΔ0"
```

Does not print threshold EI/EFI.

Data request message: When threshold EI/EFI is not printed

```
OUTPUT 700; "THR?"
ENTER 700;B$
PRINT B$
```

↓

THRΔ0 (CR/LF) is output.

⑱ Error performance data print selection (Error performance data output)

Set value	Control message	Data request message	Output message
OFF	EPFΔ0	EPF?	EPFΔ0
ON	EPFΔ1		EPFΔ1

Restrictions

Control message — In the following case, this message is invalid and a buzzer rings:
- When floppy disk is being accessed

Data request message — None

Eamples

Control message

```
OUTPUT 700; "EPFΔ0"
```

Does not print error performance.

Data request message — When error performance data is not printed

```
OUTPUT 700; "EPF?"
ENTER 700;B$
PRINT B$
```

↓

EPFΔ0 (CR/LF) is output.

Note: Error performance data are output only when the error performance data print selection is ON and the measurement time is 60 seconds or more.

⑤⑨ Intermediate data print selection (Intermediate data output option)

Set value	Control message	Data request message	Output message
OFF	ITMΔ0	ITM?	ITMΔ0
ON	ITMΔ1		ITMΔ1

Restrictions

Control message — In the following case, this message is invalid and a buzzer rings:
- When floppy disk is being accessed

Data request message — None

Eamples

Control message

```
OUTPUT 700; "ITMΔ0"
```

Does not output intermediate data to the built-in printer.

Data request message — When intermediate data is not output to the built-in printer

```
OUTPUT 700; "ITM?"
ENTER 700;B$
PRINT B$
```

↓

ITMΔ0 (CR/LF) is output.

Note: Intermediate measurement results are output to the built-in printer as follows when both the intermediate data print function and printer function are ON.

< For repeat or single measurements >

- Results are printed every two hours when the measurement time is less than two days.
- Results are printed every day when the measurement time is two days or more.

< For manual (untimed) measurements >

- Results are printed every day.

For details on print formats for the built-in printer, refer to paragraph 5.6.3 in the separate Operations Manual.

⑥⓪ One-second data print selection (Output option for one second data)

Set value	Control message	Data request message	Output message
OFF	OSCΔ0	OSC?	OSCΔ0
ON	OSCΔ1		OSCΔ1

Restrictions

Control message — In the following case, this message is invalid and a buzzer rings:
- When floppy disk is being accessed

Data request message — None

Eamples

Control message

```
OUTPUT 700; "OSCΔ0"
```

Does not print the one-second data.

Data request message — When the one-second data is not printed

```
OUTPUT 700; "OSC?"
ENTER 700;B$
PRINT B$
```

↓

OSCΔ0 (CR/LF) is output.

⑥1 One-second data print threshold selection (Data output threshold)

Set value	Control message	Data request message	Output message
Error > 0	DOTΔ0	DOT?	DOTΔ0
Error > 10E-6	DOTΔ1		DOTΔ1
Error > 10E-4	DOTΔ2		DOTΔ2
Error > 10E-3	DOTΔ3		DOTΔ3

Restrictions

Control message — In the following case, this message is invalid and a buzzer rings:
- When floppy disk is being accessed

Data request message — None

Eamples

Control message

```
OUTPUT 700; "DOTΔ0"
```

Switches the one-second data threshold to ERROR > 0.

Data request message — When the one-second data threshold is ERROR > 0

```
OUTPUT 700; "DOT?"
ENTER 700;B$
PRINT B$
```

↓

DOTΔ0 (CR/LF) is output.

⑥② Paper saving function (Paper saving option)

Set value	Control message	Data request message	Output message
OFF	PSVΔ0	PSV?	PSVΔ0
ON	PSVΔ1		PSVΔ1

Restrictions

Control message — In the following case, this message is invalid and a buzzer rings:
- When floppy disk is being accessed

Data request message — None

Eamples

Control message — OUTPUT 700; "PSVΔ0"

Switches the paper saving function for one-second data print OFF.

Data request message — When the paper saving function for one-second data print is OFF

```
OUTPUT 700; "PSV?"
ENTER 700;B$
PRINT B$
        ↓
PSVΔ0 (CR/LF) is output.
```

⑥③ Measurement result GP-IB output selection (Measurement type?)

Output method set value	Control message	Data request message	Output message
Each time measurement is completed	———	MTY?	MTYΔ0
Each time data request command is issued			MTYΔ1

Restrictions

Data request message None

Eamples

Data request message When the function is set to output measurement results each time measurement is completed

```
OUTPUT 700; "MTY?"
ENTER 700;B$
PRINT B$
```

↓

MTYΔ0 (CR/LF) is output.

Note: Use SW9 of FUNCTION2 on the rear panel to select the regularity with which measurement-results are output by GP-IB.

Setting of FUNCTION2 SW9	Output method
OFF	Output each time measurement is completed
ON	Output only when data request is issued

When the SW9 switch setting is changed, turn the power off and then on again, or send the device clear command to read the new value.

10.8 Other Sections

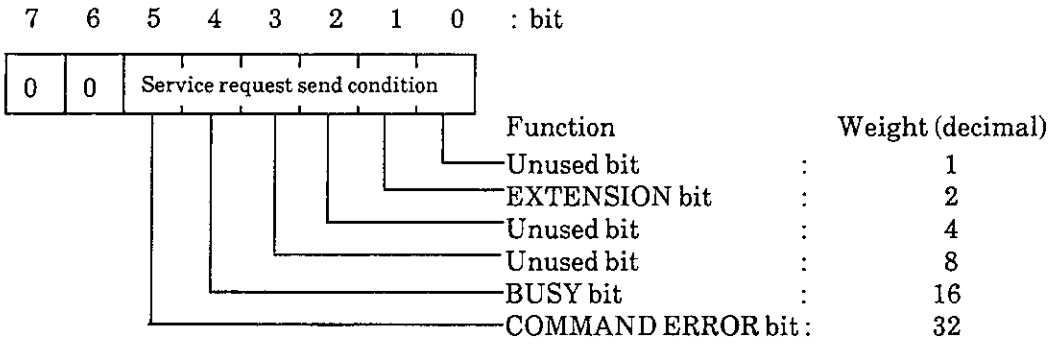
The GP-IB commands of other sections are described on the following pages.

Δ indicates space.

⑭ Service request enable register (Service request enable register)

Setting State

Service request enable register configuration



Function

Weight 0: Turns the service request send condition OFF.

1: Unused bit

2: Sends a service request when the extended event status register is set.

4: Unused bit

8: Unused bit

16: Sends a service request when there is data to be output via the GP-IB.

32: Sends a service request when the standard event status register is set.

Notes:

- When an SRQ command is received, all the previously set send conditions are cleared.
- When setting multiple service-request conditions total the weight values above.
- Immediately after the power is turned ON, or immediately after a device clear or initialize command is received, all SRQ send conditions are cleared.

Control message	Data request message	Output message
SRQΔ 0 : : SRQΔ63	SRQ?	SRQΔΔ0 : : SRQΔ63

Restrictions

Control message — In the following case, this message is invalid:

When floppy disk is being accessed.

Data request message — None

Examples

Control message — OUTPUT 700; "SRQΔ2"

Sends a service request when the extended event status register is set.

Data request message — When the extended event status register is set, a service request send condition is sent

```
OUTPUT 700; "SRQ?"
ENTER 700;B$
PRINT B$
```

↓

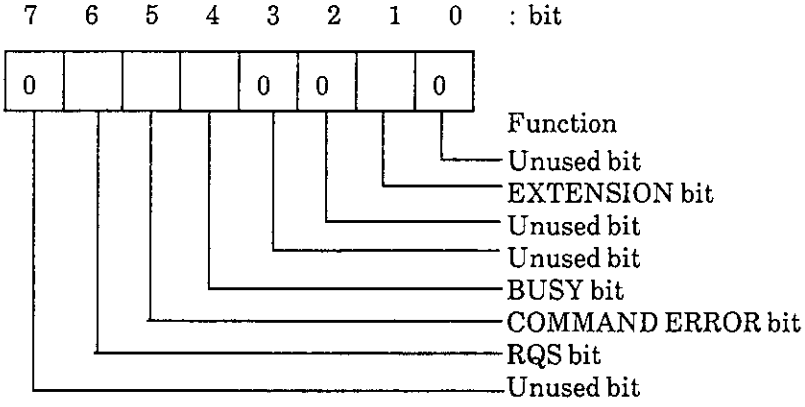
SRQΔΔ2 (CR/LF) is output.

Note: The service request enable register and GP-IB status byte are described in SECTION 11.

⑥⑤ Status byte register (Status byte register?)

Setting State

Status byte register configuration



Control message	Data request message	Output message
———	STB?	STBΔ#B********

Restrictions

Data request message — None

Examples

Data request message — When GP-IB status byte is 01010000

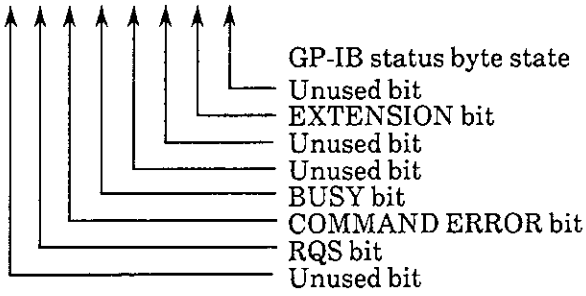
```
OUTPUT 700; "STB?"
ENTER 700;B$
PRINT B$
```

↓

TBΔ#B01010000 (CR/LF) is output.

Note: This GP-IB status byte is also read during serial pooing.

STBΔ#B * * * * * * * *

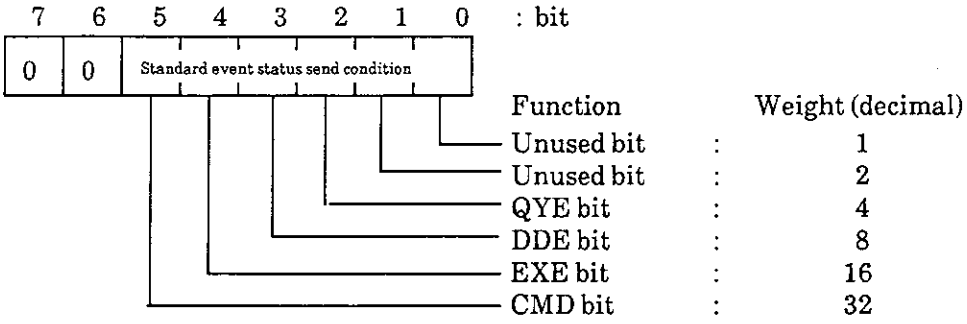


The status byte register and GP-IB status byte are described in SECTION 11.

⑯ Standard event status enable register (Event status enable register [Standard])

Setting State

Standard event status enable register configuration



Function

Weight 0: Turns the standard event status send condition OFF.

1: Unused bit

2: Unused bit

4: Sends the standard event status send condition when a talker is designated (even though the send buffer is empty) or when the send buffer has overflowed.

8: Sends the standard event status send condition when the received command is invalid under current conditions.

16: Sends the standard event status send condition if the numerical data format is wrong or if a HEX value is used when it is not allowed.

32: Sends the standard event status send condition when the header is erroneous.

Note:

- When the ESE command is received, all previously set send conditions are cleared.
- When setting multiple event-status-enable (ESE) conditions, total the weight values above.
- All the ESE send conditions are cleared immediately after the power is turned ON, or immediately after a device clear or initialize command is received.

Control message	Data request message	Output message
ESEΔ 0 : ESEΔ63	ESE?	ESEΔΔ0 : ESEΔ63

Restrictions

Control message

In the following case, this message is invalid.

When floppy disk is being accessed

Data request message

None

Examples

Control message

OUTPUT 700; "ESEΔ32"

Sends the standard event status byte to the controller when a header error is generated.

Data request message

When the standard event status send byte is output in response to a header error

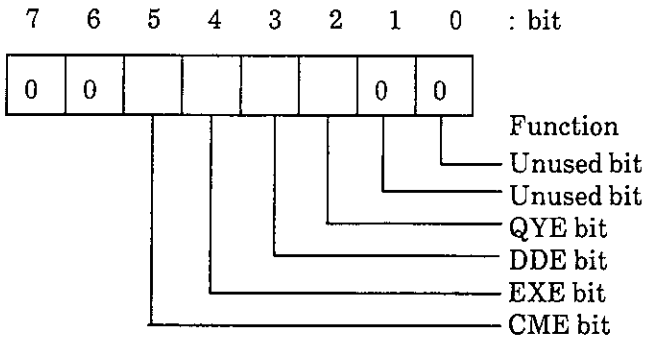
```
OUTPUT 700; "ESE?"
ENTER 700;B$
PRINT B$
```

↓

ESEΔ32 (CR/LF) is output.

Note: The standard event status enable register and GP-IB status byte are described in SECTION 11.

⑥⑦ Standard event status register (Event status byte register? [Standard])

Setting state
Status byte register configuration    ***Note:*** • After this status byte has been read, all bits are reset. • Otherwise, the standard event status byte is not cleared even when a device clear, interface clear, or initialize command is received.

Control message	Data request message	Output message
————	ESR?	ESRΔ#B********

Restrictions

Data request message None

Examples

Data request message When value of standard event status byte is 00100000

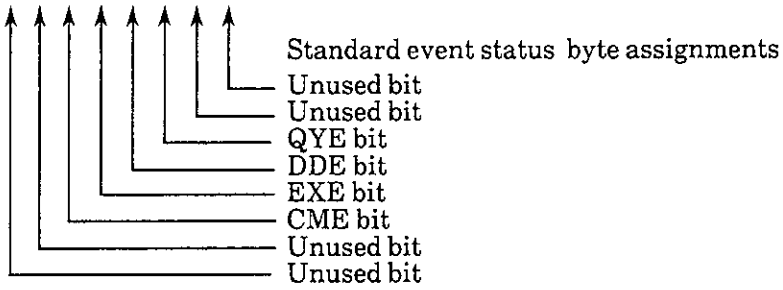
```
OUTPUT 700; "ESR?"
ENTER 700;B$
PRINT B$
```

↓

ESRΔ#B00100000 (CR/LF) is output.

Note: The standard event status byte is irrelevant to serial polling. This status byte is considered to be an extension of the COMMAND ERROR bit of the GP-IB status byte.

ESRΔ#B * * * * * * * *

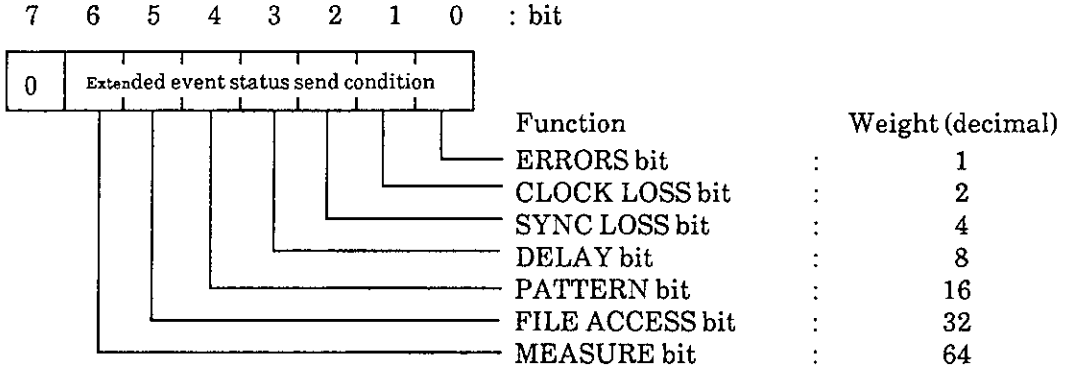


The standard event status byte register and GP-IB status byte are described in SECTION 11.

⑱ Extended event status enable register (Event status enable register [Extension])

Setting state

Extended event status enable register configuration



Function

Weight 0: Disables sending of the extended event status (EES) byte.

1: Sends the EES byte when an error is detected.

2: Sends the EES byte at clock loss and clock recovery.

4: Sends the EES byte at sync loss and sync recovery.

8: Sends the EES byte when the servo circuit of the clock delay circuit changes from the BUSY to the READY state.

16: Sends the EES byte at completion of programmable pattern setting.

32: Sends the EES byte at completion of a floppy disk.

64: Sends the ESS byte at the end of measurement.

Note:

- After an EES command is received, all previously set send conditions are cleared.
- When setting multiple EES conditions, set all appropriate bits.
- Immediately after the power is turned ON or immediately after a device clear or initialize command is received, all EES send conditions are cleared.

Control message	Data request message	Output message
EESΔ 0 : : EESΔ63	EES?	EESΔΔ0 : : EESΔ63

Restrictions

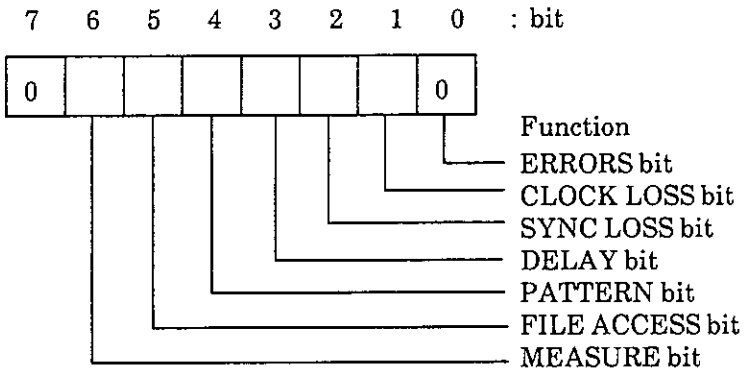
Control message	In the following case, this message is invalid: When floppy disk is being accessed
Data request message	None

Examples

Control message	OUTPUT 700; "EESΔ32" Sends the extended event status byte on completion of a floppy disk access.
Data request message	When the EES byte is sent on completion of a floppy disk access OUTPUT 700; "EES?" ENTER 700;B$ PRINT B$ ↓ EESΔ32 (CR/LF) is output.

Note: The extended event status enable register and GP-IB status byte are described in SECTION 11.

⑥⑨ Extended event status register (Event status byte register? [Extension])

Setting state
Status byte register configuration    ***Note:*** • After this status byte has been read, all bits are reset. • Otherwise, this register is not cleared even when a device clear, interface clear, or initialize command is received.

Control message	Data request message	Output message
————	EER?	EERΔ#B********

Restrictions

Data request message None

Examples

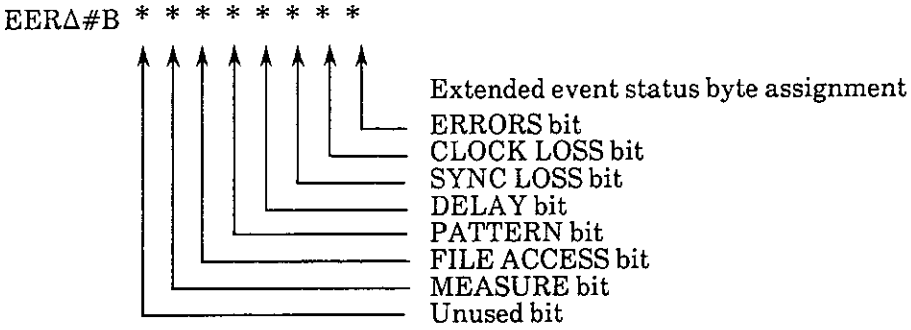
Data request message When the value of the extended event status byte is 01000000

```
OUTPUT 700; "EER?"
ENTER 700;B$
PRINT B$
```

↓

EERΔ#B00100000 (CR/LF) is output.

Note: The extended event status byte is irrelevant to serial polling. This status byte is considered to be an extension of the EXTENSION bit of the GP-IB status byte.



The extended event status byte register and GP-IB status byte are described in SECTION 11.

⑦⓪ INITIALIZE (Initialize)

Setting state	Control message	Data request message	Output Message
Initialize	INI	————	————

Restrictions

Control message None

Examples

Control message OUTPUT 700; "INI"

Reinitializes the MP1702A/MP1756A/MP1609A/MP1651A to factory settings.

Note: This is the same as the initialization operation performed when the power is turned ON while pressing the [LOCAL] key.

If the floppy disk is being accessed when this command is issued, the access is aborted.

⑦① Intermediate measurement result output function (Output intermediate measurement data?)

Function	Outputs intermediate measurement results to GP-IB.

Control message	Data request message	Output format
———	IMD?	As shown in SECTION 13

Restrictions

Data request message

When the rear-panel FUNCTION switch (for selecting when and how measurement results will be output via GP-IB) is set for output by data request command

- When measurements are not in progress
- For the first one-second interval after start of measurement

Examples

Data request mesage

When the data print format is the standard format set for output of standard data

```
OUTPUT 700;"IMD?"
LOOP
    ENTER 700;B$
    PRINT B$
EXIT IF B$ (1:3) = "SLI"
END LOOP
    ↓
```

The above code causes the following to be output:

```
STA 90, 02, 26, 12, 35, 27 (CR/LF)
STA 90, 02, 26, 12, 35, 27 (CR/LF)
INT 90, 02, 26, 12, 35, 37 (CR/LF)
ELP       0, 00, 00, 10 (CR/LF)
ER   1. 0000E - 04 (CR/LF)
EC        5000011 (CR/LF)
EI              10 (CR/LF)
EFI        0.0000 (CR/LF)
PFI              0 (CR/LF)
CLI              0 (CR/LF)
SLI              0 (CR/LF + EOI)
```

Data request message (Cont.)	When the rear-panel FUNCTION switch (for selecting when and how measurement results will be output via GP-IB) is set to output measurement results each time a measurement is completed OUTPUT 700; "IMD?" ENTER 700;B$ PRINT B$ ↓ ERR (CR/LF) is output

Note: This function operates as though the manual print (43).

⑫ Clear measurement-result-buffer function (Measurement buffer clear)

Function	Clears the buffer that stores measurement results

Control message	Data request message	Output format
BCL	———	———

Restrictions

Control message

In the following cases, this message is invalid and a buzzer rings:

- When the rear-panel FUNCTION switch (for selecting when and how measurement results will be output via GP-IB) is set for output by data request command
- When a measurement is in progress
- When floppy disk is being accessed

Examples

Control message

OUTPUT 700;"BCL"

Clears the buffer that stores measurement results.

Note: The measurement results that have already been output to the talker task cannot be cleared.

⑦③ Error ratio measurement results (Error ratio?)

Function	Outputs results of error ratio measurements in the designated format. Note that the output value is the value (final data or current data) displayed on the 7-segment LED driver.

Control message	Data request message	Output format
———	ER?	ERΔΔ*, ****E—** *Indicates the value of the error ratio whose format is fixed to represet four decimal digit exponential numbers.

Restrictions

Data request message

- When the rear-panel FUNCTION switch (for selecting when and how measurement results will be output via GP-IB) is set to initiate output each time a measurement is completed

Examples

Data request message

- When the error ratio is 1.05 × 10^-6

```
OUTPUT 700;"ER?"
ENTER 700;B$
PRINT B$
  ↓
ERΔΔ1.0500E-06(CR/LF) is output.
```

- When the error ratio display is "—"

```
OUTPUT 700;"ER?"
ENTER 700;B$
PRINT B$
  ↓
ERΔΔ0.0000E-00(CR/LF) is output.
```

- When the rear-panel FUNCTION switch (for selecting when and how measurement results will be output via GP-IB) is set to initiate output each time a measurement is completed

```
OUTPUT 700;"ER?"
ENTER 700;B$
PRINT B$
  ↓
ERR(CR/LF) is output.
```

⑦④ Error count measurement results (Error count?)

Function	Outputs results of error count measurement in the designated format. Note that the output value is the value (final data or current data) displayed on the 7-segment LED driver.

Control message	Data request message	Output format
———	EC?	• When the error count is less than 1E+10 ECΔΔ********** * indicates a digit of the error count. • This 10-digit fixed-length format is used when the error count is 1E+10 or more. ECΔΔ*.****E+** *Indicates a digit of the error count. In this format, the error count is represented by four decimal digit exponential numbers.

Restrictions

Data request message

- When the rear-panel FUNCTION switch (for selecting when and how measurement results will be output via GP-IB) is set to initiate output each time a measurement is completed

Examples

Data request message

- When the error count is 10

```
OUTPUT 700;"EC?"
ENTER 700;B$
PRINT B$
  ↓
ECΔΔΔΔΔΔΔΔΔΔ10(CR/LF) is output.
```

- When the error count is $1.05 \times 10^{+6}$

```
OUTPUT 700;"EC?"
ENTER B$
PRINT B$
  ↓
ECΔΔ1.0500E+06(CR/LF) is output.
```

- When the error count display is "–"

  OUTPUT 700;"EC?"

  ENTER 700;B$

  PRINT B$

  ↓

  ECΔΔ1.0000E-99(CR/LF) is output.

- When the rear-panel FUNCTION switch (for selecting when and how measurement results will be output via GP-IB) is set to initiate output each time a measurement is completed

  OUTPUT 700;"EC?"

  ENTER 700;B$

  PRINT B$

  ↓

  ERR(CR/LF) is output.

⑮ Error intervals measurement results (Error intervals?)

Function	Outputs results of error intervals measurement in the designated format. Note that the output value is the value (final data or intermediate data) displayed on the 7-segment LED driver.

Control message	Data request message	Output format
———	EI?	EIΔΔ********** * indicates a digit of the error intervals value. The output is a fixed-length 10-digit decimal value.

Restrictions

Data request message

- When the rear-panel FUNCTION switch (for selecting when and how measurement results will be output via GP-IB) is set to initiate output each time a measurement is completed

Examples

Data request message

- When the error interval is 10

```
OUTPUT 700;"EI?"
ENTER 700;B$
PRINT B$
  ↓
EIΔΔΔΔΔΔΔΔΔΔ10(CR/LF) is output.
```

- When the error interval is "—"

```
OUTPUT 700;"EI?"
ENTER 700;B$
PRINT B$
  ↓
EIΔΔ1.0000E-99(CR/LF) is output.
```

- When the rear-panel FUNCTION switch (for selecting when and how measurement results will be output via GP-IB) is set to initiate output each time a measurement is completed

```
OUTPUT 700;"EI?"
ENTER 700;B$
PRINT B$
  ↓
ERR(CR/LF) is output.
```

⑯ Error free interval measurement results (Error free intervals?)

Function	Outputs the results of the error-free interval measurement in the designated format. Note that the output value is the value (final data or current data) displayed on the 7-segment LED driver.

Control message	Data request message	Output format
———	EFI?	EFIΔΔΔ***.**** *Indicates a digit of the error-free interval value. In this format, the error-free interval is represented by four decimal digit numbers.

Restrictions

Data request message

- When the rear-panel FUNCTION switch (for selecting when and how measurement results will be output via GP-IB) is set to initiate output each time a measurement is completed

Examples

Data request message

- When the error-free interval is 99.01 %

```
OUTPUT 700;"EFI?"
ENTER 700;B$
PRINT B$
```

↓

EFIΔΔΔΔ99.0100(CR/LF) is output.

- When the error-free interval is "–"

```
OUTPUT 700;"EFI?"
ENTER 700;B$
PRINT B$
```

↓

EFIΔ1.0000E-99(CR/LF) is output.

- When the rear-panel FUNCTION switch (for selecting when and how measurement results will be output via GP-IB) is set to initiate output each time a measurement is completed

```
OUTPUT 700;"EFI?"
ENTER 700;B$
PRINT B$
```

↓

ERR(CR/LF) is output.

⑦⑦ One-second clock frequency measurement result (Clock frequency?)

Function	Outputs results of one-second clock frequency measurement in the designated format. Note that the output value is the value (clock frequency data for one second) displayed on the 7-segment LED driver.

Control message	Data request message	Output format
———	FRQ?	FRQΔΔ*****.*** *Indicates a digit of one-second clock frequency data. The output is a fixed-length three-digit decimal value.

Restrictions

Data request message

- When the rear-panel FUNCTION switch (for selecting when and how measurement results will be output via GP-IB) is set to initiate output each time a measurement is completed

Examples

Data request message

- When the one-second clock frequency is 3000 MHz

```
OUTPUT 700;"FRQ?"
ENTER 700;B$
PRINT B$
  ↓
FRQΔΔΔ3000.000(CR/LF) is output.
```

- When the one-second clock frequency is "–"

```
OUTPUT 700;"FRQ?"
ENTER 700;B$
PRINT B$
  ↓
FRQΔΔΔΔΔΔ0.000(CR/LF) is output.
```

- When the rear-panel FUNCTION switch (for selecting when and how measurement results will be output via GP-IB) is set to initiate output each time a measurement is completed

```
OUTPUT 700;"FRQ?"
ENTER 700;B$
PRINT B$
  ↓
ERR(CR/LF) is output.
```

SECTION 11
GP-IB STATUS BYTE

When the MP1702A/MP1756A/MP1609A/MP1651A is serially polled, it outputs a status byte to the controller. In addition, it has a standard event and extended event status bytes that are output in response to data request commands.

Note: When performing serial polls continuously, a WAIT time of about 0.1 s must be provided between serial polls.

If serial polling is performed continuously without a WAIT time, the polling is continuously executed because the polling condition is evaluated before information is sent to the slave tasks.

Example:

```
    .
    .
LOOP
    A=SPOLL (700)
EXIT IF BIT (A, 6)=1
    WAIT . 1 <------------ Indicates WAIT time.
END LOOP
    .
    .
```

Status information can also be read by the data request message STB? without performing serial polling.

11.1 Status Byte Configuration

The contents of each status byte is shown in Fig. 11-1.

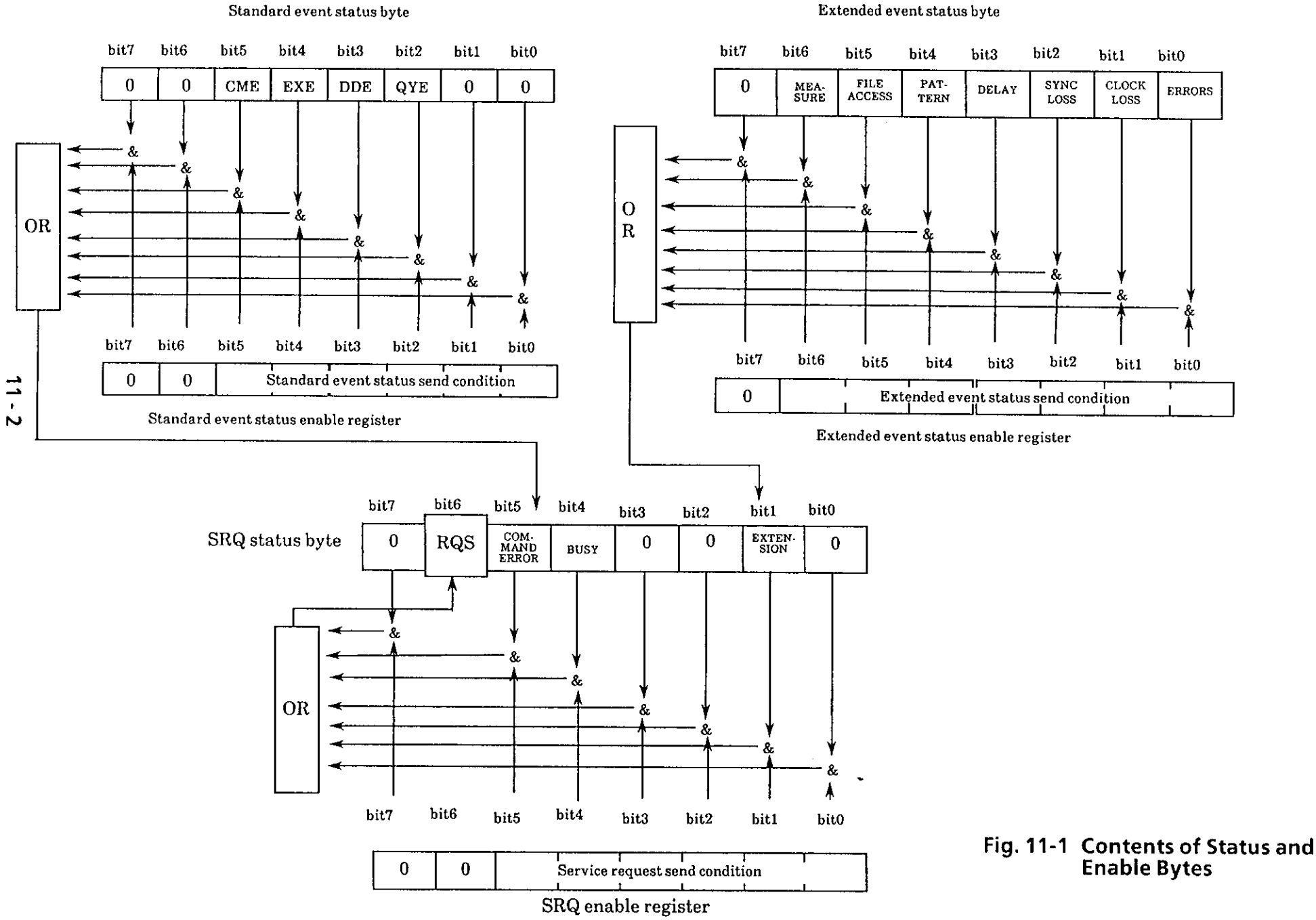


Fig. 11-1 Contents of Status and Enable Bytes

11.2 Description of Registers and Status Bytes

11.2.1 SRQ enable register

Operation

This register is used to set the conditions under which GP-IB command service requests are sent by setting individual RQA status bits ON and OFF.

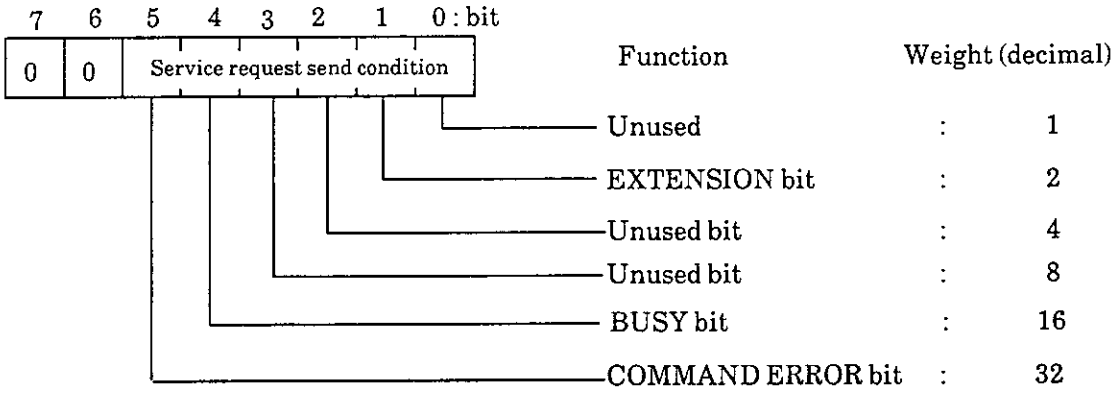
When an event for a bit which has been set to "1" in the SRQ enable register occurs, the RQS bit becomes "1" and an interrupt is sent to the controller.

- When an SRQ command is received, all the previously set send conditions are cleared.
- Multiple service request conditions can also be set.
- Immediately after the power is turned ON and immediately after a device clear or initialize command is received, all SRQ send conditions are cleared.

Command

Control message	:	SRQΔ0 to SRQΔ63
Data request message	:	SRQ?
Output message	:	SRQΔΔ0 (CR/LF) to SRQΔ63 (CR/LF)

Configuration



Function:

weight 0 : Clears the service request send condition.

1: Unused bit

2: Sends a service request when an event from the extended event status register occurs.

4: Unused bit

8: Unused bit

16: Sends a service request when there is data to be output to the GP-IB.

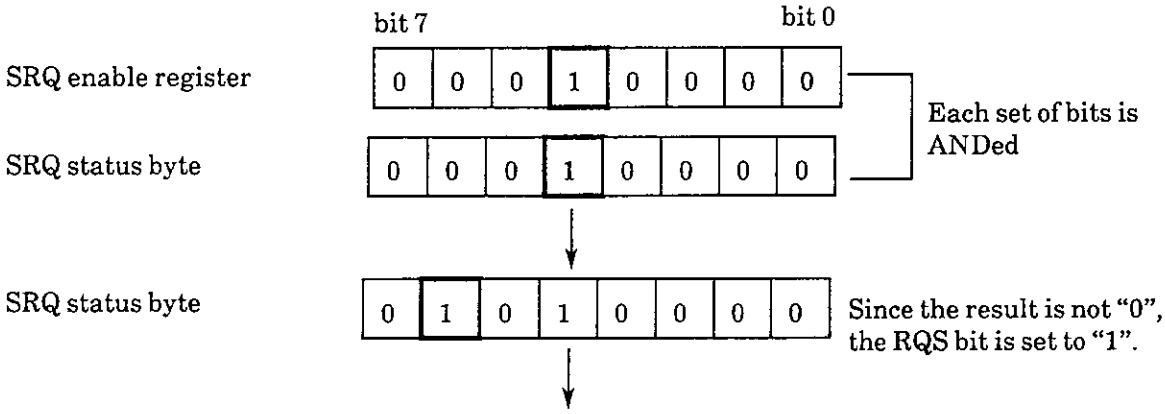
32: Sends a service request when an event from the standard event status register occurs.

11.2.2 SRQ status byte

Operation

The status byte output on the bus when serially polled by the controller.

When an event is generated, it is ANDed with the appropriate SRQ enable bit. If the result is not "0", the RQS bit (bit 6) is set to "1".



Command

Control message :

Data request message : STB?

Output message : STBΔ#B********(CR/LF)

(*: Represents "0" or "1" setting at that time)

Configuration

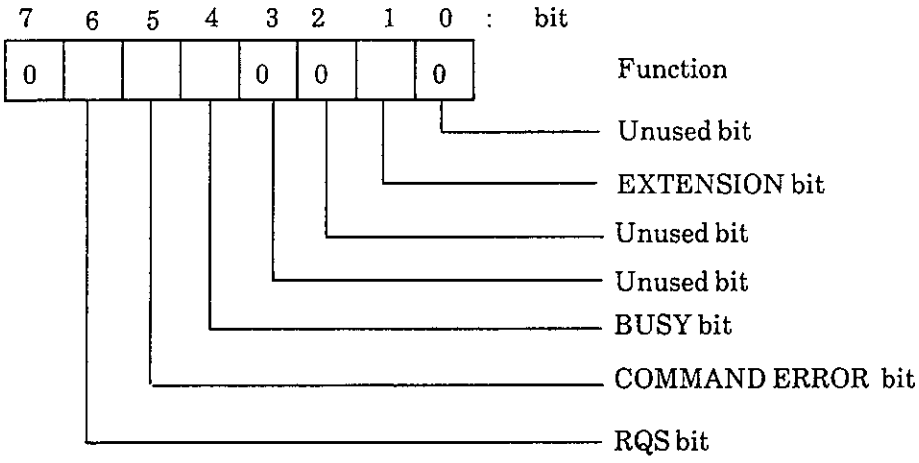


Table 11-1 GP-IB Status Byte

Name	Bit	Function
———	7	Not used.
RQS	6	This bit indicates that a service request is being issued. It is set when the bit causing the service request is set, and it is reset after the controller reads the status byte or after a device clear or initialize command has been received. It there is no SRQ event, it is not set.
COMMAND ERROR	5	This bit indicates that one of the errors represented by the standard event status byte and standard event status enable register has occurred. (See Table 11-2.). It is reset after the standard event status byte is read or when a device clear or initialize command is received. If this bit is set when the COMMAND ERROR bit of the SRQ enable register is set, an SRQ is generated.
BUSY	4	This bit is set when there are data to be sent on GP-IB. It is reset when data transfer is completed or when a device clear or initialize command is received. If this bit is set when the BUSY bit of the SRQ enable register is set, an SRQ is generated.
	3	Not used.
	2	Not used.
EXTENSION	1	This bit is set when the extended event status byte and extended event status enable register conditions are satisfied (See Table 11-3.). It is reset when the extended event status byte is read or when a device clear or initialize command is received. If this bit is set when the EXTENSION enable register is set, an SRQ is generated.
———	0	Not used.

11.2.3 Standard event status enable register

Operation

This register is used to enable/disable service requests by standard events.

The COMMAND ERROR bit of the SRQ status byte is set when standard events as represented by the bits of the standard event status byte occur.

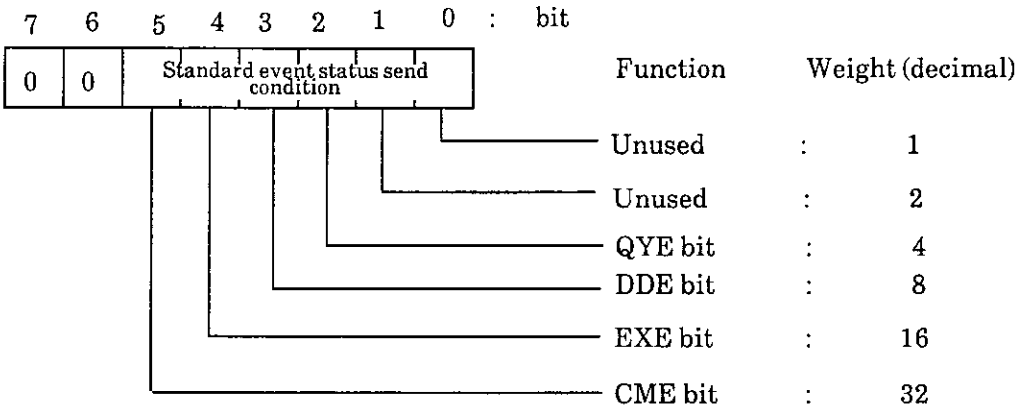
If a standard event occurs for a status register bit which is "1", the COMMON ERROR bit is set to "1".

- When an ESE command is received, all previously set send conditions are cleared.
- Multiple service request conditions can also be set.
- Immediately after the power is turned ON and immediately after a device clear or initialize command is received, all ESE send conditions are cleared.

Command

Control message	:	ESEΔ0 to ESEΔ63
Data request message	:	ESE?
Output message	:	ESEΔΔ0 (CR/LF) to ESEΔ63 (CR/LF)

Configuration



Function:

Weight 0 : Clears the standard event status byte.

1: Unused bit

2: Unused bit

4: Sends the standard event status send condition when a talker is designated event though the send buffer is empty or when the send buffer has overflowed.

8: Sends the standard event status send condition when a received command is illegally used.

16: Sends the standard event status send condition for numerical errors or when a HEX value is used illegally.

32: Sends the standard event status send condition when a header error occurs.

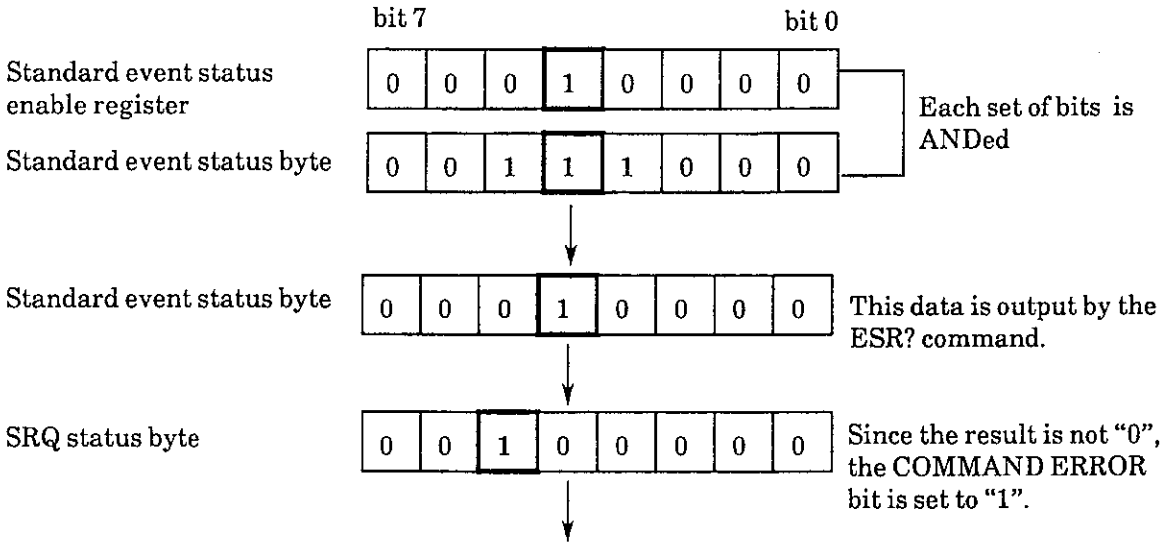
11.2.4 Standard event status byte

Operation

This byte contains information on the data to be output to the bus when a data request message is received.

When an event is generated, it is ANDed with the standard event status enable register. If the result is not "0", the COMMAND ERROR bit of the SRQ status byte is set to "1".

- After the status byte has been read by a data request message, each of its bits is cleared.
- Otherwise, this byte is not cleared even if a device clear, interface clear or initialize command is received.



This data is output during serial polling and when requested by a data request message.

Command

Control message :

Data request message : ESR?

Output message : ESRΔ#B********(CR/LF)

(*: Represents "0" or "1" setting at that time)

Configuration

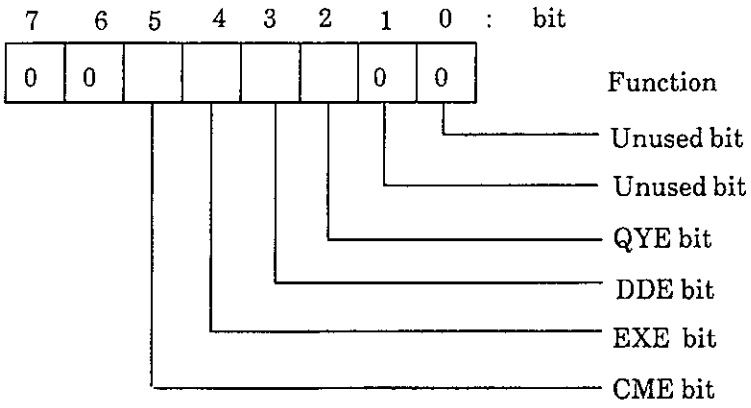


Table 11-2 Standard Event Status Byte (1/2)

Name	Bit	Function
	7	Not used
	6	Not used
CME	5	CME: Command error This bit is set when an undefined header is received. It is reset when the standard event status byte is read. Otherwise, it is not reset even if a device clear, interface clear, or initialize command is received. If this bit is set after the CME bit of the standard event status enable register has been set by the ESE command, the COMMAND ERROR bit of the status byte will be set.
EXE	4	EXE: Execution error This bit is set when the numerical value is out of range or when a HEX value is illegally used. It is reset when the standard event status byte is read. Otherwise, it is not reset even if a device clear, interface clear, or initialize command is received. If this bit is set after the EXE bit of the standard event status enable register has been set by the ESE command, the COMMAND ERROR bit of the status byte will be set.
DDE	3	DDE: Device dependent error This bit is set when a received command is illegally used. It is reset when the standard event status byte is read. Otherwise, it is not reset even if a device clear, interface clear, or initialize command is received. If this bit is set after the DDE bit of the standard event status enable register has been set by the ESE command, the COMMAND ERROR bit of the status byte will be set.
QYE	2	QYE: Query error This bit is set when the MP1702A/MP1756A/MP1609A/MP1651A is designated a talker even though the send buffer is empty or when the send buffer has overflowed. It is reset when the standard event status byte is read. Otherwise, it is not reset even if a device clear, interface clear, or initialize command is received. If this bit is set after the QYE bit of the standard event status enable register has been set by the ESE command, the COMMAND ERROR bit of the status byte will be set.

Table 11-2 Standard Event Status Byte (2/2)

Name	Bit	Function
	1	Not used
	0	Not used

11.2.5 Extended event status enable register

Operation

This register is used to enable/disable service requests by extended events.

The SRQ status byte EXTENSION bit is set to "1" or "0" when an extended event occurs.

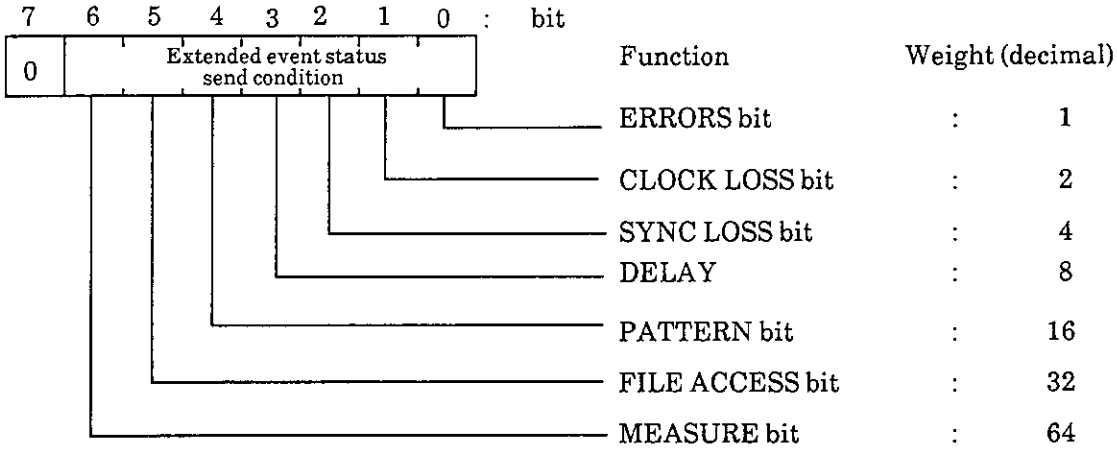
When an extended event whose enable status bit is set to "1" occurs, the EXTENSION bit is set to "1".

- When an EES command is received, all the previously set send conditions are cleared.
- Multiple service request conditions can also be set.
- All EES send conditions are cleared immediately after the power is turned ON and immediately after a device clear or initialize command is received.

Command

Command message	:	EESΔ0 to EESΔ127
Data request message	:	EES?
Output message	:	EESΔΔ0 (CR/LF) to EESΔ127 (CR/LF)

Configuration



Function

Weight 0: Clears the extended event status send (EESS) byte

1: Sends the EESS condition when an error is detected.

2: Sends the EESS condition at clock loss and clock recovery.

4: Sends the EESS condition at sync loss and sync recovery.

8: Sends the EESS condition when the servo subcircuit of the clock delay circuit changes from the BUSY to READY state.

16: Sends the EESS condition on completion of programmable pattern setting.

32: Sends the EESS condition on completion of a floppy disk access.

64: Sends the EESS condition when measurements have been completed.

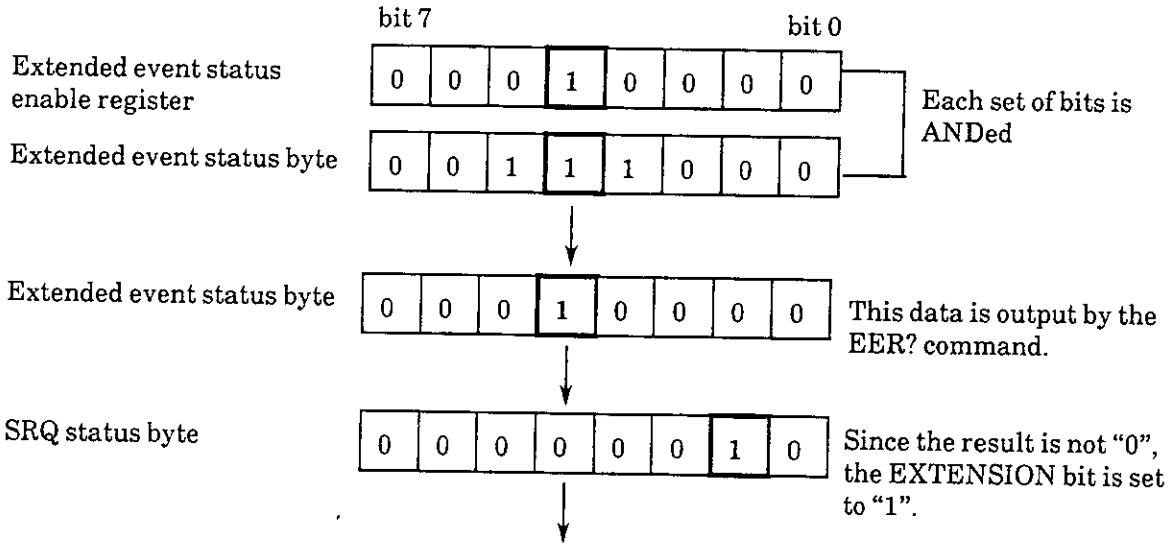
11.2.6 Extended event status byte

Operation

This byte contains information on the data to be output to the bus when a data request message is received.

When an event is generated, the status for that event is ANDed with the same bit in the extended event status enable register. If the result is not "0", the EXTENSION bit of the SRQ status byte is set to "1".

- After the status byte is read by a data request message, all status bytes are cleared.
- Otherwise, this byte is not cleared even if a device clear, interface clear, or initialize command is received.



This data is output during serial polling and in response to a data request message.

Command

Control message :

Data request message : EER?

Output message : EERΔ#B********(CR/LF)

(*: Represents "0" or "1" at that time)

Configuration

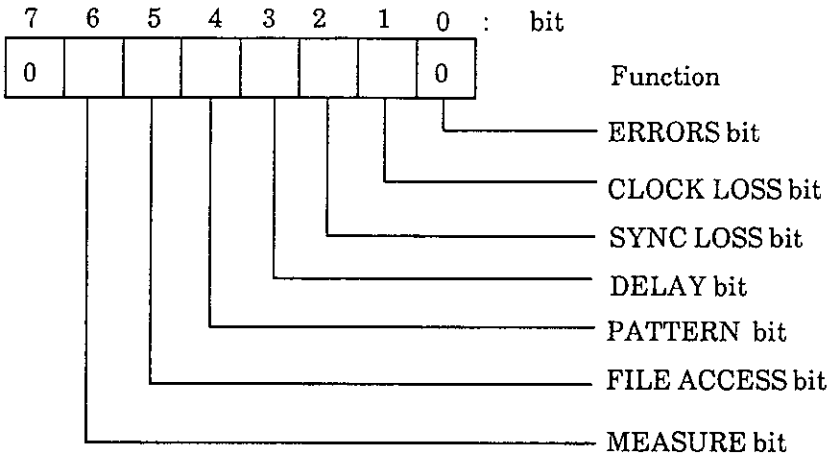


Table 11-3 Contents of Extended Event Status Byte (1/2)

Name	Bit	Function
	7	Unused
MEASURE	6	This bit is set when measurement is completed and reset when the extended event status byte is read. Otherwise, this bit is not reset, even when the device clear, interface clear, or initialize command is received. If this bit is set after the MEASURE bit of the extended event status enable register has been set by the EES command, the EXTENSION bit of the status byte will be set.
FILE ACCESS	5	This bit is set when access to the floppy disk is completed and reset when the extended event status byte is read. Otherwise, this bit is not reset even when the device clear, interface clear, or initialize command is received. If this bit is set after the FILE ACCESS bit of the extended event status enable register has been set by the EES command, the EXTENSION bit of the status byte will be set.
PATTERN	4	This bit is set upon completion of programmable pattern setting and reset when the extended event status byte is read. Otherwise, this bit is not reset even when the device clear, interface clear, or initialize command is received. If this bit is set after the PATTERN bit of the extended event status enable register has been set by the EES command, the EXTENSION bit of the status register will be set.
DELAY	3	This bit is set when the clock-input-phase servo circuit changes from the BUSY to the READY state and is reset when the extended event status byte is read. Otherwise, this bit is not reset even when the device clear, interface clear, or initialize command is received. If this bit is set after the DLEAY bit of the extended event status enable register has been set by the EES command, the EXTENSION bit of the status register will be set.

Table 11-3 Contents of Extended Event Status Byte (2/2)

Name	Bit	Function
SYNC LOSS	2	This bit is set when synchronization is lost or recovered and reset when the extended event status byte is read. Otherwise, this bit is not reset even when the device clear, interface clear, or initialize command is received. If this bit is set after the SYNC LOSS bit of the extended event status enable register has been set by the EES command, the EXTENSION bit of the status register will be set.
CLOCK LOSS	1	This bit is set when clock is lost or recovered and reset when the extended event status byte is read. Otherwise, this bit is not reset even when the device clear, interface clear, or initialize command is received. If this bit is set after the CLOCK LOSS bit of the extended event status enable register has been set by the EES command, the EXTENSION bit of the status byte will be set.
ERRORS	0	This bit is set the first time an error is detected and reset when the extended event status byte is read. Otherwise, this bit is not reset even when the device clear, interface clear, or initialize command is received. If this bit is set after the ERRORS bit of the extended event status enable register has been set by the EES command, the EXTENSION bit of the status byte will be set.

SECTION 12
PATTERN DATA TRANSFER BY DMA

As with the conventional instruments (e.g. MP1601A/MP1604A), programmable patterns of up to 512 k bits can be transferred by the BIT command via GP-IB; however, a large amount of time is required.

Therefore, the MP1702A/MP1756A/MP1609A/MP1651A has been equipped with a DMA transfer function to facilitate the rapid transfer of pattern data.

12.1 DMA

DMA is the abbreviation of Direct Memory Access and is a method of transferring a large volume of data at high speed (memory to memory transfer).

12.2 Commands for Number of Pattern Data Input Bytes and Number of Pattern Data Output Bytes

The number of pattern data input bytes (WRT command) and number of pattern data output bytes (RED command) GP-IB commands of the MP1702A/MP1756A/MP1609A/MP1651A are described below:

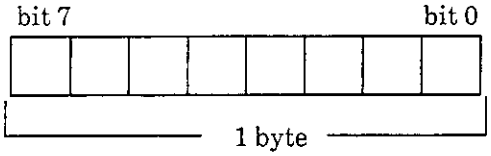
To transfer pattern data, the MP1702A/MP1756A/MP1609A/MP1651A must be given the following information by using the WRT and RED commands:

- Number of bytes of pattern data to be transferred
- The start address for the MP1702A/MP1756A/MP1609A/MP1651A internal RAM location at which received pattern data is to be stored, or the start address for the MP1702A/MP1756A/MP1609A/MP1651A internal RAM location at which the pattern data to be output are stored.

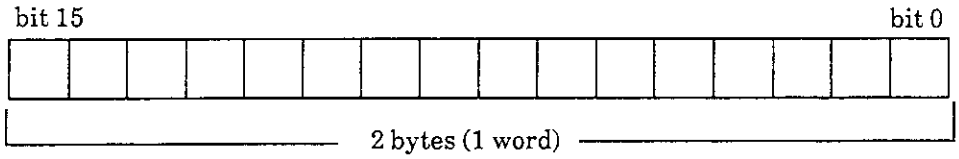
12.2.1 Number of bytes of pattern data to be transferred

A byte consists of 8-bits.

The MP1702A/MP1756A/MP1609A/MP1651A has a 16-bit (2 bytes or 1 word) pattern data format.

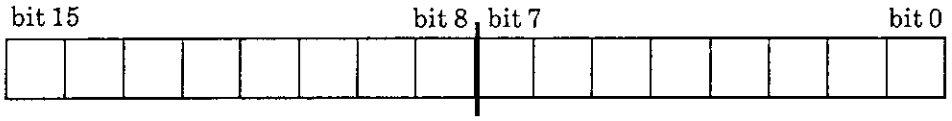


Value range: 0 to 255 (decimal)
0H to FFH (hexadecimal)



Value range: 0 to 65535 (decimal)
0H to FFFFH (hexadecimal)

The pattern data format is shown below:



16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 : BIT LED No.

The number of pattern data bytes to be transferred equals the number of 8-bit data blocks to be transferred.

2 bytes (data words) are always transferred for each page. Therefore, when only one byte exists, only the high-order byte of the data word (bits 15 to 8, in the figure above) will contain the data byte.

12.2.2 Start address of the MP1702A/MP1756A/MP1609A/MP1651A internal RAM to store transferred pattern data and that to output pattern data to be transferred

The MP1702A/MP1756A/MP1609A/MP1651A internal RAM address ranges from 0 to 32767. This address range is common to both WORD and DATA patterns.

The relationship between the displayed page number and the pattern address is shown below:

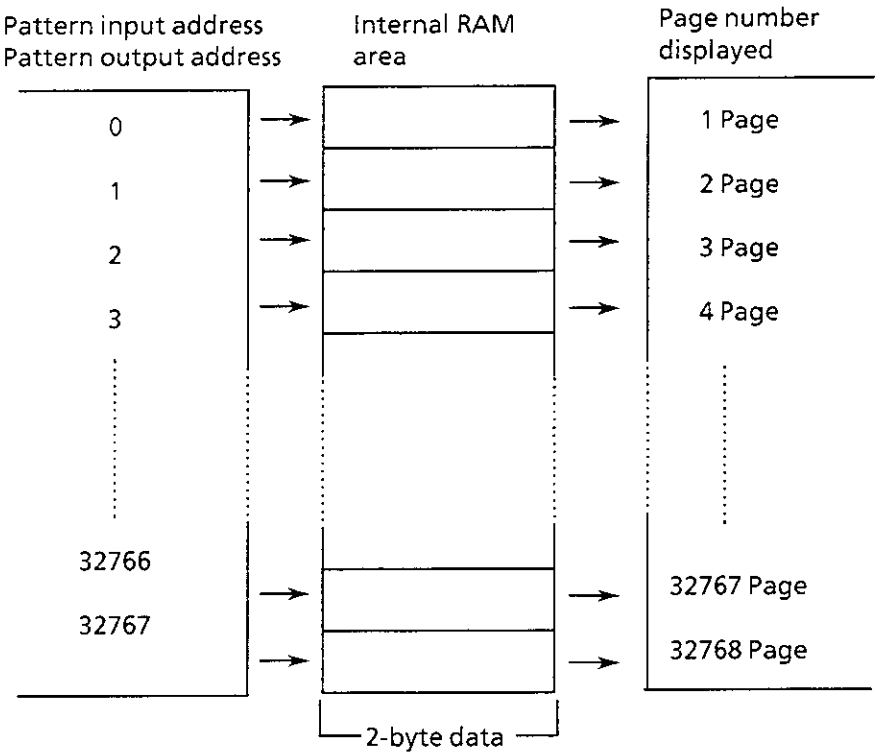


Fig. 12-1 Relationship between Page Number and Pattern Address

12.3 DMA Program Examples

Sample programs that use the DMA function are listed in items (15) and (16) of paragraph 15.4 and 15.5. The functions and formats of each DMA command are illustrated via these programs.

Since odd-numbered addresses cannot be defined on a 16-bit CPU, the following equation defines the relationship between the pattern input/output start address and the page number.

(Pattern input/output start address + 1) = actual page number

SECTION 13
MEASUREMENT RESULT OUTPUT FORMAT

This section describes the measurement result output format when measurement results are output after each measurement is completed.

13.1 Types of Output Data

The table below shows output items and conditions.

Table 13-1 Measurement Result Output Items and Conditions (1/2)

Types of output data		Output items	Time of output
Item data	Measurement data at start	· Data and time at start of measurement	· When measurement is started (For repeat measurements, this data is output only at time measurements are started)
	One-second data *	· Measurement time · Average error ratio for one second · Error count for one second	· When one-second data print select function is ON and the average one-second error ratio exceeds the one-second data print threshold
	Alarm data	· Time of event occurrence and recovery; Event log	· When clock is lost · When clock loss is recovered · When synchronization is lost · When sync loss is recovered (However, this data is not output if sync is lost or recovered while measurement is not in progress.)

* Output thresholds for one-second data are NO ERROR, >1E-6, >1E-4, or 1E-3.

Table 13-1 Measurement Result Output Items and Conditions (2/2)

Types of output data		Output items	Time of output
Block data	Measurement data at end	Varies depending on the data print format. For details, see the next page.	<Repeat measurement> · When each measurement is completed or when aborted. <Single measurement> · When measurement is completed or aborted. <Untimed measurement> · When measurement is stopped
	Intermediate data	Same as measurement data at end. (In this case, however, the most recent measurement time is output in place of the end of measurement time.)	· When intermediate result are requested by the (IMD?) data request command

The table below shows output items for the measurement data at end.

Table 13-2 Output Items of Measurement Data at End (1/2)

Output item \ Data print format		Standard	Short (Abridged)
Measurement start time		○	–
Measurement end time (Note 1)		○	○
Elapsed time		○	–
Error measurement data	Average error ratio	○	○
	Total error count	○	○
Interval measurement data	No. of error intervals	○	–
	Error-free intervals ratio	○	–
	No. of power failure intervals	○	–
	No. of clock loss intervals	○	–
	No. of sync loss intervals	○	–
Threshold interval measurement data (Note 2)	No. of threshold EI (1E−3, 1E−4, 1E−5,1E−6, 1E−7, 1E−8, ≦1E−8)	○	○
	Threshold EFI ratio (1E−3, 1E−4, 1E−5,1E−6, 1E−7, 1E−8, ≦1E−8)	○	○

Note 1: The most recent measurement time is output for intermediate data.

Note 2: These data are output when the threshold EI/EFI measurement data are selected for printing.

Table 13-2 Output Items of Measurement Data at End (2/2)

Output item	Data print format	Standard	Short (Abridged)
Error performance data (Note 3)	Errored seconds ratio	○	○
	Error-free seconds ratio	○	○
	Severely errored seconds ratio	○	○
	Degraded minutes ratio	○	○
	Unavailable seconds ratio	○	○

Note 3: These data are output when error performance measurement data is ON and the measurement time is 60 seconds or more.

13.2 Output Format

13.2.1 Item data

(1) Output format (△ denotes a space.)

. Start of measurement STA△ **, **, **, **, **, **,
Year Month Day Hour Minute Second

. One-second data OSD△**.****E-**,**********
Average 1-second error ratio; One-second error count

. Clock loss CL△△ **, **, **, **, **, **,
Year Month Day Hour Minute Second

. Clock recovery CR△△ **, **, **, **, **, **,
Year Month Day Hour Minute Second

. Sync loss SL△△ **, **, **, **, **, **,
Year Month Day Hour Minute Second

. Sync recovery SR△△ **, **, **, **, **, **,
Year Month Day Hour Minute Second

(2) Terminator

SR2 (CR.LF) + SR3 (EOI)

13.2.2 Block data

(1) Output format (△ denotes a space.)

(a) When the data is to be printed in the standard format

. 1st line : Measurement start time

STA△ **, **, **, **, **, **,

Year Month Day Hour Minute Second

. 2nd line : Measurement end time for measurement data at end

END△ **, **, **, **, **, **,

Year Month Day Hour Minute Second

: Most recent measurement time for intermediate data

INT△ **, **, **, **, **, **,

Year Month Day Hour Minute Second

. 3rd line : Elapsed time

ELP△*****, **, **, **

Day Hour Minute Second

. 4th line : Average error ratio

ER△△*.****E − **

. 5th line : Total error count (less than IE + 10)

EC△△**********

: Total error count (IE + 10 or more)

EC△△*.****E + **

. 6th line : Error intervals

EI△△**********

. 7th line : Error-free intervals

EFI△△△***.****

. 8th line : No. of power failure intervals

PFI△**********

. 9th line : No. of clock loss intervals

CLI△**********

. 10th line: No. of sync loss intervals

SLI△**********

When the threshold EI/EFI data print function is ON, the following lines are output after the 10th line:

- 11th line: No. of threshold EI, threshold EFI ratio (>E-3)

  TH3Δ**********,***.****

- 12th line: No. of threshold EI, threshold EFI ratio (>E-4)

  TH4Δ**********,***.****

- 13th line: No. of threshold EI, threshold EFI ratio (>E-5)

  TH5Δ**********,***.****

- 14th line: No. of threshold EI, threshold EFI ratio (>E-6)

  TH6Δ**********,***.****

- 15th line: No. of threshold EI, threshold EFI ratio (>E-7)

  TH7Δ**********,***.****

- 16th line: No. of threshold EI, threshold EFI ratio (>E-8)

  TH8Δ**********,***.****

- 17th line: No. of threshold EI, threshold EFI ratio (≦E-8)

  TH9Δ**********,***.****

When the error performance data print function is ON, the following lines are output after the 10th line:

- 11th line: Errored seconds ratio

  ESΔΔΔΔ***.****

- 12th line: Error-free seconds ratio

  EFSΔΔΔ***.****

- 13th line: Severely errored seconds ratio

  SESΔΔΔ***.****

- 14th line: Degraded minutes ratio

  DMΔΔΔΔ***.****

- 15th line: Unavailable seconds ratio

  USΔΔΔΔ***.****

When both the threshold interval measurement and error performance data print functions are ON, threshold interval measurement data are output first.

(b) When data is to be printed in the short (abridged) format

. 1st line : Measurement end time for measurement data at end

ENDΔ **, **, **, **, **, **,

Year Month Day Hour Minute Second

: Most recent measurement time for intermediate data

INTΔ **, **, **, **, **, **,

Year Month Day Hour Minute Second

. 2nd line : Average error ratio

ER$\Delta\Delta$*. ****E – **

. 3rd line : Total error count (less than 1E + 10)

EC$\Delta\Delta$**********

: Total error count (IE + 10 or more)

EC$\Delta\Delta$*. ****E + **

Both the threshold interval measurement and error performance data output formats are output after the third line as in the standard format.

(2) Terminator

For other than the last line : SR2 (CR.LF)

For the last line : SR2 (CR.LF) + SR3 (EOI)

13.2.3 Output examples

(1) Basic output example for standard format

```
1st line  :  STA  90,  03,  18,  20,  05,  30
1st line  :  STA  90,  03,  18,  20,  05,  30
2nd line  :  END  90,  03,  18,  20,  05,  31
3rd line  :  ELP        0,  00,  00,  01
4th line  :  ER   0.    0000E – 10
5th line  :  EC                 0
6th line  :  EI                 0
7th line  :  EFI          100.  0000
8th line  :  PFI                0
9th line  :  CLI                0
10th line :  SLI                0
```

Data at start of measurement

Data at end of measurement

(2) Extended output example for standard format (including threshold and error performance measurement output)

```
1st line  :  STA  90,  03,  18,  20,  05,  30
1st line  :  STA  90,  03,  18,  20,  05,  30
2nd line  :  END  90,  03,  18,  20,  51,  30
3rd line  :  ELP        0,  00,  01,  00
4th line  :  ER   0.    0000E – 10
5th line  :  EC                 0
6th line  :  EI                 0
7th line  :  EFI          100.  0000
8th line  :  PFI                0
9th line  :  CLI                0
10th line :  SLI                0
11th line :  TH3                0,    100. 0000
12th line :  TH4                0,    100. 0000
13th line :  TH5                0,    100. 0000
14th line :  TH6                0,    100. 0000
15th line :  TH7                0,    100. 0000
```

Data at start of measurement

Data at end of measurement

```
16th line  :  TH8              0. 100. 0000
17th line  :  TH9         0. 0000
18th line  :  ES        100. 0000
19th line  :  EFS         0. 0000
20th line  :  SES         0. 0000
21st line  :  DM          0. 0000
22nd line  :  US          0. 0000
```

Data at end of measurement

(3) Basic output example for short (abridged) format

```
1st line  :  STA  90,  03,  18,  20,  53,  10
1st line  :  END  90,  03,  18,  20,  53,  30
2nd line  :  ER   0. 000E – 09
3rd line  :  EC             0
```

Data at start of measurement

Data at end of measurement

(4) Data output example for standard format

```
1st line  :  STA  90,  03,  18,  20,  54,  00
2nd line  :  OSD   1.  0000E – 03,      2999996
3rd line  :  OSD   1.  0000E – 03,      2999996
4th line  :  OSD   1.  0000E – 03,      2999996
5th line  :  OSD   1.  0000E – 03,      2999996
6th line  :  CL   90,  03,  18,  20,  54,  05
7th line  :  CR   90,  03,  18,  20,  54,  51
8th line  :  SL   90,  03,  18,  20,  54,  52
9th line  :  SR   90,  03,  18,  21,  00,  00
10th line  :  OSD   1.  0000E – 03,      2999996
11th line  :  OSD   1.  0000E – 03,      2999996
```

Data at start of measurement

One-second data

Alarm data

One-second data

```
1st line   :   STA  90,  03,  18,  20,  54,  00
2nd line   :   END  90,  03,  18,  21,  00,  02
3rd line   :   ELP            0,  06,  02
4th line   :   ER   1.  0000E – 03
5th line   :   EC         14999980
6th line   :   EI                5
7th line   :   EFI           0. 000
8th line   :   PFI               0
9th line   :   CLI              46
10th line  :   SLI              38
```

Data at end of measurement

(5) Basic output example for standard format (for "no data" display)

```
1st line   :   STA  90,  03,  18,  20,  05,  30
1st line :     STA  90,  03,  18,  20,  05,  30
2nd line   :   END  90,  03,  18,  20,  05,  33
3rd line   :   ELP            0,  00,  00,  03
4th line   :   ER   0. 0000E – 00
5th line   :   EC                0
6th line   :   EI                0
7th line   :   EFI          0.  0000
8th line   :   PFI               0
9th line   :   CLI               1
10th line  :   SLI               2
```

Data at start of measurement

Data at end of measurement

13.3 Data Buffering

Data which are not read by the controller via GP-IB, are stored in a buffer.

When there is space in the buffer, data are stored in the order in which output requests are generated. When recalled, they are output chronologically beginning with old data. When the bufffer runs out of space, the oldest data with the lowest priority are overwritten.

Data priorities are as follows:

- Block data

  Priority
  1. Measurement data at end : ↑ Higher
  2. Intermediate data : ↓ Lower

- Item data

  Priority
  1. Measurement data at start : ↑ Highest
  2. Alarm data |
  3. One-second data : ↓ Lowest

SECTION 14
EXECUTING PROGRAMS WRITTEN FOR OTHER DEVICES

The following four items must be editted when executing programs written for other devices (MP1601A/MP1604A) on the MP1702A/MP1756A/MP1609A/MP1651A.

1. Leave at least one space after the header (HR) field.

   The header field and numeric data field (or identifier) are distinguished by this space.

   Example:

   OUTPUT 700;"DTM0"
   ↓
   OUTPUT 700;"DTM△0"
   ↑
   └——— Space inserted here.

2. Use the semicolon (;) separator to separate commands and the comma (,) separate to separate data items.

   Examples:

   - Command separation

     OUTPUT 700;"DTM△0,PTN△0"
     ↓
     OUTPUT 700;"DTM△0;PTN△0"
     ↑
     └——— Use semicolon(;) here.

   - Data separation

     OUTPUT 700;"BIT△10;20"

     OUTPUT 700;"BIT△10,20"
     ↑
     └——— Use comma(,) here.

3. When sequential data output requests are received by the MP1702A/MP1756A/MP1609A/MP1651A, only information from the last data request is output, as shown below:

Examples:

- When one of program code contains multiple data output requests

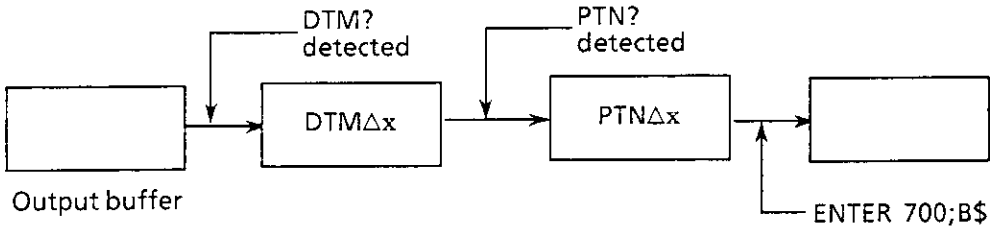
  OUTPUT 700;"DTM?;PTN?"

  ENTER 700;B$

  PRINT B$ ← Outputs PTN△x.

  (x: Status at time of output)

  When the above program is executed, the following output buffer operations occur.



  The correct data output request program code is as follows:

  OUTPUT 700; "DTM?"

  ENTER 700;B$

  PRINT B$ ← Outputs DTM△x.

  !

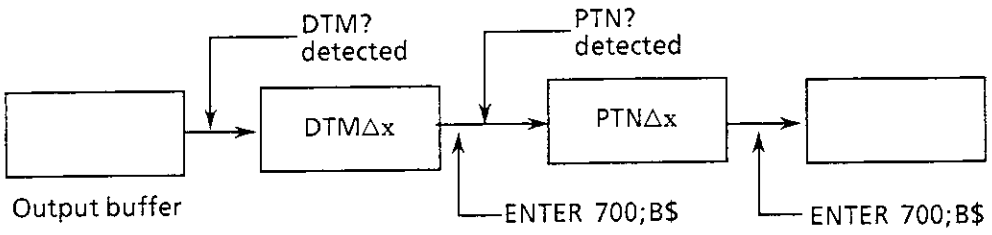
  OUTPUT 700; "PTN?"

  ENTER 700; B$

  PRINT B$ ← Outputs PTN△x.

  (x: Status at time of output)

When this program is exeucted, the following output buffer operations occur.



- When multiple data request messages were sent

  OUTPUT 700;"DTM"

  OUTPUT 700;"PTN?"

  ENTER 700;B$

  PRINT B$ ← Outputs PTNΔx.

  (x: Status at time of output)

  ↓

  Above messages should be rewritten as follows:

  OUTPUT 700;"DTM?"

  ENTER 700;B$

  PRINT B$ ← Outputs DTMΔx.

  !

  OUTPUT 700; "PTN?"

  ENTER 700; B$

  PRINT B$ ← Outputs PTNΔx.

  (x: Status at time of output)

4. The GP-IB status byte format for the MP1702A/MP1756A/MP1609A/MP1651A includes all conventional as well as some special functions.

   The MP1702A/MP1756A/MP1609A/MP1651A GP-IB status byte is described in SECTION 11. A sample program is shown in SECTION 15.

SECTION 15
SAMPLE PROGRAMS

This section provides sample programs and flow charts for each function for reference when controlling the MP1702A/MP1756A/MP1609A/MP1651A via GP-IB.

Note: There is no description about the MP1756A/MP1755A in the sample programs in this section, but the GP-IB commands for the MP1756A/MP1755A can be used commonly with the MP1702A/MP1701B. However, when using the MP1755A, internal clock frequency setting is ineffective.

15.1 List of Sample Programs

Table 15-1 List of Sample Programs (1/2)

Item No. of Paragraphs 15.4 and 15.5	Description of Sample Program Functions
(1)	Sets input signal.
(2)	Sets automatic phase threshold search (auto search).
(3)	Sets pattern.
(4)	Measurement (Measurement type → Measurement results are output each time a measurement is completed) These programs are listed according to the measurement-end judgement and the data print format.
(5)	Measurement (Measurement type → Measurement results are output each time a measurement is completed) Intermediate measurement results are also output. These programs are listed according to the data print format.
(6)	Measurement (Measurement type → Measurement results are output when a data request command is issued.) (6) - (a) Outputs of error ratio measurement results (6) - (b) Outputs of measurement results
(7)	Sets internal timer.
(8)	Reads file information from a floppy disk. Check for access to floppy disk → Serial polling
(9)	Reads file information from a floppy disk. Check for access to floppy disk → Request command: STB?

Table 15-1 List of Sample Programs (2/2)

Item No. of Paragraphs 14.4 and 14.5	Sample Program Control Function
(10)	Serves, resaves, and recalls data.
(11)	Sets mask route length and the channel number.
(12)	Sets the mark ratio AND bit shifts.
(13)	Checks the standard event status byte. Check the COMMAND ERROR bit → Serial polling
(14)	Check the standard event status byte. Check the COMMAND ERROR bit → Request command: STB?
(15)	Transfer pattern data by DMA * (15) - (a) Pattern data output → Request command: BIT? (15) - (b) Pattern data output → DMA transfer
(16)	Transfer pattern data by DMA *

* The programs for transfer of pattern data by DMA are provided for the HP9000 only.

15.2 Controllers Used

The sample programs which appear in this section were written for the HP9000 series computer (paragraph 15.4) and for a PC-compatible computer with GP-IB interface card of National Instruments (paragraph 15.5).

The programs for the HP-9000 were written in HP-BASIC while those for the IBM PC-compatible were written in Microsoft QUICK-BASIC Version 3.00.

The programs were verified by running them on the HP9000-200/300 using HP-BASIC Version 5.12 and a COMPAQ computer with National Instruments GP-IB interface card using QUICK-BASIC Version 3.00.

15.3 Preparation for Program Execution

Table 15-2 shows the preparations that must be made for each controller prior to sample program execution.

Table 15-2 Preparations before Executing Sample Programs (1/2)

Controller	Preparations
HP9000	• Set the MP1702A/MP1756A/MP1609A/MP1651A's GP-IB Address to '1'. • Set the MP1701B/MP1755A/MP1608A/MP1650A's GP-IB address to '0'. • Connect the MP1702A/MP1756A/MP1609A/MP1651A, MP1701B/MP1755A/MP1608A/MP1650A, and HP9000 with GP-IB cables.
COMPAQ	• Set the MP1702A/MP1756A/MP1609A/MP1651A's GP-IB address to '1'. • Set the MP1701B/MP1755A/MP1608A/MP1650A's GP-IB address to '0'. • Set IBCONF as follows: ① < Board Characteristics > Board: GPIB0 (Defines board as "GPIB0") Primary GPIB Address 10 Secondary GPIB Address NONE Timeout setting T30S EOS byte 0AH Terminal Read on EOS yes Set EOI with EOS on Write yes Type of compare on EOS 7-bit Set EOI w/last byte of Write yes GPIB-PC Model PC2A Board is System Controller yes Local Lockout on all devices no Disable Auto Serial Polling yes High-speed timing no Interrupt jumper setting 7 Base I/O Address 02E1H DMA channel NONE Internal clock Freq (in MHZ) 6

Table 15-2 Preparations before Executing Sample Programs (2/2)

Controller	Preparations
COMPAQ (Cont.)	② < Device Characteristics > Device: ED (Defines device name as "ED") Primary GPIB Address: 1 Secondary GPIB Address: NONE Timeout setting: T30S EOS byte: 0AH Terminal Read on EOS: yes Set EOI with EOS on Write: yes Type of compare on EOS: 7-bit Set EOI w/last byte of Write: yes ③ <Device characteristics> Device: PPG (Defines device name as "PPG") Primary GPIB address: 0 Secondary GPIB Address: NONE Timeout setting: T30S EOS byte: 0AH Terminal Read on EOS: yes Set EOI with EOS on Write: yes Type of compare on EOS: 7-bit Set EOI w/last byte of Write: yes ④ Device Map for GPIB0 Connect the device defined as "ED" in ② and the device defined as "PPG" in ③ to Board: GPIB0 in ①. • Connect the MP1702A/MP1756A/MP1609A/MP1651A, MP1701B/ MP1755A/MP1608A/MP1650A, and COMPAQ with GP-IB cables.

The sample programs in paragraph 15.4 were written for systems using the HP9000 as the controller. The sample programs in paragraph 15.5 were written for systems using a COMPAQ Computer as the controller.

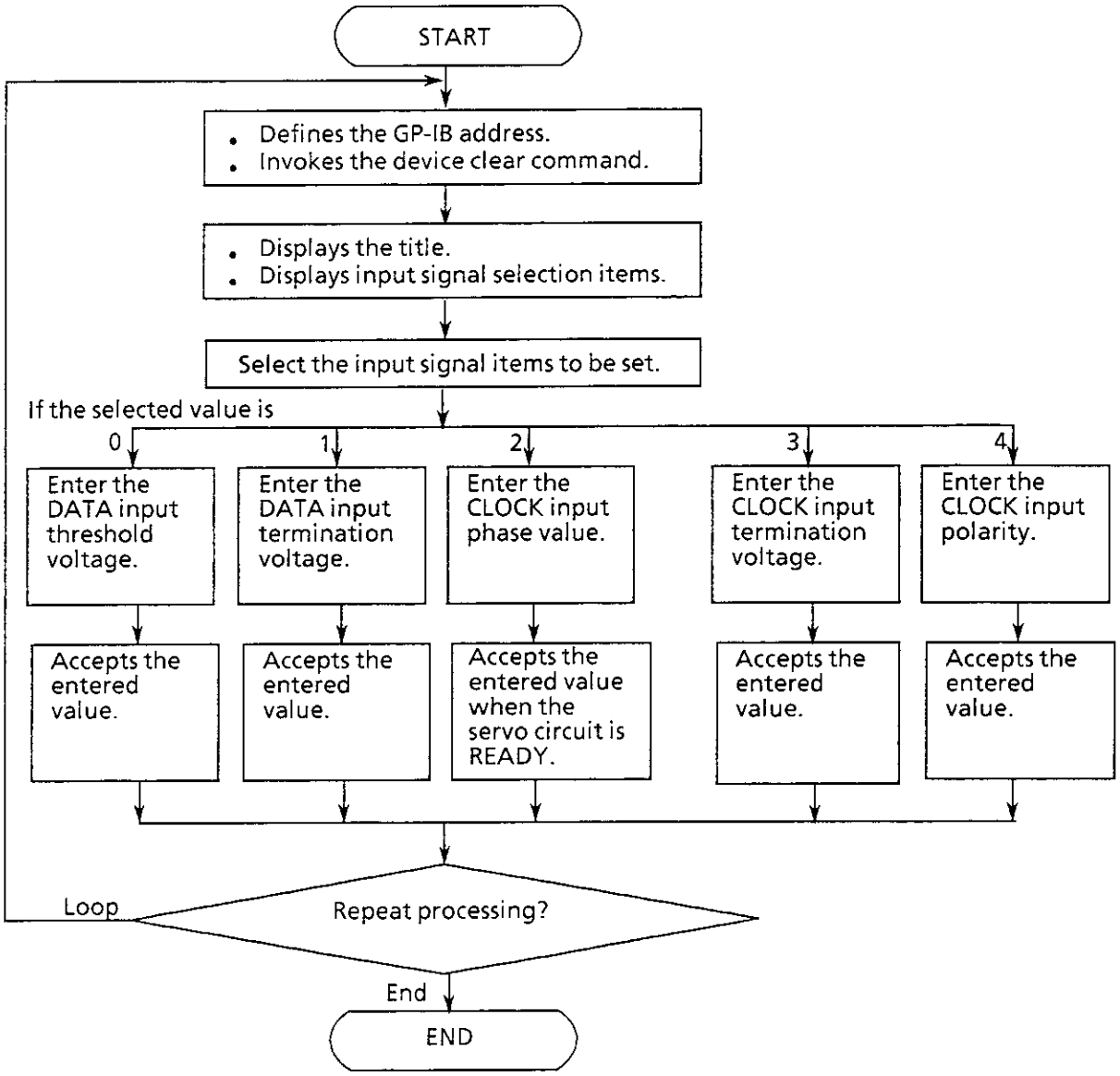
15.4 Sample Programs for the HP9000 as Controller

(1) Setting the input signal

This program controls input signal conventions and characteristics.

Values of various MP1702A/MP1756A/MP1609A/MP1651A signals are input after the appropriate selection has been made. (DATA and CLOCK voltages, phases, polarities, etc.).

Note that when setting the clock input phase, the program enters a delay state to ensure that the instrument is READY before applying the delay value.



Program listing

```
10    !*****************************************************************
20    !*                                                               *
30    !*   MP1702A/MP1609A/MP1651A  INPUT SIGNAL  SAMPLE SOFT          *
40    !*                                                    ED_IN      *
50    !*****************************************************************
60    !
70    LET Add=701
80    CLEAR Add
90     !
100   LOOP
110     CLEAR SCREEN
120     !
130     PRINT "** MP1702A/MP1609A/MP1651A  INPUT SIGNAL  SAMPLE SOFT **"
140     PRINT
150     PRINT "INPUT SIGNAL  * DATA THRESHOLD = [0] "
160     PRINT "              * DATA TERMINATION = [1] "
170     PRINT "              * CLOCK PHASE ADJUST= [2] "
180     PRINT "              * CLOCK TERMINATION = [3] "
190     PRINT "              * CLOCK POLARITY = [4] "
200     PRINT
210      !
220     INPUT "SELECT INPUT SIGNAL [ 0 OR 1 OR 2 OR 3 OR 4 ] ?",Inp$
230      !
240     SELECT Inp$
250      !
260     CASE "0"
270       INPUT "DATA THRESHOLD [ -3.000 ~ +1.750 V ] STEP 0.002 V ?",Dth$
280       OUTPUT Add;"DTH "&Dth$
290        !
300     CASE "1"
310       INPUT "DATA TERMINATION  [ GND=0 , -2V=1 ] ?",Dtm$
320       OUTPUT Add;"DTM "&Dtm$
330        !
340     CASE "2"
350       PRINT "     <<<    CLOCK PHASE ADJUST    >>>"
360       PRINT "MP1702A/MP1609A -->  -500 ~  500  1ps  STEP "
370       PRINT "MP1651A -->         -1000 ~ 1000  2ps  STEP"
380       INPUT "CLOCK PHASE ADJUST ?",Cpa$
390        !
400       LOOP
410         OUTPUT Add;"DLY?"     ! REQUEST Delay unlock ?
420         ENTER Add;Dly$
430       EXIT IF Dly$="DLY 0"
440       END LOOP
450        !
460       OUTPUT Add;"CPA "&Cpa$
470        !
480     CASE "3"
490       INPUT "CLOCK TERMINATION  [ GND=0 , -2V=1 ] ?",Ctm$
500       OUTPUT Add;"CTM "&Ctm$
510        !
520     CASE "4"
530       INPUT "CLOCK POLARITY  [ CLK=0 , N CLK=1 ] ?",Cpl$
540       OUTPUT Add;"CPL "&Cpl$
550        !
560     END SELECT
570      !
580     INPUT " NEXT DATA SET [ YES=0 , NO=1 ] ",Loop$
590      !
600   EXIT IF Loop$="1"
610   END LOOP
620    !
630   END
```

(2) Enabling automatic search for input thresholds (auto search)

This program executes "auto search" after the MP1702A/MP1756A/MP1609A/MP1651A has been connected to the MP1701B/MP1755A/MP1608A/MP1650A.

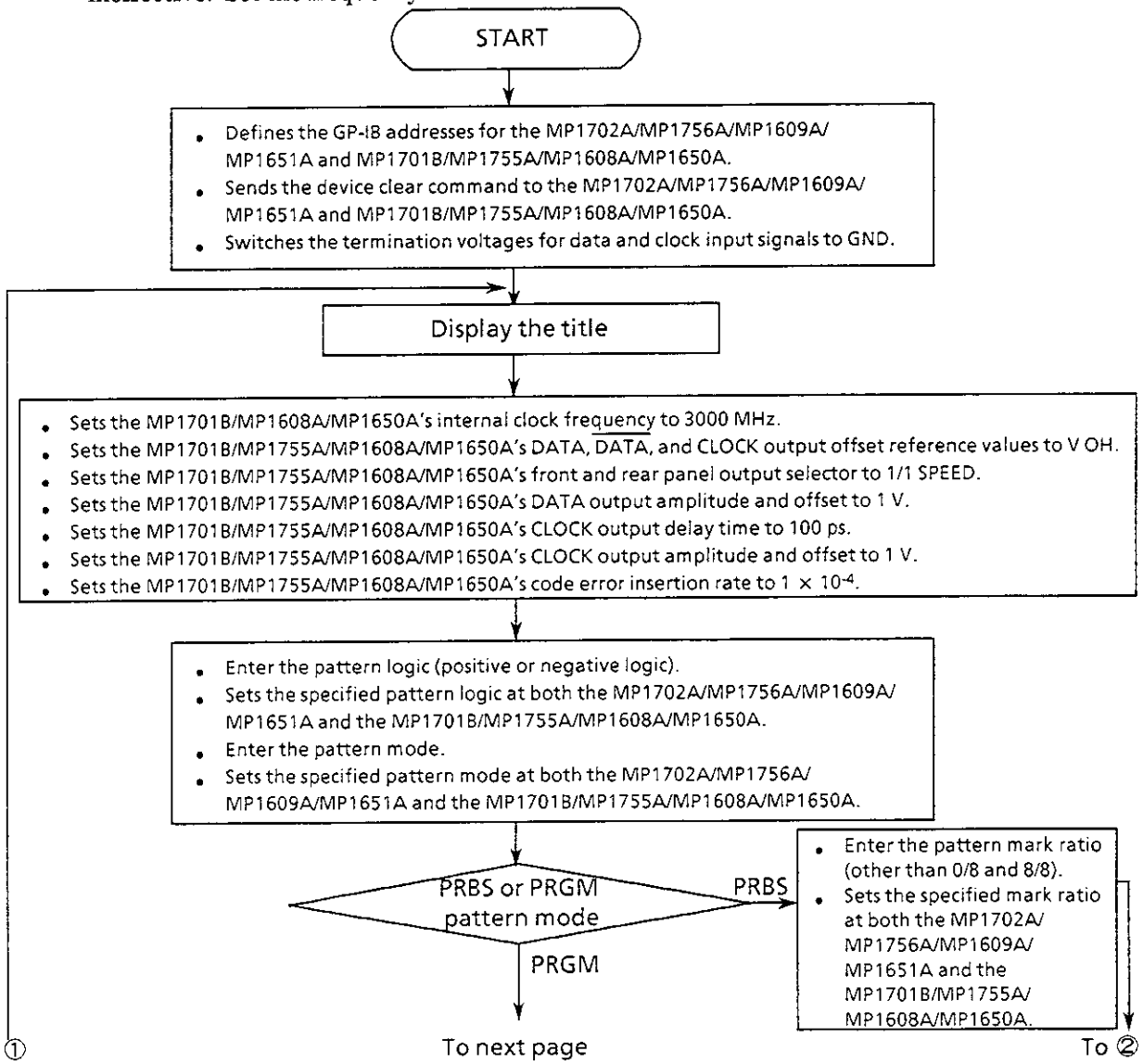
First, the program sets the conditions necessary to execute auto search on the MP1702A/MP1756A/ MP1609A/ MP1651A and MP1701B/MP1755A/MP1608A/MP1650A.

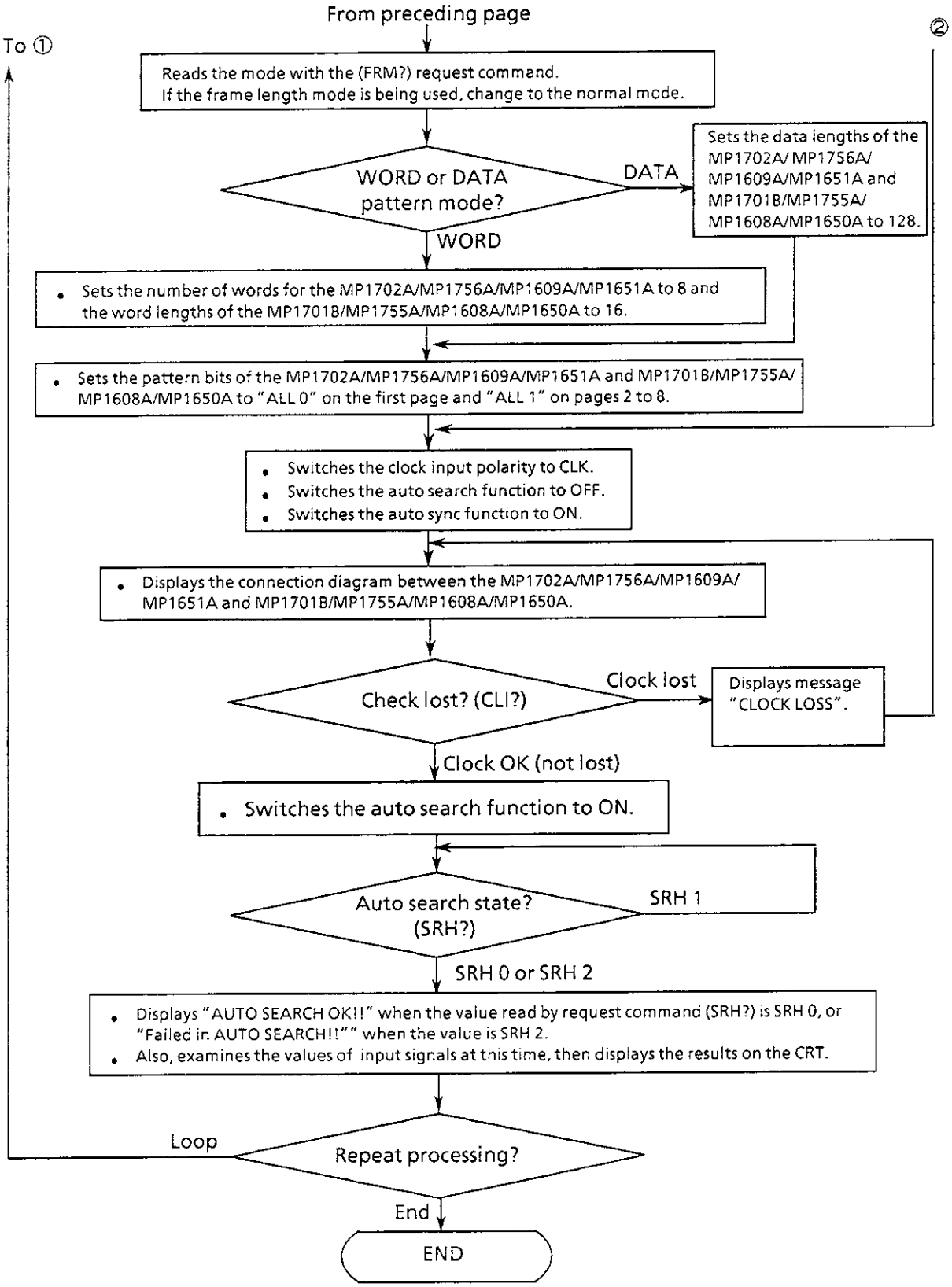
Next, it confirms that the clock is not lost (since auto search will not work if the clock is lost) and then turns ON the AUTO SEARCH function.

Afterwards, it reads the status of the AUTO SEARCH with the SRH? request command and displays this status on the CRT screen.

Finally, it reads the values of input signal and displays results on the CRT screen.

Note: When using the MP1755A for the transmitter, internal clock frequency setting is ineffective. Set the frequency of the external clock source to 3000 MHz in advance.




From preceding page
To ①
②
Reads the mode with the (FRM?) request command.
If the frame length mode is being used, change to the normal mode.
WORD or DATA pattern mode?
DATA
Sets the data lengths of the MP1702A/ MP1756A/ MP1609A/MP1651A and MP1701B/MP1755A/ MP1608A/MP1650A to 128.
WORD
• Sets the number of words for the MP1702A/MP1756A/MP1609A/MP1651A to 8 and the word lengths of the MP1701B/MP1755A/MP1608A/MP1650A to 16.
• Sets the pattern bits of the MP1702A/MP1756A/MP1609A/MP1651A and MP1701B/MP1755A/ MP1608A/MP1650A to "ALL 0" on the first page and "ALL 1" on pages 2 to 8.
• Switches the clock input polarity to CLK.
• Switches the auto search function to OFF.
• Switches the auto sync function to ON.
• Displays the connection diagram between the MP1702A/MP1756A/MP1609A/ MP1651A and MP1701B/MP1755A/MP1608A/MP1650A.
Check lost? (CLI?)
Clock lost
Displays message "CLOCK LOSS".
Clock OK (not lost)
• Switches the auto search function to ON.
Auto search state? (SRH?)
SRH 1
SRH 0 or SRH 2
• Displays "AUTO SEARCH OK!!" when the value read by request command (SRH?) is SRH 0, or "Failed in AUTO SEARCH!!"" when the value is SRH 2.
• Also, examines the values of input signals at this time, then displays the results on the CRT.
Loop
Repeat processing?
End
END

Program listing

```
10    !*************************************************************
20    !*                                                           *
30    !*   MP1702A/MP1609A/MP1651A  AUTO SEARCH  SAMPLE SOFT       *
40    !*                                                           *
50    !*                                                ED_SEARCH  *
60    !*************************************************************
70    !
80    !-----------------------------------------------------------!
90    !                          MAIN ROUTINE                     !
100   !-----------------------------------------------------------!
110   !
120   LET Ad1=700               ! Set Device Address (PPG)
130   LET Ad2=701               ! Set Device Address (ED)
140   CLEAR Ad1                 ! Device Clear (PPG)
150   CLEAR Ad2                 ! Device Clear (ED)
160    !
170   OUTPUT Ad2;"DTM 0;CTM 0"
180    !
190   LOOP
200      !
210     CLEAR SCREEN
220     PRINT "** MP1702A/MP1609A/MP1651A  AUTO SEARCH SAMPLE SOFT **"
230     PRINT
240      !
250     GOSUB D_set             ! Data setting
260     GOSUB Clock             ! Check Clock loss ?
270     GOSUB Srch              ! Auto search : ON
280     GOSUB Result            ! Display Result
290      !
300     INPUT " NEXT DATA SET  [ YES=0 , NO=1 ]  ?",Loop$
310      !
320   EXIT IF Loop$="1"
330    !
340   END LOOP
350    !
360    !
370   STOP
380    !
390    !-----------------------------------------------------------!
400    !                            SUB ROUTINE                    !
410    !-----------------------------------------------------------!
420    !
430 D_set:  !------------------------------------ SET Device mode
440    !
450   OUTPUT Ad1;"CLK 1;RES 1;FRQ 3000"
460   OUTPUT Ad1;"OFS 0;SPD 0"
470   OUTPUT Ad1;"DAP 1;DOS 1"
480   OUTPUT Ad1;"CDL 100;CAP 1;COS 1"
490   OUTPUT Ad1;"EAD 1"
500    !
510   INPUT " LOGIC MODE [ POSITIVE=0 , NEGATIVE=1 ] ?",Lgc$
520   OUTPUT Ad1;"LGC "&Lgc$
530   OUTPUT Ad2;"LGC "&Lgc$
540      !
550   PRINT "PATTERN MODE [ WORD=0, DATA=1, PN7=2, PN9=3, PN11=5,"
560   PRINT "                PN15=6, PN20=7, PN23=8, PN31=9 ] "
570   PRINT
580      !
590   LOOP
600     INPUT "PATTERN MODE [ 0 OR 1 OR 2 OR 3 OR 5 OR 6 OR 7 OR 8 OR 9 ]?",Ptn
610      !
```

```
620    EXIT IF Ptn<>4 AND Ptn>=0 AND Ptn<=9
630     !
640    END LOOP
650     !
660    OUTPUT Ad1;"PTN "&VAL$(Ptn)
670    OUTPUT Ad2;"PTN "&VAL$(Ptn)
680     !
690     !
700    IF Ptn=0 OR Ptn=1 THEN
710      GOSUB Prog
720    ELSE
730      GOSUB Prbs
740    END IF
750     !
760     !
770    OUTPUT Ad2;"CPL 0"
780    OUTPUT Ad2;"SRH 0"
790    OUTPUT Ad2;"SYN 1"
800     !
810    RETURN
820     !
830     !
840 Clock: !------------------------------------ Check connection
850     !
860    LOOP
870       !
880      GOSUB Connect                    ! Display Device connection
890       !
900      OUTPUT Ad2;"CLI?"
910      ENTER Ad2;Cli$
920       !
930      IF Cli$="CLI 1" THEN PRINT "** CLOCK LOSS **"
940    EXIT IF Cli$="CLI 0"
950    END LOOP
960     !
970    RETURN
980     !
990     !
1000 Srch:  !---------------------------------------- Auto search
1010    !
1020   OUTPUT Ad2;"SRH 1"                          ! AUTO SEARCH : ON
1030    !
1040   LOOP
1050     OUTPUT Ad2;"SRH?"                    ! Request Auto search ?
1060     ENTER Ad2;Srh$
1070      !
1080   EXIT IF Srh$="SRH 0" OR Srh$="SRH 2"
1090   END LOOP
1100    !
1110   IF Srh$="SRH 0" THEN
1120     PRINT "** AUTO SEARCH OK !! **"
1130   ELSE
1140     PRINT "** Failed in AUTO SEARCH !! **"
1150   END IF
1160    !
1170   RETURN
1180    !
1190    !
1200 Result: !------------------------------------ Display Result
1210    !
1220   OUTPUT Ad2;"DTH?"
1230   ENTER Ad2;Dth$
1240    !
```

```
1250  OUTPUT Ad2;"DTM?"
1260  ENTER Ad2;Dtm$
1270  IF Dtm$="DTM 0" THEN
1280    Dtm$="GND"
1290  ELSE
1300    Dtm$="-2V"
1310  END IF
1320   !
1330  OUTPUT Ad2;"CPA?"
1340  ENTER Ad2;Cpa$
1350   !
1360  OUTPUT Ad2;"CTM?"
1370  ENTER Ad2;Ctm$
1380  IF Ctm$="CTM 0" THEN
1390    Ctm$="GND"
1400  ELSE
1410    Ctm$="-2V"
1420  END IF
1430   !
1440  OUTPUT Ad2;"CPL?"
1450  ENTER Ad2;Cpl$
1460  IF Cpl$="CPL 0" THEN
1470    Cpl$="CLK"
1480  ELSE
1490    Cpl$="N CLK"
1500  END IF
1510   !
1520  PRINT "DATA THRESHOLD = "&Dth$[5,10]&" V"
1530  PRINT "DATA TERMINATION = "&Dtm$
1540  PRINT "CLOCK PHASE ADJUST = "&Cpa$[6,9]&" ps"
1550  PRINT "CLOCK TERMINATION = "&Ctm$
1560  PRINT "CLOCK POLARITY = "&Cpl$
1570  PRINT
1580   !
1590  RETURN
1600   !
1610   !
1620 Prog:!------------------------------ SET  PROG (Word),(Data)
1630   !
1640  OUTPUT Ad2;"FRM?"
1650  ENTER Ad2;Frm$
1660  IF Frm$="FRM 1" THEN OUTPUT Ad2;"FRM 0"
1670  !
1680  IF Ptn=1 THEN
1690     !
1700    PRINT "*** PATTERN MODE PROG (DATA) ***"
1710    PRINT
1720     !
1730    OUTPUT Ad1;"DLN 128"
1740    OUTPUT Ad2;"DLN 128"
1750     !
1760  ELSE
1770     !
1780    PRINT "*** PATTERN MODE PROG (WORD) ***"
1790    PRINT
1800     !
1810    OUTPUT Ad1;"WNB 8;WLN 16"
1820    OUTPUT Ad2;"WNB 8;WLN 16"
1830     !
1840  END IF
1850   !
1860  OUTPUT Ad1;"PAG 1;PST 0"
1870  OUTPUT Ad2;"PAG 1;PST 0"
1880   !
```

```
1890  FOR I=2 TO 8
1900    OUTPUT Ad1;"PAG "&VAL$(I)&";PST 1"
1910  NEXT I
1920   !
1930  FOR I=2 TO 8
1940    OUTPUT Ad2;"PAG "&VAL$(I)&";PST 1"
1950  NEXT I
1960   !
1970  RETURN
1980   !
1990   !
2000 Prbs:!---------------------------------------- SET  Mark ratio
2010   !
2020  LOOP
2030    INPUT "MARK RATIO [ 1/8:7/8=1, 1/4:3/4=2, 1/2:N1/2=3 ?",Mrk$
2040  EXIT IF Mrk$="1" OR Mrk$="2" OR Mrk$="3"
2050  END LOOP
2060   !
2070  OUTPUT Ad1;"MRK "&Mrk$
2080  OUTPUT Ad2;"MRK "&Mrk$
2090   !
2100  RETURN
2110   !
2120   !
2130 Connect: !-------------------------------- Display Connection
2140   !
2150  PEN 3
2160  VIEWPORT 70,140,50,100
2170  SHOW 0,70,0,50
2180   !
2190  CLIP 0,70,5,70
2200  FRAME
2210   !
2220  CSIZE 3,.4
2230  MOVE 25,45
2240  LABEL "<< CONNECTION >>"
2250   !
2260  CSIZE 3,.35
2270  MOVE 6,39
2280  LABEL "MP1701B/MP1608A/MP1650A     MP1702A/MP1609A/MP1651A"
2290  !
2300  MOVE 7,20
2310  RECTANGLE 25,18
2320   !
2330  MOVE 38,20
2340  RECTANGLE 25,18
2350   !
2360  MOVE 26,14
2370  IDRAW 0,9
2380   !
2390  FOR I=0 TO PI*2 STEP PI/12
2400    IDRAW .2*COS(I),.2*SIN(I)
2410  NEXT I
2420   !
2430  MOVE 26,14
2440  IDRAW 21,0
2450  IDRAW 0,9
2460   !
2470  FOR I=0 TO PI*2 STEP PI/12
2480    IDRAW .2*COS(I),.2*SIN(I)
2490  NEXT I
2500   !
2510  MOVE 21,17
2520  IDRAW 0,6
```

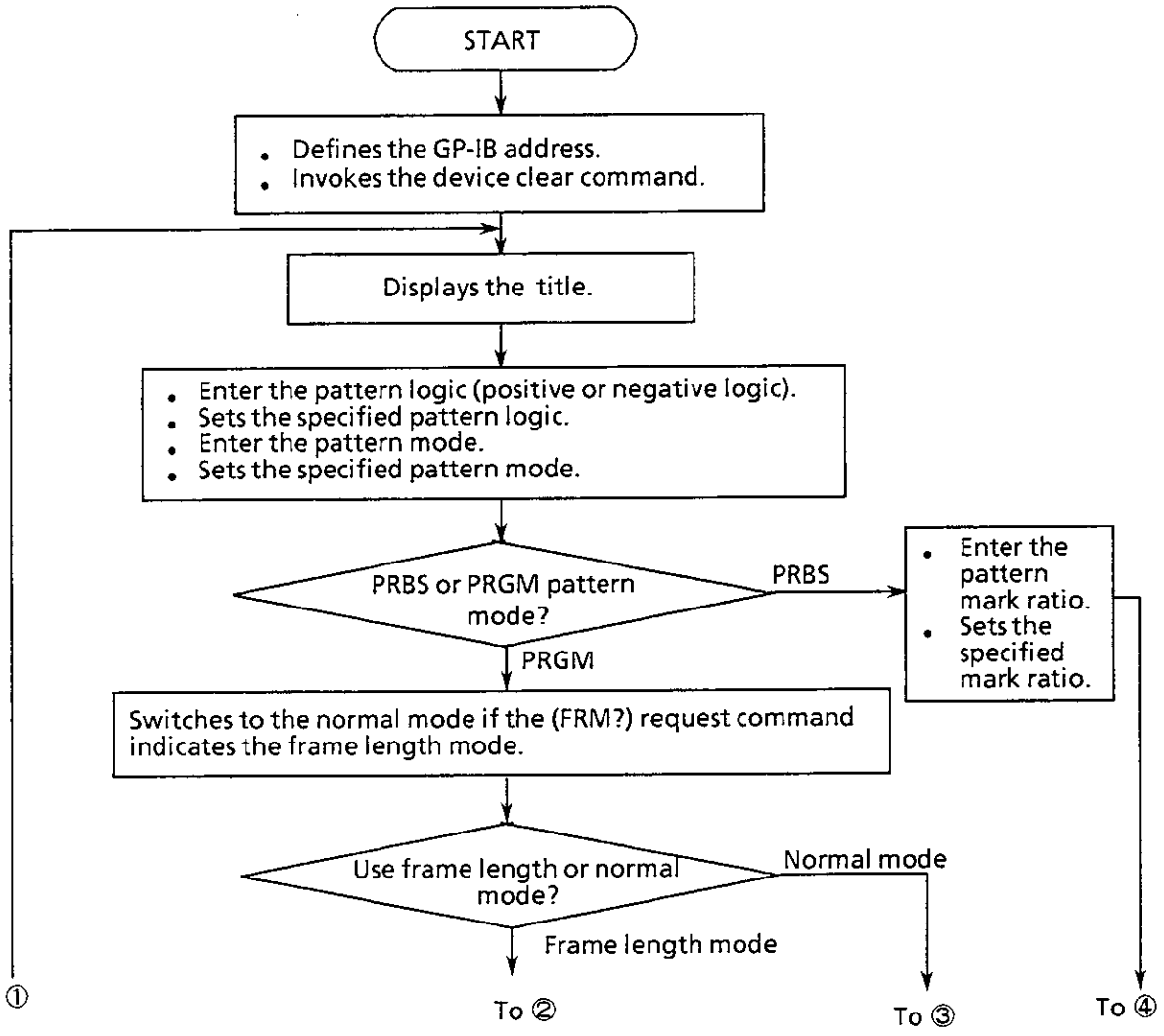
```
2530   !
2540  FOR I=0 TO PI*2 STEP PI/12
2550    IDRAW .2*COS(I),.2*SIN(I)
2560  NEXT I
2570   !
2580  MOVE 21,17
2590  IDRAW 21,0
2600  IDRAW 0,6
2610   !
2620  FOR I=0 TO PI*2 STEP PI/12
2630    IDRAW .2*COS(I),.2*SIN(I)
2640  NEXT I
2650   !
2660  MOVE 16,25
2670  CSIZE 2.3,.5
2680  LABEL "DATA  CLOCK1       DATA  CLOCK"
2690   !
2700  INPUT "Aer you ready ?  Press return key to start.",A
2710   !
2720  RETURN
2730   !
2740  END
```

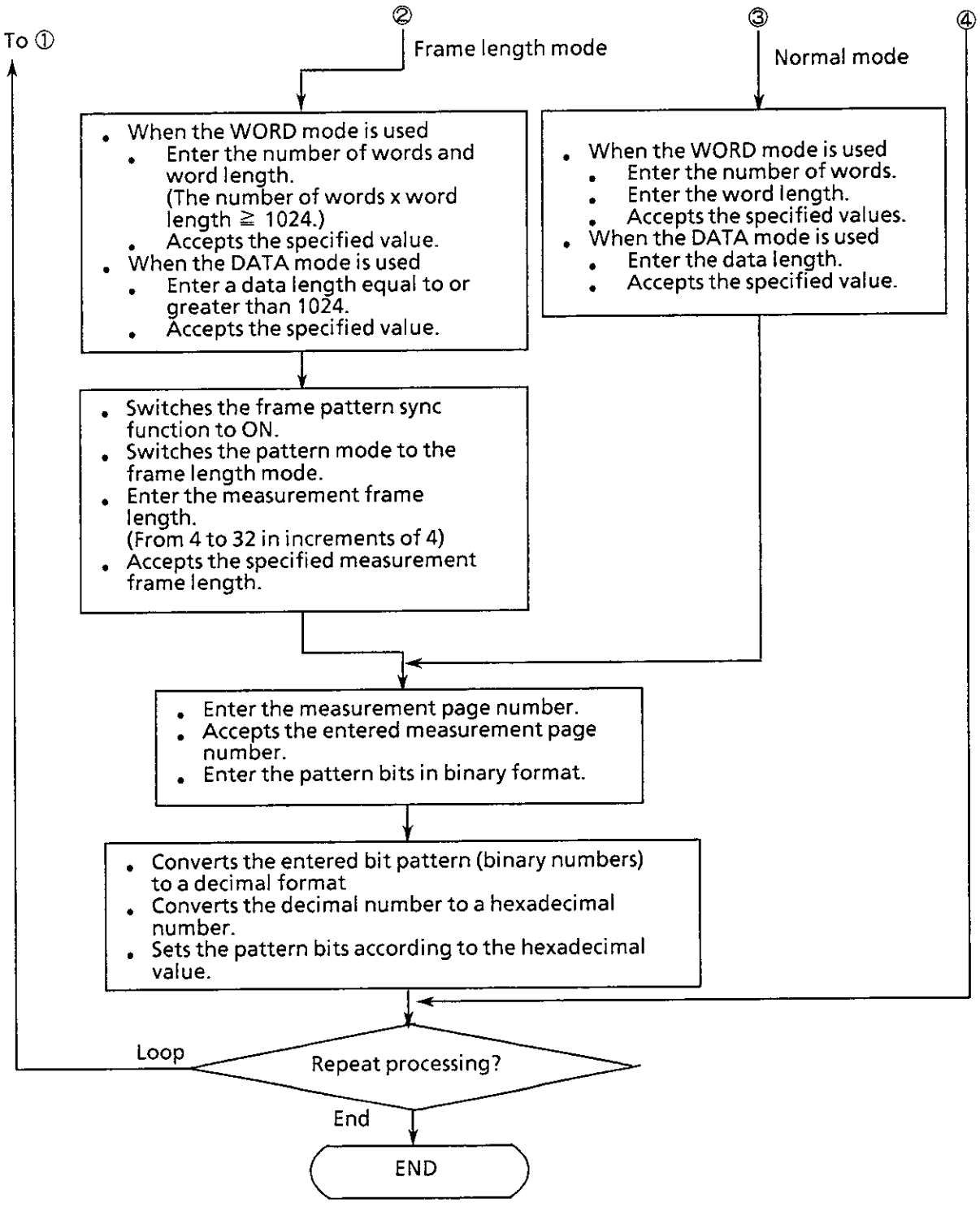
(3) Setting measurement patterns

This program is used to control the characteristics and features of measurement patterns.

First, the program selects the measurement pattern logic and pattern mode. Then, it sets necessary conditions for each measurement pattern mode. However, if the PRGM mode is set, either the WORD (normal mode) or FRAME (frame length mode) data length may be selected along with the conditions for that mode.

Note that the hexadecimal pattern data from the previously entered page number is set in the PRGM mode.




To ①
②
Frame length mode
③
Normal mode
④
• When the WORD mode is used
• Enter the number of words and word length. (The number of words x word length ≧ 1024.)
• Accepts the specified value.
• When the DATA mode is used
• Enter a data length equal to or greater than 1024.
• Accepts the specified value.
• When the WORD mode is used
• Enter the number of words.
• Enter the word length.
• Accepts the specified values.
• When the DATA mode is used
• Enter the data length.
• Accepts the specified value.
• Switches the frame pattern sync function to ON.
• Switches the pattern mode to the frame length mode.
• Enter the measurement frame length. (From 4 to 32 in increments of 4)
• Accepts the specified measurement frame length.
• Enter the measurement page number.
• Accepts the entered measurement page number.
• Enter the pattern bits in binary format.
• Converts the entered bit pattern (binary numbers) to a decimal format
• Converts the decimal number to a hexadecimal number.
• Sets the pattern bits according to the hexadecimal value.
Loop
Repeat processing?
End
END

Program listing

```
10    !*************************************************************
20    !*                                                           *
30    !*      MP1702A/MP1609A/MP1651A  PATTERN SAMPLE SOFT         *
40    !*                                                           *
50    !*                                                 ED_PAT    *
60    !*************************************************************
70    !
80    !-----------------------------------------------------------!
90    !                          MAIN ROUTINE                     !
100   !-----------------------------------------------------------!
110   !
120   LET Add=701                ! Set Device Address
130   CLEAR Add                  ! Device Clear
140    !
150   LOOP
160      !
170     CLEAR SCREEN
180      !
190     PRINT "*** MP1702A/MP1609A/MP1651A  PATTERN SAMPLE SOFT ***"
200     PRINT
210      !
220     GOSUB Pattern
230      !
240     INPUT " NEXT DATA SET  [ YES=0 , NO=1 ]  ?",Loop$
250      !
260   EXIT IF Loop$="1"
270    !
280   END LOOP
290    !
300    !
310   STOP
320    !
330    !-----------------------------------------------------------!
340    !                          SUB ROUTINE                      !
350    !-----------------------------------------------------------!
360    !
370 Pattern:!------------------------------ SET Logic , Pattern mode
380    !
390   INPUT "LOGIC MODE [ POSITIVE=0 , NEGATIVE=1 ] ?",Lgc$
400   OUTPUT Add;"LGC "&Lgc$
410      !
420   LOOP
430     PRINT "PATTERN MODE [ WORD=0, DATA=1, PN7=2, PN9=3, PN11=5,"
440     PRINT "                PN15=6, PN20=7, PN23=8, PN31=9 ] "
450     PRINT
460      INPUT "PATTERN MODE [ 0 OR 1 OR 2 OR 3 OR 5 OR 6 OR 7 OR 8 OR 9 ]?",Ptn
470   EXIT IF Ptn<>4 AND Ptn>=0 AND Ptn<=9
480   END LOOP
490    !
500   OUTPUT Add;"PTN "&VAL$(Ptn)
510    !
520    !
530   IF Ptn=0 OR Ptn=1 THEN
540      !
550     GOSUB Prog_mode
560      !
570   ELSE
580      !
590     LOOP
600       INPUT "MARK RATIO [ 0/8:8/8=0, 1/8:7/8=1, 1/4:3/4=2, 1/2:N1/2=3 ?",Mrk
610     EXIT IF Mrk>=0 AND Mrk<=3
```

```
620      END LOOP
630       !
640      OUTPUT Add;"MRK "&VAL$(Mrk)
650       !
660    END IF
670     !
680    RETURN
690     !
700     !
710 Prog_mode:!----------------------Select FRAME/WORD & Set Page,Bit
720     !
730    OUTPUT Add;"FRM?"                    ! Request Frame/Word length ?
740    ENTER Add;Frm$
750     !
760    IF Frm$="FRM 1" THEN OUTPUT Add;"FRM 0"
770     !
780    LOOP
790      INPUT "FRAME/WORD LENGTH MODE ? [ WORD=0 , FRAME=1 ]",Mod
800    EXIT IF Mod=0 OR Mod=1
810    END LOOP
820     !
830    IF Mod=1 THEN
840      GOSUB Frmlen
850    ELSE
860      GOSUB Wrdlen
870    END IF
880     !
890    INPUT " PAGE ?",Pag$
900    OUTPUT Add;"PAG "&Pag$
910     !
920    INPUT " BIT PATTERN SET DATA  BIT16-->BIT1 [0/1] ?",Bit$
930    GOSUB Btoh
940    OUTPUT Add;"BIT #H"&B$
950     !
960    RETURN
970     !
980     !
990 Frmlen:  !---------------------------------- Frame length mode
1000    !
1010   PRINT "*** Setting FRAME LENGTH mode ***"
1020   PRINT
1030    !
1040   IF Ptn=1 THEN
1050     LOOP
1060       INPUT " DATA LENGTH  [ 1024 ~ 524288 ] ?",Dln
1070     EXIT IF Dln>=1024 AND Dln<=524288
1080     END LOOP
1090        !
1100     OUTPUT Add;"DLN "&VAL$(Dln)
1110        !
1120   ELSE
1130     PRINT "(*1) --- NUMBER OF WORD X WORD LENGTH >=1024 ---"
1140        !
1150     LOOP
1160        !
1170       LOOP
1180         INPUT " NUMBER OF WORD  [ 64~32768 ]   (*1) ?",Wnb
1190       EXIT IF Wnb>=64 AND Wnb<=32768
1200       END LOOP
1210       OUTPUT Add;"WNB "&VAL$(Wnb)
1220          !
1230       LOOP
1240         INPUT " WORD LENGTH [ 2~16 ]   (*1) ?",Wln
1250       EXIT IF Wln>=2 AND Wln<=16
1260       END LOOP
1270       OUTPUT Add;"WLN "&VAL$(Wln)
```

```
1280        !
1290     EXIT IF Wnb*Wln>=1024
1300     END LOOP
1310          !
1320   END IF
1330      !
1340   OUTPUT Add;"FSY 1"          ! Frame sync : ON
1350   OUTPUT Add;"FRM 1"          ! Frame/Word length : FRAME
1360    !
1370    !
1380   LOOP
1390     INPUT "FRAME LENGTH [ 4 ~ 32 ] STEP 4 ?",Fln
1400     F=Fln MODULO 4
1410   EXIT IF Fln>=4 AND Fln<=32 AND F=0
1420   END LOOP
1430    !
1440    !
1450   OUTPUT Add;"FLN "&VAL$(Fln)
1460    !
1470   RETURN
1480    !
1490    !
1500 Wrdlen:  !---------------------------------- Word length mode
1510    !
1520   PRINT "*** Setting  NORMAL LENGTH mode ***"
1530   PRINT
1540    !
1550   IF Ptn=1 THEN
1560      !
1570     LOOP
1580       INPUT " DATA LENGTH [ 2~524288 ] ?",Dln
1590     EXIT IF Dln>=2 AND Dln<=524288
1600     END LOOP
1610      !
1620     OUTPUT Add;"DLN "&VAL$(Dln)
1630      !
1640   ELSE
1650      !
1660     LOOP
1670       INPUT " NUMBER OF WORD [ 1~32768 ] ?",Wnb
1680     EXIT IF Wnb>=1 AND Wnb<=32768
1690     END LOOP
1700      !
1710     OUTPUT Add;"WNB "&VAL$(Wnb)
1720      !
1730     LOOP
1740       INPUT " WORD LENGTH [ 2~16 ] ?",Wln
1750     EXIT IF Wln>=2 AND Wln<=16
1760     END LOOP
1770      !
1780     OUTPUT Add;"WLN "&VAL$(Wln)
1790      !
1800   END IF
1810    !
1820   RETURN
1830    !
1840    !
1850 Btoh: !---------------------------------------- BIN TO HEX
1860    !
1870   A=IVAL(Bit$,2)              ! BIN --> DIG
1880   B$=IVAL$(A,16)              ! DIG --> HEX
1890    !
1900   RETURN
1910    !
1920    !
1930   END
```

(4) Measurement (Measurement type: Output data each time a measurement is completed)

The following programs (a) to (g), are created for measurement where the measurement result GP-IB output selection is set for output to be made each time a measurement is completed.

In these programs, measurement is initiated after connecting DATA to DATA and CLOCK1 to CLOCK between the MP1702A/MP1756A/MP1609A/MP1651A and MP1701B/MP1755A/MP1608A/MP1650A.

First, the program initializes conditions for auto search and for the appropriate measurement.

Next, it confirms that the clock is not lost and, after confirmation, switches the auto search function ON.

If auto search is successfully completed, measurement is begun.

When measurement is completed, the measurement results are read and displayed on the CRT screen. The data read terminate conditions depend on the print format.

If auto search fails, the program is aborted.

Table 15-3 shows methods for determining the end of measurement, the measurement result print format, and the whether one-second data will be printed.

Table 15-3

Paragraph	Data print format	Data output format (Standard or Extended)	Method of determining end of measurement	Print one-second data
(a)	Standard	Standard	Serial polling	As selected
(b)	Standard	Standard	Request command STB?	No
(c)	Standard	Standard	Request command MSR?	No
(d)	Standard	Extended: Includes threshold and error performance measurement data	Serial polling	No
(e)	Standard	Extended: Includes threshold measurement data	Serial polling	No
(f)	Standard	Extended: Includes error performance measurement data	Serial polling	No
(g)	Short (abridged)	Standard	Serial polling	No

Note: In paragraphs (b) and (c), the end of measurement is determined using request command (STB?) or (MSR?). Therefore, since these commands are continuously sent during measurement, the MP1702A/MP1756A/MP1609A/MP1651A remains the talker until measurement is completed. Therefore, measurement results are overwritten by the results of the (STB?) or (MSR?) request command.

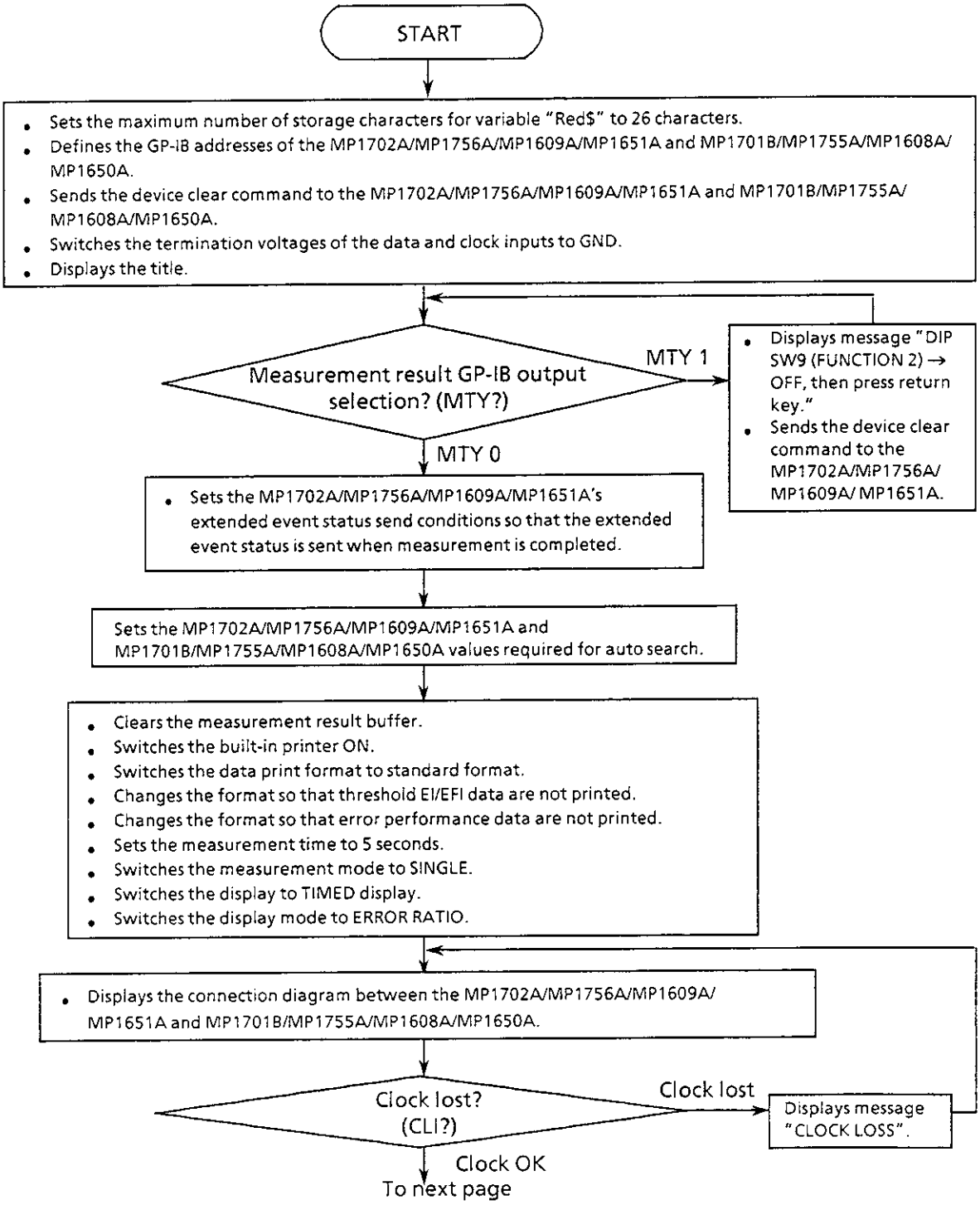
To prevent data loss, the results of the (STB?) or (MSR?) command are read and followed by an interface clear command which cancels the talker specification so that measurement data can be output and read.

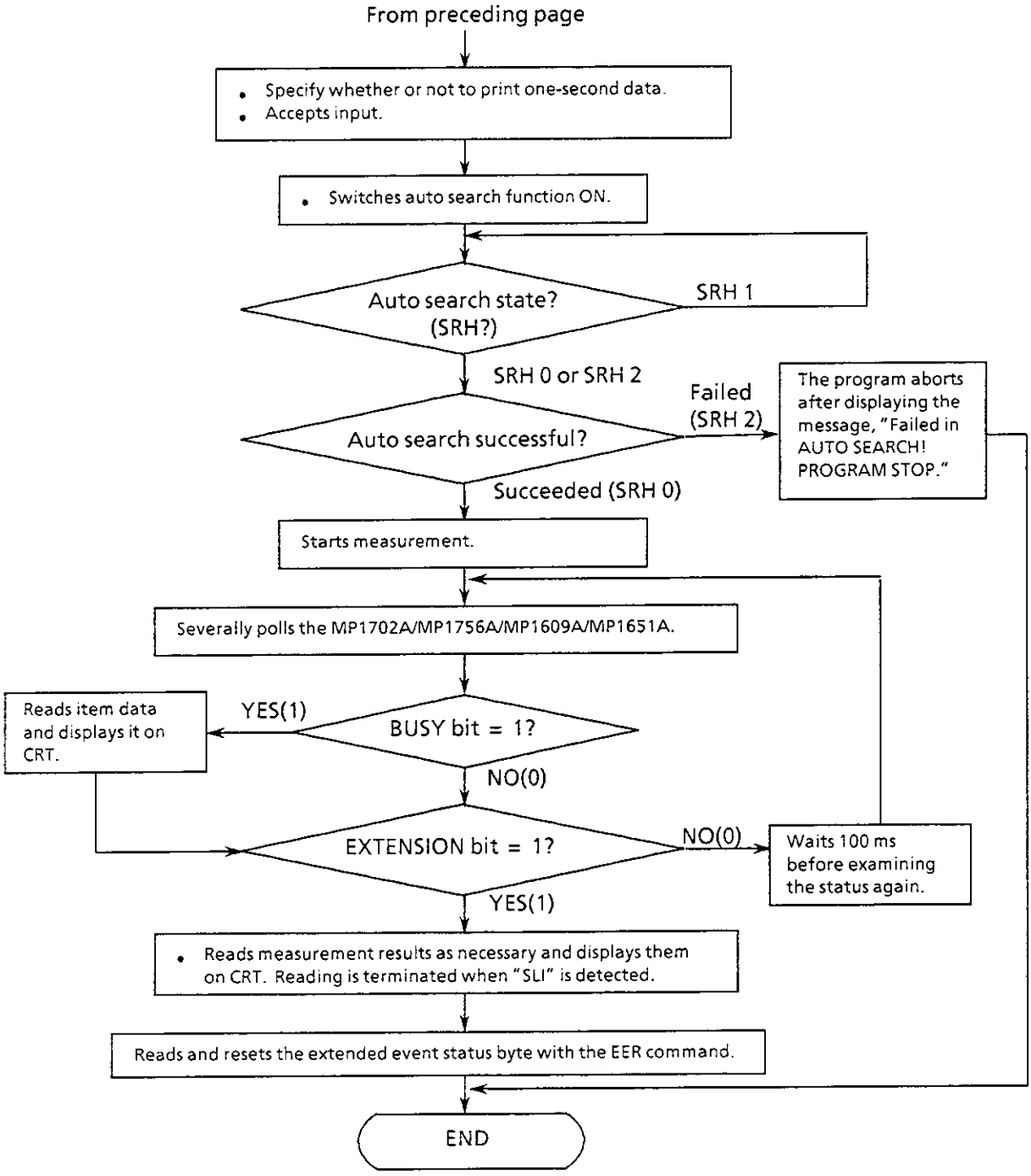
To simplify programming, the end of measurement should be determined by serial polling.

When using the MP1755A for the transmitter, internal clock frequency setting is ineffective. Set the frequency of the external clock source to 3000 MHz in advance.

(a) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Standard data are output in standard format at the end of measurement, as determined by serial polling.




From preceding page
• Specify whether or not to print one-second data.
• Accepts input.
• Switches auto search function ON.
Auto search state? (SRH?)
SRH 1
SRH 0 or SRH 2
Auto search successful?
Failed (SRH 2)
The program aborts after displaying the message, "Failed in AUTO SEARCH! PROGRAM STOP."
Succeeded (SRH 0)
Starts measurement.
Severally polls the MP1702A/MP1756A/MP1609A/MP1651A.
BUSY bit = 1?
YES(1)
Reads item data and displays it on CRT.
NO(0)
EXTENSION bit = 1?
NO(0)
Waits 100 ms before examining the status again.
YES(1)
• Reads measurement results as necessary and displays them on CRT. Reading is terminated when "SLI" is detected.
Reads and resets the extended event status byte with the EER command.
END

Program listing

```
10    !*************************************************************
20    !*                                                           *
30    !*    MP1702A/MP1609A/MP1651A  MEASUREMENT  SAMPLE SOFT      *
40    !*          -- Standard format , Standard output --          *
50    !*         -- Measurement end ? >> Serial polling --         *
60    !*                                                  ED_M1    *
70    !*************************************************************
80    !
90    !-----------------------------------------------------------!
100   !                        MAIN ROUTINE                       !
110   !-----------------------------------------------------------!
120   !
130   DIM Red$[26]
140   LET Ad1=700              ! Set Device Address (PPG)
150   LET Ad2=701              ! Set Device Address (ED)
160   CLEAR Ad1                ! Device Clear (PPG)
170   CLEAR Ad2                ! Device Clear (ED)
180   OUTPUT Ad2;"DTM 0;CTM 0"
190    !
200   CLEAR SCREEN
210    !
220   PRINT "** MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT **"
230   PRINT
240    !
250   GOSUB M_type             ! Check Measurement type
260    !
270   OUTPUT Ad2;"EES 64"      ! EES : MEASURE bit
280    !
290   GOSUB D_set              ! Data setting
300   GOSUB Mode               ! Measurement mode
310   GOSUB Clock              ! Check clock loss
320   GOSUB One_sec            ! One second data print ?
330   GOSUB Srch               ! Auto search : ON
340    !
350   OUTPUT Ad2;"STA"         ! Measurement Start
360    !
370   GOSUB S_poll             ! Serial polling-->Extension bit=1
380   GOSUB Result             ! Display Result
390    !
400   STOP
410    !
420    !-----------------------------------------------------------!
430    !                          SUB ROUTINE                      !
440    !-----------------------------------------------------------!
450    !
460 M_type: !---------------------------- Check Measurement type
470   !
480   LOOP
490      !
500     OUTPUT Ad2;"MTY?"
510     ENTER Ad2;Mty$
520      !
530     IF Mty$="MTY 1" THEN
540       INPUT "DIP SW 9 (Function 2)-->OFF, then press return key",A
550       CLEAR Ad2
560     END IF
570      !
580   EXIT IF Mty$="MTY 0"
590   END LOOP
600    !
610   RETURN
```

```
620     !
630     !
640 D_set:  !------------------------------------------ Set Data
650     !
660    OUTPUT Ad1;"LGC 0;PTN 2;MRK 1"
670    OUTPUT Ad2;"LGC 0;PTN 2;MRK 1"
680     !
690    OUTPUT Ad1;"CLK 1;RES 1;FRQ 3000"
700    OUTPUT Ad1;"OFS 0;SPD 0"
710    OUTPUT Ad1;"DAP 1;DOS 1"
720    OUTPUT Ad1;"CDL 100;CAP 1;COS 1"
730    OUTPUT Ad1;"EAD 1"
740     !
750    OUTPUT Ad2;"CPL 0"
760    OUTPUT Ad2;"SRH 0"
770    OUTPUT Ad2;"SYN 1"
780     !
790    RETURN
800     !
810     !
820 Mode: !------------------------------------ Measurement mode
830     !
840    OUTPUT Ad2;"BCL"          ! Measurement buffer clear
850    OUTPUT Ad2;"PRN 1"        ! Printer ON
860    OUTPUT Ad2;"FMT 0"        ! Data Output:Standard format
870    OUTPUT Ad2;"THR 0"        ! Threshold output : OFF
880    OUTPUT Ad2;"EPF 0"        ! Error performance Output:OFF
890     !
900    OUTPUT Ad2;"PRD 0,0,0,5" ! Measurement period : 5sec
910    OUTPUT Ad2;"MOD 1"        ! Measurement mode : SINGLE
920    OUTPUT Ad2;"TIM 3"        ! Display TIMED
930    OUTPUT Ad2;"DMS 0"        ! Display ERROR RATIO
940     !
950    RETURN
960     !
970     !
980 Clock: !------------------------------------ Check connection
990     !
1000   LOOP
1010     GOSUB Connect           ! Display Device connection
1020      !
1030     OUTPUT Ad2;"CLI?"       ! REQUEST Clock loss ?
1040     ENTER Ad2;Cli$
1050      !
1060     IF Cli$="CLI 1" THEN
1070       PRINT "** CLOCK LOSS **"
1080     END IF
1090      !
1100   EXIT IF Cli$="CLI 0"
1110   END LOOP
1120    !
1130   RETURN
1140    !
1150    !
1160 One_sec: !---------------------------- One second data output ?
1170    !
1180   INPUT "One second data output [ OFF=0 , ON=1 ]",Osc$
1190   OUTPUT Ad2;"OSC "&Osc$
1200    !
1210   RETURN
1220    !
1230    !
```

```
1240 Srch:!------------------------------------------ Auto search
1250   !
1260  OUTPUT Ad2;"SRH 1"                              ! AUTO SEARCH : ON
1270   !
1280  LOOP
1290     !
1300    OUTPUT Ad2;"SRH?"
1310    ENTER Ad2;Srh$
1320     !
1330  EXIT IF Srh$="SRH 0" OR Srh$="SRH 2"
1340  END LOOP
1350   !
1360  IF Srh$="SRH 2" THEN
1370    PRINT "** Failed in AUTO SEARCH !  PROGRAM STOP **"
1380    STOP
1390     !
1400  END IF
1410   !
1420  RETURN
1430   !
1440   !
1450 S_poll:!------------------------------------- Serial Polling
1460   !
1470  LOOP
1480    A=SPOLL(Ad2)                                ! Serial Polling
1490     !
1500    IF BIT(A,4)=1 THEN                          ! BUSY bit=1
1510      GOSUB Item_read
1520    END IF
1530     !
1540  EXIT IF BIT(A,1)=1                            ! EXTENSION bit = 1
1550    WAIT .1
1560  END LOOP
1570   !
1580  RETURN
1590   !
1600   !
1610 Item_read: !--------------------------- Read & Print Item data
1620   !
1630  ENTER Ad2;Red$
1640  PRINT Red$
1650   !
1660  RETURN
1670   !
1680   !
1690 Result: !------------------------------------- Print Result
1700   !
1710  LOOP
1720    ENTER Ad2;Red$
1730    PRINT Red$
1740  EXIT IF Red$[1,3]="SLI"
1750  END LOOP
1760   !
1770  OUTPUT Ad2;"EER?"                  ! RESET Extension register
1780  ENTER Ad2;Eer$
1790   !
1800  RETURN
1810   !
1820   !
```

```
1830 Connect: !-------------------------------- Display Connection
1840   !
1850  PEN 3
1860  VIEWPORT 70,140,50,100
1870  SHOW 0,70,0,50
1880   !
1890  CLIP 0,70,5,70
1900  FRAME
1910   !
1920  CSIZE 3,.4
1930  MOVE 25,45
1940  LABEL "<< CONNECTION >>"
1950   !
1960  CSIZE 3,.35
1970  MOVE 6,39
1980  LABEL "MP1701B/MP1608A/MP1650A     MP1702A/MP1609A/MP1651A"
1990  !
2000  MOVE 7,20
2010  RECTANGLE 25,18
2020   !
2030  MOVE 38,20
2040  RECTANGLE 25,18
2050   !
2060  MOVE 26,14
2070  IDRAW 0,9
2080   !
2090  FOR I=0 TO PI*2 STEP PI/12
2100    IDRAW .2*COS(I),.2*SIN(I)
2110  NEXT I
2120   !
2130  MOVE 26,14
2140  IDRAW 21,0
2150  IDRAW 0,9
2160   !
2170  FOR I=0 TO PI*2 STEP PI/12
2180    IDRAW .2*COS(I),.2*SIN(I)
2190  NEXT I
2200   !
2210  MOVE 21,17
2220  IDRAW 0,6
2230   !
2240  FOR I=0 TO PI*2 STEP PI/12
2250    IDRAW .2*COS(I),.2*SIN(I)
2260  NEXT I
2270   !
2280  MOVE 21,17
2290  IDRAW 21,0
2300  IDRAW 0,6
2310   !
2320  FOR I=0 TO PI*2 STEP PI/12
2330    IDRAW .2*COS(I),.2*SIN(I)
2340  NEXT I
2350   !
2360  MOVE 16,25
2370  CSIZE 2.3,.5
2380  LABEL "DATA  CLOCK1       DATA  CLOCK"
2390   !
2400  INPUT "Aer you ready ?  Press return key to start.",A
2410   !
2420  RETURN
2430   !
2440  END
```

EXECUTED RESULT

① When one-second data is not specified for printing

Output to GP-IB

```
STA 90,02,26,11,52,33
STA 90,02,26,11,52,33
END 90,02,26,11,52,38
ELP      0,00,00,05
ER  1.0000E-04
EC      2500006
EI            5
EFI      0.0000
PFI           0
CLI           0
SLI           0
```

Printout from built-in printer

```
90-02-26/11:52:33 START

**************************
 START 90-02-26/11:52:33
 END   90-02-26/11:52:38
 ELP         0/00:00:05
 **ERROR MEASUREMENT**
 ERROR RATIO 1.0000E-04
 ERROR COUNT    2500006
 EI                   5
 EFI               0.0000%
 PFI                  0
 CLI                  0
 SLI                  0
**************************
```

② When one-second data is specified for printing

Output to GP-IB

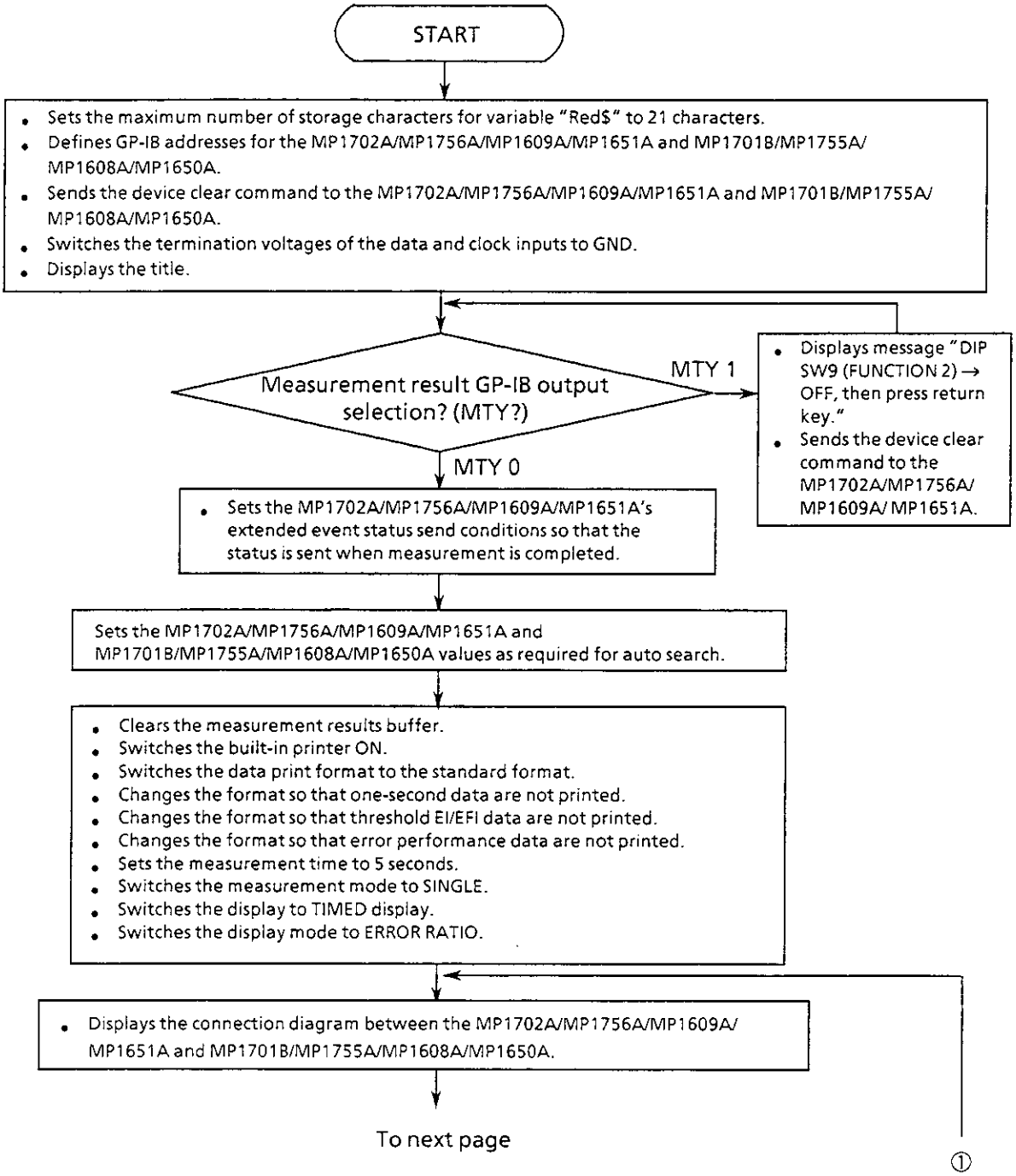
```
STA 90,02,26,11,53,03
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500002
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
STA 90,02,26,11,53,03
END 90,02,26,11,53,08
ELP      0,00,00,05
ER  1.0000E-04
EC      2500006
EI            5
EFI      0.0000
PFI           0
CLI           0
SLI           0
```

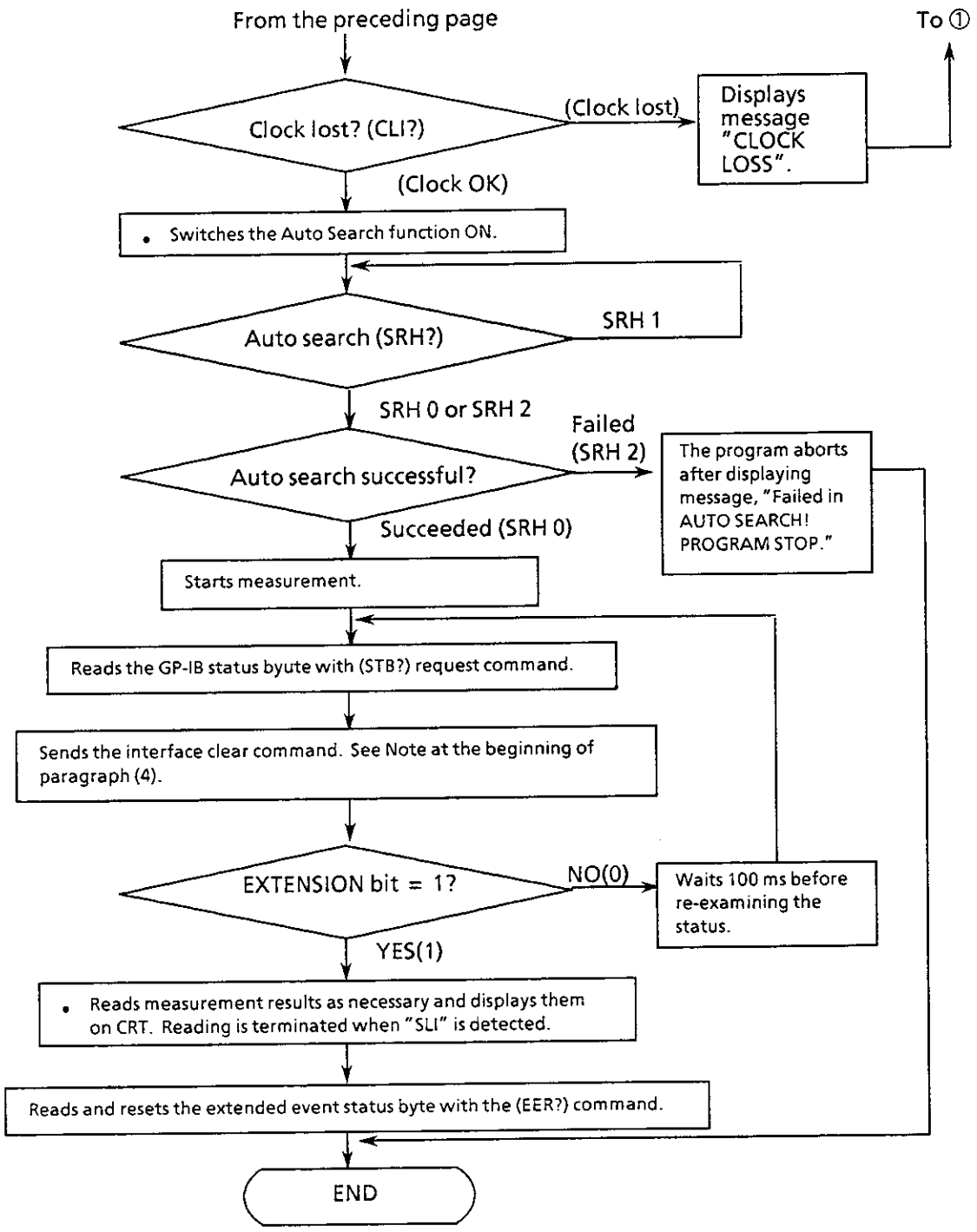
Printout from built-in printer

```
90-02-26/11:53:03 START
53:04 1.00E-04    500001
53:05 1.00E-04    500002
53:06 1.00E-04    500001
53:07 1.00E-04    500001
53:08 1.00E-04    500001

**************************
 START 90-02-26/11:53:03
 END   90-02-26/11:53:08
 ELP         0/00:00:05
 **ERROR MEASUREMENT**
 ERROR RATIO 1.0000E-04
 ERROR COUNT    2500006
 EI                   5
 EFI               0.0000%
 PFI                  0
 CLI                  0
 SLI                  0
**************************
```

(b) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Standard data is output in the standard format at the end of measurement, as determined by the (STB?) request command.




From the preceding page
To ①
Clock lost? (CLI?)
(Clock lost)
Displays message "CLOCK LOSS".
(Clock OK)
• Switches the Auto Search function ON.
Auto search (SRH?)
SRH 1
SRH 0 or SRH 2
Failed (SRH 2)
Auto search successful?
The program aborts after displaying message, "Failed in AUTO SEARCH! PROGRAM STOP."
Succeeded (SRH 0)
Starts measurement.
Reads the GP-IB status byute with (STB?) request command.
Sends the interface clear command. See Note at the beginning of paragraph (4).
EXTENSION bit = 1?
NO(0)
Waits 100 ms before re-examining the status.
YES(1)
• Reads measurement results as necessary and displays them on CRT. Reading is terminated when "SLI" is detected.
Reads and resets the extended event status byte with the (EER?) command.
END

Program listing

```
10    !*************************************************************
20    !*                                                           *
30    !*    MP1702A/MP1609A/MP1651A  MEASUREMENT  SAMPLE SOFT      *
40    !*        -- Standard format , Standard output--             *
50    !*        -- Measurement end ? >> Request STB? --            *
60    !*                                                  ED_M2    *
70    !*************************************************************
80    !
90    !-----------------------------------------------------------!
100   !                        MAIN ROUTINE                       !
110   !-----------------------------------------------------------!
120   !
130   DIM Red$[21]
140   LET Ad1=700               ! Set Device Address (PPG)
150   LET Ad2=701               ! Set Device Address (ED)
160   CLEAR Ad1                 ! Device Clear (PPG)
170   CLEAR Ad2                 ! Device Clear (ED)
180   OUTPUT Ad2;"DTM 0;CTM 0"
190    !
200   CLEAR SCREEN
210    !
220   PRINT "** MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT **"
230   PRINT
240    !
250   GOSUB M_type              ! Check Measurement type
260    !
270   OUTPUT Ad2;"EES 64"       ! EES : MEASURE bit
280    !
290   GOSUB D_set               ! Data setting
300   GOSUB Mode                ! Measurement mode
310   GOSUB Clock               ! Check clock loss
320   GOSUB Srch                ! Auto search : ON
330    !
340   OUTPUT Ad2;"STA"          ! Measurement Start
350    !
360   GOSUB St_byte             ! Request STB? --> Extension bit=1
370   GOSUB Result              ! Display Result
380    !
390   STOP
400    !
410    !----------------------------------------------------------!
420    !                          SUB ROUTINE                     !
430    !----------------------------------------------------------!
440    !
450 M_type: !------------------------------ Check Measurement type
460   !
470   LOOP
480      !
490     OUTPUT Ad2;"MTY?"
500     ENTER Ad2;Mty$
510      !
520     IF Mty$="MTY 1" THEN
530       INPUT "DIP SW 9 (Function 2)-->OFF, then press return key",A
540       CLEAR Ad2
550     END IF
560      !
570   EXIT IF Mty$="MTY 0"
580   END LOOP
590    !
600   RETURN
610    !
```

```
620 D_set:  !------------------------------------------ Set Data
630    !
640    OUTPUT Ad1;"LGC 0;PTN 2;MRK 1"
650    OUTPUT Ad2;"LGC 0;PTN 2;MRK 1"
660    !
670    OUTPUT Ad1;"CLK 1;RES 1;FRQ 3000"
680    OUTPUT Ad1;"OFS 0;SPD 0"
690    OUTPUT Ad1;"DAP 1;DOS 1"
700    OUTPUT Ad1;"CDL 100;CAP 1;COS 1"
710    OUTPUT Ad1;"EAD 1"
720    !
730    OUTPUT Ad2;"CPL 0"
740    OUTPUT Ad2;"SRH 0"
750    OUTPUT Ad2;"SYN 1"
760    !
770    RETURN
780    !
790 Mode: !------------------------------------ Measurement mode
800    !
810    OUTPUT Ad2;"BCL"          ! Measurement buffer clear
820    OUTPUT Ad2;"PRN 1"        ! Printer ON
830    OUTPUT Ad2;"FMT 0"        ! Data Output:Standard format
840    OUTPUT Ad2;"OSC 0"        ! One second data output : OFF
850    OUTPUT Ad2;"THR 0"        ! Threshold output : OFF
860    OUTPUT Ad2;"EPF 0"        ! Error performance Output:OFF
870    !
880    OUTPUT Ad2;"PRD 0,0,0,5" ! Measurement period : 5sec
890    OUTPUT Ad2;"MOD 1"        ! Measurement mode : SINGLE
900    OUTPUT Ad2;"TIM 3"        ! Display TIMED
910    OUTPUT Ad2;"DMS 0"        ! Display ERROR RATIO
920    !
930    RETURN
940    !
950    !
960 Clock: !------------------------------------ Check connection
970    !
980    LOOP
990      GOSUB Connect           ! Display Device connection
1000     !
1010     OUTPUT Ad2;"CLI?"       ! REQUEST Clock loss ?
1020     ENTER Ad2;Cli$
1030     !
1040     IF Cli$="CLI 1" THEN
1050       PRINT "** CLOCK LOSS **"
1060     END IF
1070     !
1080   EXIT IF Cli$="CLI 0"
1090   END LOOP
1100   !
1110   RETURN
1120   !
1130   !
1140 Srch:!------------------------------------------ Auto search
1150     !
1160   OUTPUT Ad2;"SRH 1"                          ! AUTO SEARCH : ON
1170   !
1180   LOOP
1190     !
1200     OUTPUT Ad2;"SRH?"
1210     ENTER Ad2;Srh$
1220     !
1230   EXIT IF Srh$="SRH 0" OR Srh$="SRH 2"
1240   END LOOP
1250   !
```

```
1260  IF Srh$="SRH 2" THEN
1270    PRINT "** Failed in AUTO SEARCH !  PROGRAM STOP **"
1280    STOP
1290     !
1300  END IF
1310   !
1320  RETURN
1330   !
1340   !
1350 St_byte:!---------------------------------------- Request STB?
1360   !
1370  LOOP
1380     !
1390    OUTPUT Ad2;"STB?"          ! Request Status byte register ?
1400    ENTER Ad2;Stb$
1410     !
1420    ABORT 7
1430     !
1440  EXIT IF Stb$[13,13]="1"      ! EXTENSION bit=1
1450    WAIT .1
1460  END LOOP
1470   !
1480  RETURN
1490   !
1500   !
1510 Result: !---------------------------------------- Print Result
1520   !
1530  LOOP
1540    ENTER Ad2;Red$
1550    PRINT Red$
1560  EXIT IF Red$[1,3]="SLI"
1570  END LOOP
1580   !
1590  OUTPUT Ad2;"EER?"                  ! RESET Extension register
1600  ENTER Ad2;Eer$
1610   !
1620  RETURN
1630   !
1640   !
1650 Connect: !-------------------------------- Display Connection
1660   !
1670  PEN 3
1680  VIEWPORT 70,140,50,100
1690  SHOW 0,70,0,50
1700   !
1710  CLIP 0,70,5,70
1720  FRAME
1730   !
1740  CSIZE 3,.4
1750  MOVE 25,45
1760  LABEL "<< CONNECTION >>"
1770   !
1780  CSIZE 3,.35
1790  MOVE 6,39
1800  LABEL "MP1701B/MP1608A/MP1650A    MP1702A/MP1609A/MP1651A"
1810  !
1820  MOVE 7,20
1830  RECTANGLE 25,18
1840   !
1850  MOVE 38,20
1860  RECTANGLE 25,18
1870   !
1880  MOVE 26,14
1890  IDRAW 0,9
```

```
1900   !
1910  FOR I=0 TO PI*2 STEP PI/12
1920    IDRAW .2*COS(I),.2*SIN(I)
1930  NEXT I
1940   !
1950  MOVE 26,14
1960  IDRAW 21,0
1970  IDRAW 0,9
1980   !
1990  FOR I=0 TO PI*2 STEP PI/12
2000    IDRAW .2*COS(I),.2*SIN(I)
2010  NEXT I
2020   !
2030  MOVE 21,17
2040  IDRAW 0,6
2050   !
2060  FOR I=0 TO PI*2 STEP PI/12
2070    IDRAW .2*COS(I),.2*SIN(I)
2080  NEXT I
2090   !
2100  MOVE 21,17
2110  IDRAW 21,0
2120  IDRAW 0,6
2130   !
2140  FOR I=0 TO PI*2 STEP PI/12
2150    IDRAW .2*COS(I),.2*SIN(I)
2160  NEXT I
2170   !
2180  MOVE 16,25
2190  CSIZE 2.3,.5
2200  LABEL "DATA  CLOCK1       DATA  CLOCK"
2210   !
2220  INPUT "Aer you ready ?  Press return key to start.",A
2230   !
2240  RETURN
2250   !
2260  END
```

Output to GP-IB

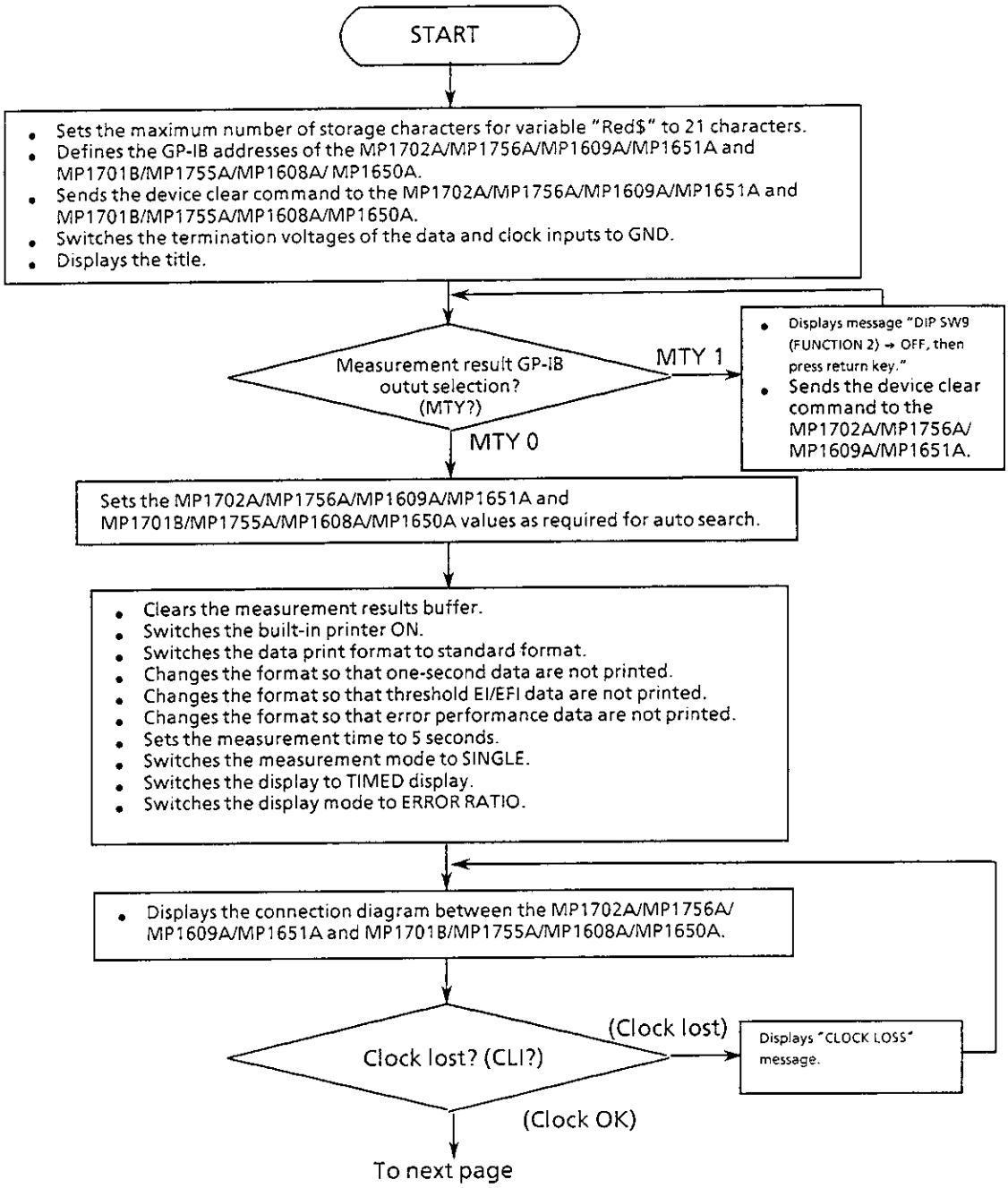
```
STA 90,02,26,11,53,43
STA 90,02,26,11,53,43
END 90,02,26,11,53,48
ELP      0,00,00,05
ER   1.0000E-04
EC       2500006
EI             5
EFI       0.0000
PFI            0
CLI            0
SLI            0
```

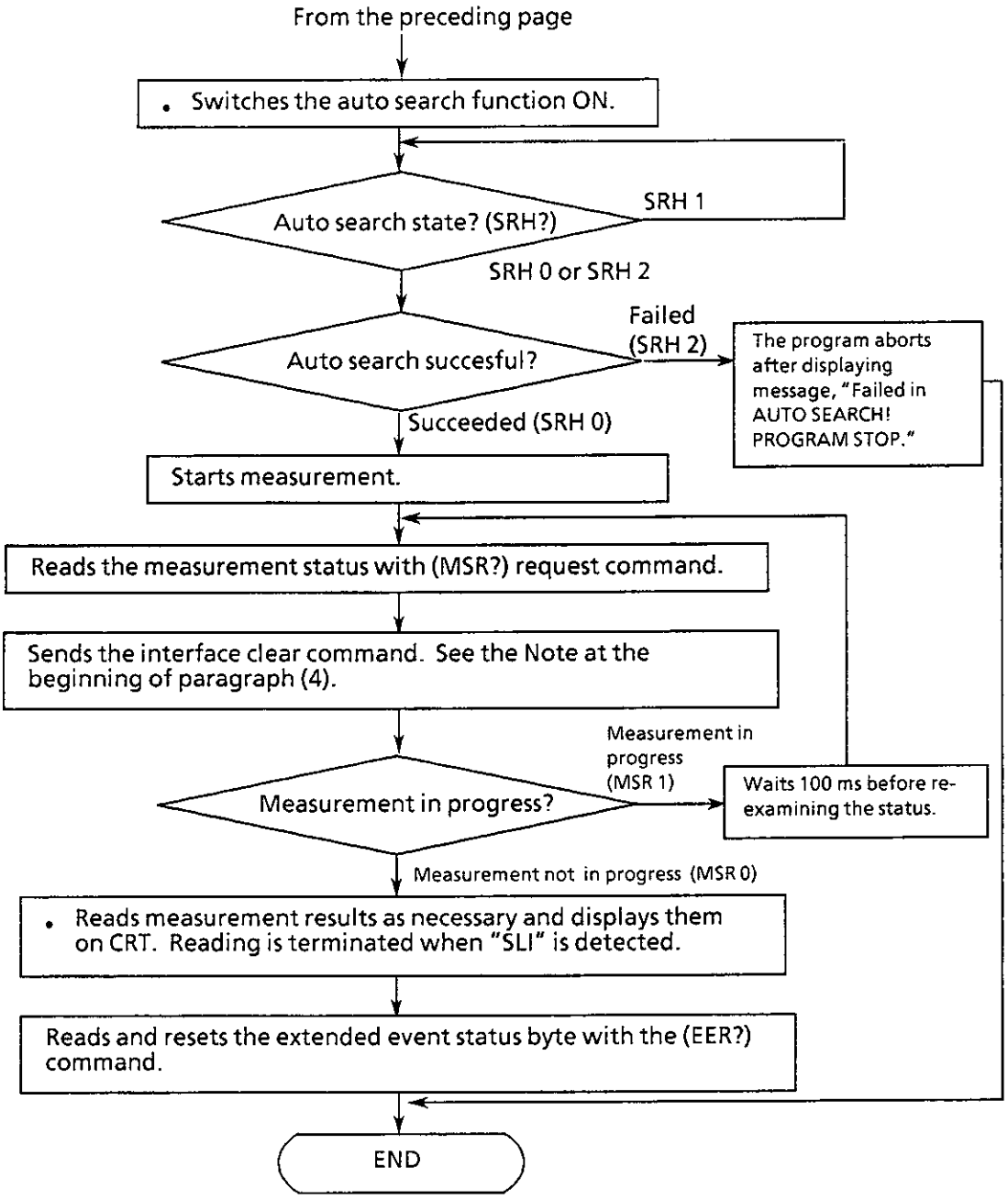
Printout from built-in printer

```
90-02-26/11:53:43 START

**********************
 START 90-02-26/11:53:43
 END   90-02-26/11:53:48
 ELP             0/00:00:05
 --ERROR MEASUREMENT--
 ERROR RATIO 1.0000E-04
 ERROR COUNT     2500006
 EI                    5
 EFI                0.0000%
 PFI                    0
 CLI                    0
 SLI                    0
**********************
```

(c) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Standard data are output in standard format at the end of measuremen, as determined by (MSR?) request command.




From the preceding page
• Switches the auto search function ON.
Auto search state? (SRH?)
SRH 1
SRH 0 or SRH 2
Auto search succesful?
Failed (SRH 2)
The program aborts after displaying message, "Failed in AUTO SEARCH! PROGRAM STOP."
Succeeded (SRH 0)
Starts measurement.
Reads the measurement status with (MSR?) request command.
Sends the interface clear command. See the Note at the beginning of paragraph (4).
Measurement in progress?
Measurement in progress (MSR 1)
Waits 100 ms before re-examining the status.
Measurement not in progress (MSR 0)
• Reads measurement results as necessary and displays them on CRT. Reading is terminated when "SLI" is detected.
Reads and resets the extended event status byte with the (EER?) command.
END

Program listing

```
10    !*************************************************************
20    !*                                                           *
30    !*    MP1702A/MP1609A/MP1651A  MEASUREMENT  SAMPLE SOFT      *
40    !*          -- Standard format , Standard output --          *
50    !*          -- Measurement end ? >> Request MSR? --          *
60    !*                                                  ED_M3    *
70    !*************************************************************
80    !
90    !-----------------------------------------------------------!
100   !                        MAIN ROUTINE                       !
110   !-----------------------------------------------------------!
120   !
130   DIM Red$[21]
140   LET Ad1=700               ! Set Device Address (PPG)
150   LET Ad2=701               ! Set Device Address (ED)
160   CLEAR Ad1                 ! Device Clear (PPG)
170   CLEAR Ad2                 ! Device Clear (ED)
180   OUTPUT Ad2;"DTM 0;CTM 0"
190    !
200   CLEAR SCREEN
210    !
220   PRINT "** MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT **"
230   PRINT
240    !
250   GOSUB M_type              ! Check Measurement type
260    !
270   GOSUB D_set               ! Data setting
280   GOSUB Mode                ! Measurement mode
290   GOSUB Clock               ! Check clock loss
300   GOSUB Srch                ! Auto search : ON
310    !
320   OUTPUT Ad2;"STA"          ! Measurement Start
330    !
340   GOSUB Msr                 ! Request MSR? --> Extension bit=1
350   GOSUB Result              ! Display Result
360    !
370   STOP
380    !
390    !-----------------------------------------------------------!
400    !                          SUB ROUTINE                      !
410    !-----------------------------------------------------------!
420    !
430 M_type: !------------------------------ Check Measurement type
440   !
450   LOOP
460      !
470     OUTPUT Ad2;"MTY?"
480     ENTER Ad2;Mty$
490      !
500     IF Mty$="MTY 1" THEN
510       INPUT "DIP SW 9 (Function 2)-->OFF, then press return key",A
520       CLEAR Ad2
530     END IF
540      !
550   EXIT IF Mty$="MTY 0"
560   END LOOP
570    !
580   RETURN
590    !
600    !
```

```
610 D_set:  !------------------------------------------ Set Data
620    !
630    OUTPUT Ad1;"LGC 0;PTN 2;MRK 1"
640    OUTPUT Ad2;"LGC 0;PTN 2;MRK 1"
650    !
660    OUTPUT Ad1;"CLK 1;RES 1;FRQ 3000"
670    OUTPUT Ad1;"OFS 0;SPD 0"
680    OUTPUT Ad1;"DAP 1;DOS 1"
690    OUTPUT Ad1;"CDL 100;CAP 1;COS 1"
700    OUTPUT Ad1;"EAD 1"
710    !
720    OUTPUT Ad2;"CPL 0"
730    OUTPUT Ad2;"SRH 0"
740    OUTPUT Ad2;"SYN 1"
750    !
760    RETURN
770    !
780    !
790 Mode: !------------------------------------ Measurement mode
800    !
810    OUTPUT Ad2;"BCL"          ! Measurement buffer clear
820    OUTPUT Ad2;"PRN 1"        ! Printer ON
830    OUTPUT Ad2;"FMT 0"        ! Data Output:Standard format
840    OUTPUT Ad2;"OSC 0"        ! One second data output : OFF
850    OUTPUT Ad2;"THR 0"        ! Threshold output : OFF
860    OUTPUT Ad2;"EPF 0"        ! Error performance Output:OFF
870    !
880    OUTPUT Ad2;"PRD 0,0,0,5" ! Measurement period : 5sec
890    OUTPUT Ad2;"MOD 1"        ! Measurement mode : SINGLE
900    OUTPUT Ad2;"TIM 3"        ! Display TIMED
910    OUTPUT Ad2;"DMS 0"        ! Display ERROR RATIO
920    !
930    RETURN
940    !
950    !
960 Clock: !------------------------------------ Check connection
970    !
980    LOOP
990      GOSUB Connect           ! Display Device connection
1000     !
1010     OUTPUT Ad2;"CLI?"       ! REQUEST Clock loss ?
1020     ENTER Ad2;Cli$
1030     !
1040     IF Cli$="CLI 1" THEN
1050       PRINT "** CLOCK LOSS **"
1060     END IF
1070     !
1080   EXIT IF Cli$="CLI 0"
1090   END LOOP
1100   !
1110   RETURN
1120   !
1130   !
1140 Srch:!------------------------------------------ Auto search
1150   !
1160   OUTPUT Ad2;"SRH 1"                          ! AUTO SEARCH : ON
1170   !
1180   LOOP
1190     !
1200     OUTPUT Ad2;"SRH?"
1210     ENTER Ad2;Srh$
1220     !
1230   EXIT IF Srh$="SRH 0" OR Srh$="SRH 2"
1240   END LOOP
```

```
1250   !
1260  IF Srh$="SRH 2" THEN
1270    PRINT "** Failed in AUTO SEARCH !  PROGRAM STOP **"
1280    STOP
1290     !
1300  END IF
1310   !
1320  RETURN
1330   !
1340   !
1350 Msr:!------------------------------------------ Request MSR?
1360   !
1370  LOOP
1380     !
1390    OUTPUT Ad2;"MSR?"        ! Measurement in progress or stop ?
1400    ENTER Ad2;Msr$
1410     !
1420    ABORT 7
1430     !
1440  EXIT IF Msr$="MSR 0"       ! Measurement stop
1450    WAIT .1
1460  END LOOP
1470   !
1480  RETURN
1490   !
1500   !
1510 Result: !------------------------------------- Print Result
1520   !
1530  LOOP
1540    ENTER Ad2;Red$
1550    PRINT Red$
1560  EXIT IF Red$[1,3]="SLI"
1570  END LOOP
1580   !
1590  OUTPUT Ad2;"EER?"                  ! RESET Extension register
1600  ENTER Ad2;Eer$
1610   !
1620  RETURN
1630   !
1640   !
1650 Connect: !-------------------------------- Display Connection
1660   !
1670  PEN 3
1680  VIEWPORT 70,140,50,100
1690  SHOW 0,70,0,50
1700   !
1710  CLIP 0,70,5,70
1720  FRAME
1730   !
1740  CSIZE 3,.4
1750  MOVE 25,45
1760  LABEL "<< CONNECTION >>"
1770   !
1780  CSIZE 3,.35
1790  MOVE 6,39
1800  LABEL "MP1701B/MP1608A/MP1650A    MP1702A/MP1609A/MP1651A"
1810  !
1820  MOVE 7,20
1830  RECTANGLE 25,18
1840   !
1850  MOVE 38,20
1860  RECTANGLE 25,18
1870   !
```

```
1880  MOVE 26,14
1890  IDRAW 0,9
1900   !
1910  FOR I=0 TO PI*2 STEP PI/12
1920    IDRAW .2*COS(I),.2*SIN(I)
1930  NEXT I
1940   !
1950  MOVE 26,14
1960  IDRAW 21,0
1970  IDRAW 0,9
1980   !
1990  FOR I=0 TO PI*2 STEP PI/12
2000    IDRAW .2*COS(I),.2*SIN(I)
2010  NEXT I
2020   !
2030  MOVE 21,17
2040  IDRAW 0,6
2050   !
2060  FOR I=0 TO PI*2 STEP PI/12
2070    IDRAW .2*COS(I),.2*SIN(I)
2080  NEXT I
2090   !
2100  MOVE 21,17
2110  IDRAW 21,0
2120  IDRAW 0,6
2130   !
2140  FOR I=0 TO PI*2 STEP PI/12
2150    IDRAW .2*COS(I),.2*SIN(I)
2160  NEXT I
2170   !
2180  MOVE 16,25
2190  CSIZE 2.3,.5
2200  LABEL "DATA  CLOCK1       DATA  CLOCK"
2210   !
2220  INPUT "Aer you ready ?  Press return key to start.",A
2230   !
2240  RETURN
2250   !
2260  END
```

EXECUTED RESULT

Output to GP-IB

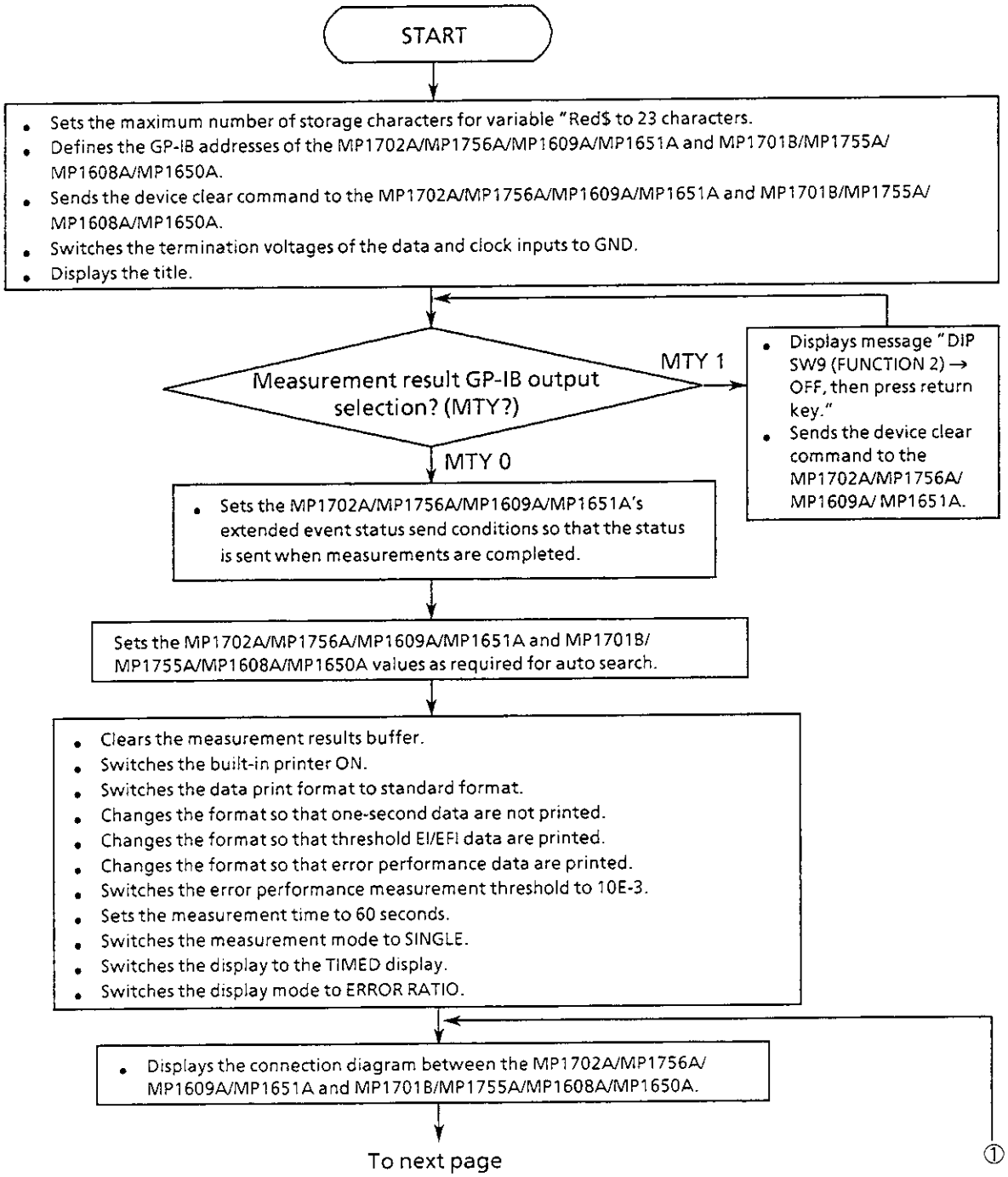
```
STA 90,02,26,11,59,00
STA 90,02,26,11,59,00
END 90,02,26,11,59,05
ELP      0,00,00,05
ER  1.0000E-04
EC      2500006
EI            5
EFI      0.0000
PFI           0
CLI           0
SLI           0
```

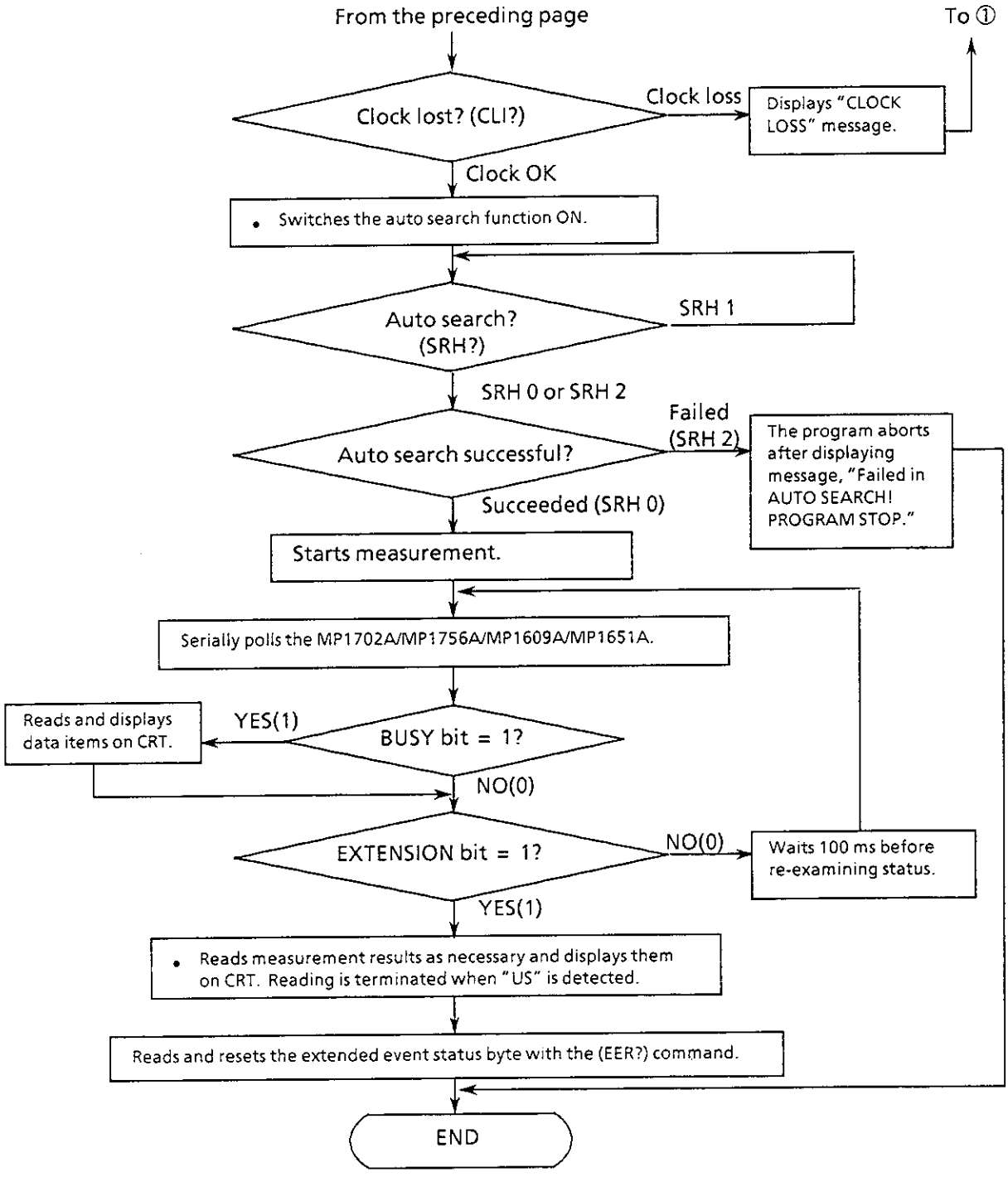
Printout from built-in printer

```
90-02-26/11:59:00 START

**********************
 START 90-02-26/11:59:00
 END   90-02-26/11:59:05
 ELP           0/00:00:05
 **ERROR MEASUREMENT**
 ERROR RATIO 1.0000E-04
 ERROR COUNT    2500006
 EI                   5
 EFI              0.0000%
 PFI                  0
 CLI                  0
 SLI                  0
**********************
```

(d) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Extended data (threshold measurement and error performance measurement data) are output in standard format at the end of measurement, as determined by serial polling.




From the preceding page
To ①
Clock lost? (CLI?)
Clock loss
Displays "CLOCK LOSS" message.
Clock OK
• Switches the auto search function ON.
Auto search? (SRH?)
SRH 1
SRH 0 or SRH 2
Auto search successful?
Failed (SRH 2)
The program aborts after displaying message, "Failed in AUTO SEARCH! PROGRAM STOP."
Succeeded (SRH 0)
Starts measurement.
Serially polls the MP1702A/MP1756A/MP1609A/MP1651A.
Reads and displays data items on CRT.
YES(1)
BUSY bit = 1?
NO(0)
EXTENSION bit = 1?
NO(0)
Waits 100 ms before re-examining status.
YES(1)
• Reads measurement results as necessary and displays them on CRT. Reading is terminated when "US" is detected.
Reads and resets the extended event status byte with the (EER?) command.
END

Program listing

```
10    !************************************************************
20    !*                                                          *
30    !*     MP1702A/MP1609A/MP1651A  MEASUREMENT  SAMPLE SOFT    *
40    !*        -- Standard format , Extension output --          *
50    !*        -- Threshold EI/EFI , Error performance --        *
60    !*                                                  ED_M4   *
70    !************************************************************
80    !
90    !-----------------------------------------------------------!
100   !                        MAIN ROUTINE                       !
110   !-----------------------------------------------------------!
120   !
130   DIM Red$[23]
140   LET Ad1=700               ! Set Device Address (PPG)
150   LET Ad2=701               ! Set Device Address (ED)
160   CLEAR Ad1                 ! Device Clear (PPG)
170   CLEAR Ad2                 ! Device Clear (ED)
180   OUTPUT Ad2;"DTM 0;CTM 0"
190    !
200   CLEAR SCREEN
210    !
220   PRINT "** MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT **"
230   PRINT
240    !
250   GOSUB M_type              ! Check Measurement type
260    !
270   OUTPUT Ad2;"EES 64"       ! EES : MEASURE bit
280    !
290   GOSUB D_set               ! Data setting
300   GOSUB Mode                ! Measurement mode
310   GOSUB Clock               ! Check clock loss
320   GOSUB Srch                ! Auto search : ON
330    !
340   OUTPUT Ad2;"STA"          ! Measurement Start
350    !
360   GOSUB S_poll              ! Serial polling-->Extension bit=1
370   GOSUB Result              ! Display Result
380    !
390   STOP
400    !
410    !-----------------------------------------------------------!
420    !                          SUB ROUTINE                      !
430    !-----------------------------------------------------------!
440    !
450 M_type: !------------------------------ Check Measurement type
460   !
470   LOOP
480      !
490     OUTPUT Ad2;"MTY?"
500     ENTER Ad2;Mty$
510      !
520     IF Mty$="MTY 1" THEN
530       INPUT "DIP SW 9 (Function 2)-->OFF, then press return key",A
540       CLEAR Ad2
550     END IF
560      !
570   EXIT IF Mty$="MTY 0"
580   END LOOP
590    !
600   RETURN
610    !
```

```
620 D_set:  !------------------------------------------ Set Data
630     !
640    OUTPUT Ad1;"LGC 0;PTN 2;MRK 1"
650    OUTPUT Ad2;"LGC 0;PTN 2;MRK 1"
660     !
670    OUTPUT Ad1;"CLK 1;RES 1;FRQ 3000"
680    OUTPUT Ad1;"OFS 0;SPD 0"
690    OUTPUT Ad1;"DAP 1;DOS 1"
700    OUTPUT Ad1;"CDL 100;CAP 1;COS 1"
710    OUTPUT Ad1;"EAD 1"
720     !
730    OUTPUT Ad2;"CPL 0"
740    OUTPUT Ad2;"SRH 0"
750    OUTPUT Ad2;"SYN 1"
760     !
770    RETURN
780     !
790 Mode: !------------------------------------ Measurement mode
800     !
810    OUTPUT Ad2;"BCL"            ! Measurement buffer clear
820    OUTPUT Ad2;"PRN 1"          ! Printer ON
830    OUTPUT Ad2;"FMT 0"          ! Data Output:Standard format
840    OUTPUT Ad2;"OSC 0"          ! One second data output : OFF
850    OUTPUT Ad2;"THR 1"          ! Threshold output : ON
860    OUTPUT Ad2;"EPF 1"          ! Error performance Output:ON
870    OUTPUT Ad2;"ETH 0"          ! Error performance threshold : 10E-3
880     !
890    OUTPUT Ad2;"PRD 0,0,1,0"   ! Measurement period : 60sec
900    OUTPUT Ad2;"MOD 1"          ! Measurement mode : SINGLE
910    OUTPUT Ad2;"TIM 3"          ! Display TIMED
920    OUTPUT Ad2;"DMS 0"          ! Display ERROR RATIO
930     !
940    RETURN
950     !
960     !
970 Clock: !------------------------------------ Check connection
980     !
990    LOOP
1000     GOSUB Connect             ! Display Device connection
1010      !
1020     OUTPUT Ad2;"CLI?"         ! REQUEST Clock loss ?
1030     ENTER Ad2;Cli$
1040      !
1050     IF Cli$="CLI 1" THEN
1060       PRINT "** CLOCK LOSS **"
1070     END IF
1080      !
1090   EXIT IF Cli$="CLI 0"
1100   END LOOP
1110    !
1120   RETURN
1130    !
1140    !
1150 Srch:!------------------------------------------ Auto search
1160       !
1170   OUTPUT Ad2;"SRH 1"                          ! AUTO SEARCH : ON
1180    !
1190   LOOP
1200      !
1210     OUTPUT Ad2;"SRH?"
1220     ENTER Ad2;Srh$
1230      !
1240   EXIT IF Srh$="SRH 0" OR Srh$="SRH 2"
1250   END LOOP
```

```
1260   !
1270   IF Srh$="SRH 2" THEN
1280     PRINT "** Failed in AUTO SEARCH !  PROGRAM STOP **"
1290     STOP
1300     !
1310   END IF
1320   !
1330   RETURN
1340   !
1350   !
1360 S_poll:!------------------------------------ Serial Polling
1370   !
1380   LOOP
1390     A=SPOLL(Ad2)                                ! Serial Polling
1400     !
1410     IF BIT(A,4)=1 THEN                          ! BUSY bit=1
1420       GOSUB Item_read
1430     END IF
1440     !
1450   EXIT IF BIT(A,1)=1                            ! EXTENSION bit=1
1460     WAIT .1
1470   END LOOP
1480   !
1490   RETURN
1500   !
1510   !
1520 Item_read: !-------------------------- Read & Print Item data
1530   !
1540   ENTER Ad2;Red$
1550   PRINT Red$
1560   !
1570   RETURN
1580   !
1590   !
1600 Result: !---------------------------------------- Print Result
1610   !
1620   LOOP
1630     ENTER Ad2;Red$
1640     PRINT Red$
1650   EXIT IF Red$[1,2]="US"
1660   END LOOP
1670   !
1680   OUTPUT Ad2;"EER?"                  ! RESET Extension register
1690   ENTER Ad2;Eer$
1700   !
1710   RETURN
1720   !
1730   !
1740 Connect: !-------------------------------- Display Connection
1750   !
1760   PEN 3
1770   VIEWPORT 70,140,50,100
1780   SHOW 0,70,0,50
1790   !
1800   CLIP 0,70,5,70
1810   FRAME
1820   !
1830   CSIZE 3,.4
1840   MOVE 25,45
1850   LABEL "<< CONNECTION >>"
1860   !
```

```
1870  CSIZE 3,.35
1880  MOVE 6,39
1890  LABEL "MP1701B/MP1608A/MP1650A     MP1702A/MP1609A/MP1651A"
1900   !
1910  MOVE 7,20
1920  RECTANGLE 25,18
1930   !
1940  MOVE 38,20
1950  RECTANGLE 25,18
1960   !
1970  MOVE 26,14
1980  IDRAW 0,9
1990   !
2000  FOR I=0 TO PI*2 STEP PI/12
2010    IDRAW .2*COS(I),.2*SIN(I)
2020  NEXT I
2030   !
2040  MOVE 26,14
2050  IDRAW 21,0
2060  IDRAW 0,9
2070   !
2080  FOR I=0 TO PI*2 STEP PI/12
2090    IDRAW .2*COS(I),.2*SIN(I)
2100  NEXT I
2110   !
2120  MOVE 21,17
2130  IDRAW 0,6
2140   !
2150  FOR I=0 TO PI*2 STEP PI/12
2160    IDRAW .2*COS(I),.2*SIN(I)
2170  NEXT I
2180   !
2190  MOVE 21,17
2200  IDRAW 21,0
2210  IDRAW 0,6
2220   !
2230  FOR I=0 TO PI*2 STEP PI/12
2240    IDRAW .2*COS(I),.2*SIN(I)
2250  NEXT I
2260   !
2270  MOVE 16,25
2280  CSIZE 2.3,.5
2290  LABEL "DATA  CLOCK1      DATA  CLOCK"
2300   !
2310  INPUT "Aer you ready ?  Press return key to start.",A
2320   !
2330  RETURN
2340   !
2350  END
```

EXECUTED RESULT

Output to GP-IB

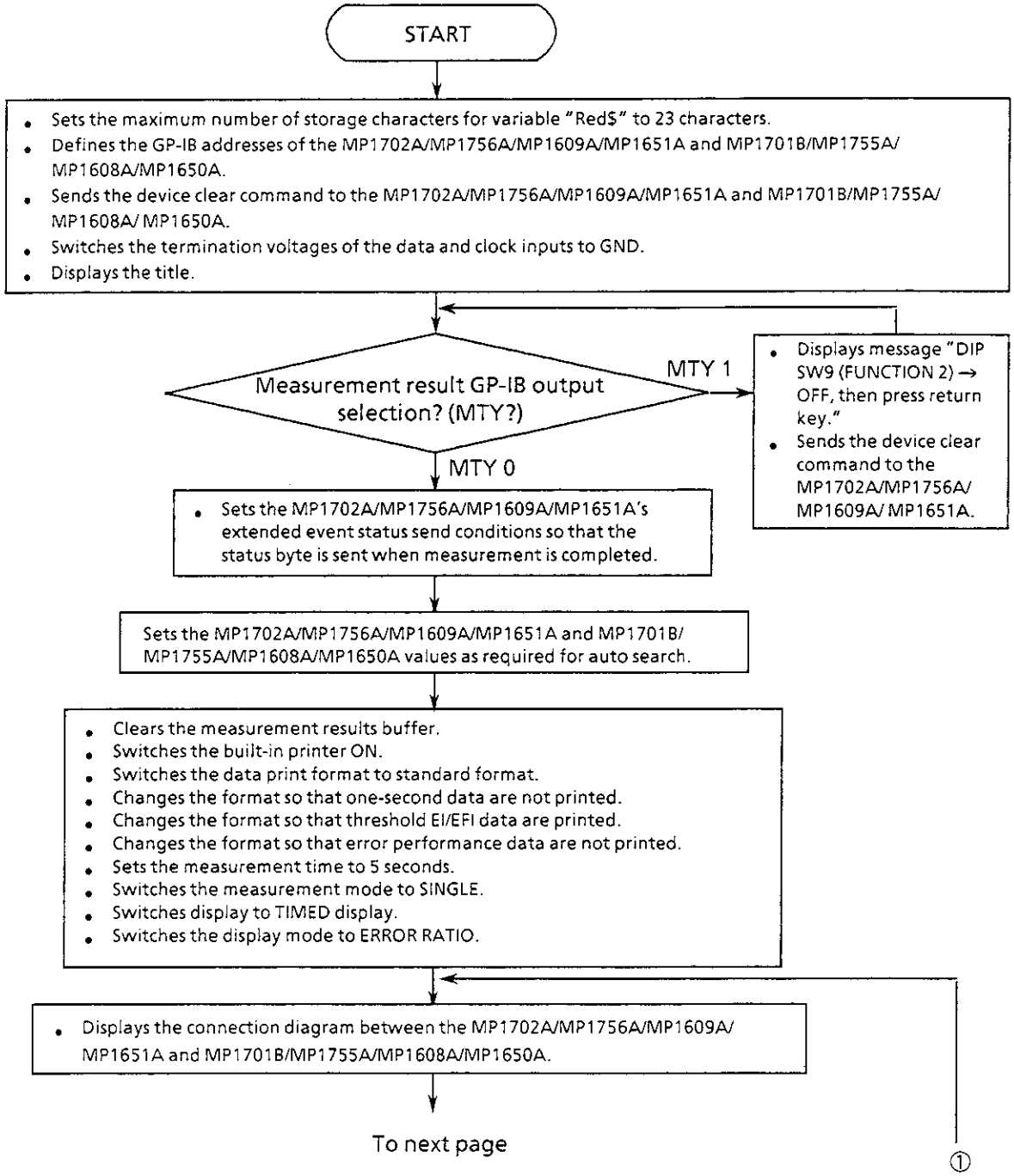
```
STA 90,02,26,11,55,08
STA 90,02,26,11,55,08
END 90,02,26,11,56,08
ELP      0,00,01,00
ER  1.0000E-04
EC     30000071
EI           60
EFI       0.0000
PFI            0
CLI            0
SLI            0
TH3            0,100.0000
TH4           60,   0.0000
TH5           60,   0.0000
TH6           60,   0.0000
TH7           60,   0.0000
TH8           60,   0.0000
TH9            0,100.0000
ES      100.0000
EFS       0.0000
SES       0.0000
DM      100.0000
US        0.0000
```

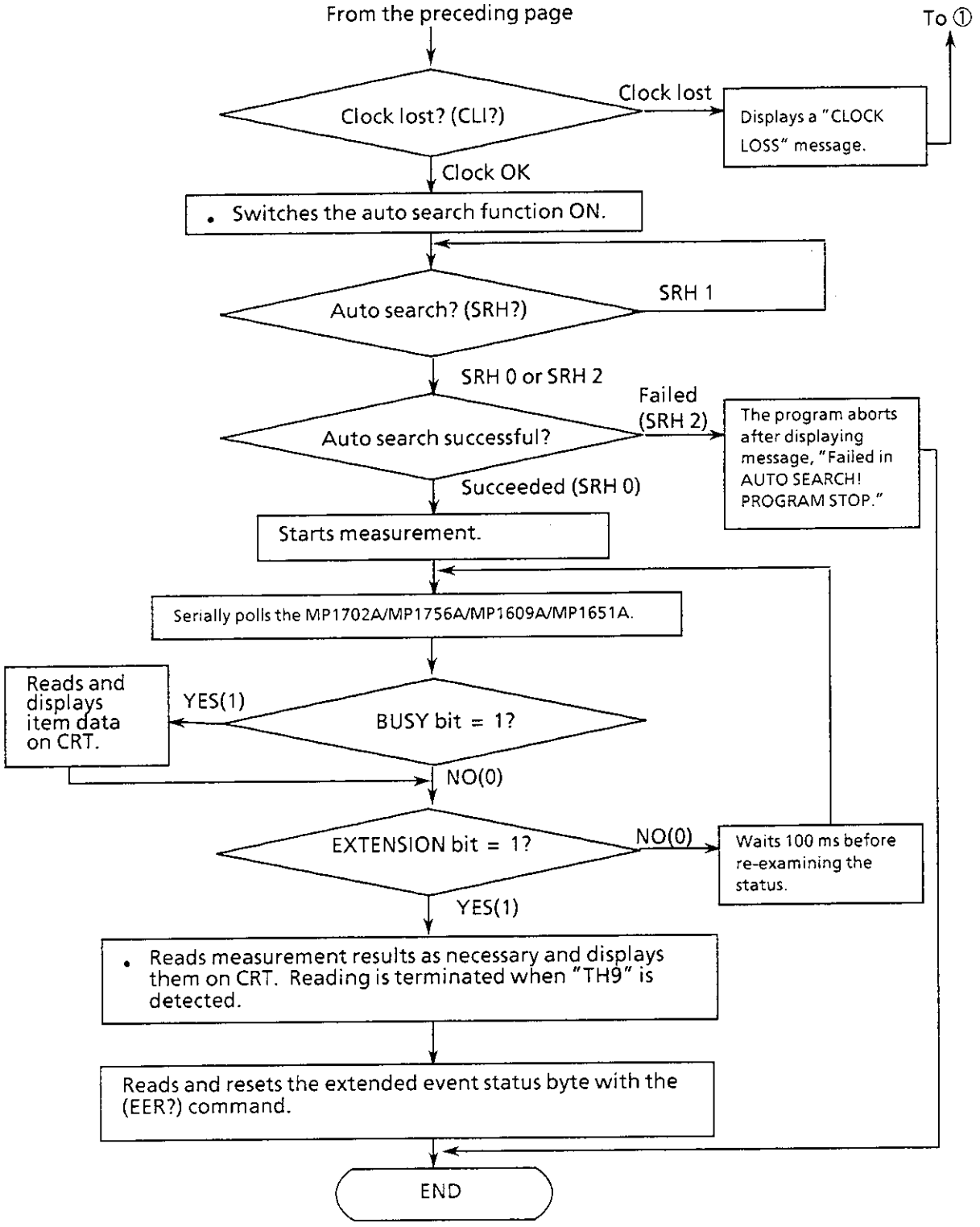
Printout from built-in printer

```
90-02-26/11:55:08 START

************************
 START 90-02-26/11:55:08
 END   90-02-26/11:56:08
 ELP            0/00:01:00
 ==ERROR MEASUREMENT==
 ERROR RATIO 1.0000E-04
 ERROR COUNT   30000071
 EI                   60
 EFI                0.0000%
 PFI                     0
 CLI                     0
 SLI                     0
 == THRESHOLD EI,EFI ==
 ER    EI        EFI
 >-3          0 100.0000%
 >-4         60   0.0000%
 >-5         60   0.0000%
 >-6         60   0.0000%
 >-7         60   0.0000%
 >-8         60   0.0000%
 =<-8         0 100.0000%
 ==ERROR PERFORMANCE==
 ERRORED SEC    100.0000%
 ER FREE SEC      0.0000%
 SES (E-3)        0.0000%
 DM  (E-6)      100.0000%
 UNAVAIL SEC      0.0000%
************************
```

(e) Measurement (Measurment type: Measurement results are output each time a measurement is completed)

Extended data (threshold measurement data) are output in standard format at the end of measurement, as determined by serial polling.




From the preceding page
To ①
Clock lost? (CLI?)
Clock lost
Displays a "CLOCK LOSS" message.
Clock OK
• Switches the auto search function ON.
Auto search? (SRH?)
SRH 1
SRH 0 or SRH 2
Auto search successful?
Failed (SRH 2)
The program aborts after displaying message, "Failed in AUTO SEARCH! PROGRAM STOP."
Succeeded (SRH 0)
Starts measurement.
Serially polls the MP1702A/MP1756A/MP1609A/MP1651A.
Reads and displays item data on CRT.
YES(1)
BUSY bit = 1?
NO(0)
EXTENSION bit = 1?
NO(0)
Waits 100 ms before re-examining the status.
YES(1)
• Reads measurement results as necessary and displays them on CRT. Reading is terminated when "TH9" is detected.
Reads and resets the extended event status byte with the (EER?) command.
END

Program listing

```
10    !*************************************************************
20    !*                                                           *
30    !*    MP1702A/MP1609A/MP1651A  MEASUREMENT  SAMPLE SOFT      *
40    !*       -- Standard format , Extension output --            *
50    !*                  -- Threshold EI/EFI --                   *
60    !*                                                  ED_M5    *
70    !*************************************************************
80    !
90    !-----------------------------------------------------------!
100   !                        MAIN ROUTINE                       !
110   !-----------------------------------------------------------!
120   !
130   DIM Red$[23]
140   LET Ad1=700               ! Set Device Address (PPG)
150   LET Ad2=701               ! Set Device Address (ED)
160   CLEAR Ad1                 ! Device Clear (PPG)
170   CLEAR Ad2                 ! Device Clear (ED)
180   OUTPUT Ad2;"DTM 0;CTM 0"
190    !
200   CLEAR SCREEN
210    !
220   PRINT "** MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT **"
230   PRINT
240    !
250   GOSUB M_type              ! Check Measurement type
260    !
270   OUTPUT Ad2;"EES 64"       ! EES : MEASURE bit
280    !
290   GOSUB D_set               ! Data setting
300   GOSUB Mode                ! Measurement mode
310   GOSUB Clock               ! Check clock loss
320   GOSUB Srch                ! Auto search : ON
330    !
340   OUTPUT Ad2;"STA"          ! Measurement Start
350    !
360   GOSUB S_poll              ! Serial polling-->Extension bit=1
370   GOSUB Result              ! Display Result
380    !
390   STOP
400    !
410    !-----------------------------------------------------------!
420    !                          SUB ROUTINE                      !
430    !-----------------------------------------------------------!
440    !
450 M_type: !------------------------------ Check Measurement type
460    !
470   LOOP
480      !
490     OUTPUT Ad2;"MTY?"
500     ENTER Ad2;Mty$
510      !
520     IF Mty$="MTY 1" THEN
530       INPUT "DIP SW 9 (Function 2)-->OFF, then press return key",A
540       CLEAR Ad2
550     END IF
560      !
570   EXIT IF Mty$="MTY 0"
580   END LOOP
590    !
600   RETURN
610    !
```

```
620 D_set:  !------------------------------------------ Set Data
630    !
640    OUTPUT Ad1;"LGC 0;PTN 2;MRK 1"
650    OUTPUT Ad2;"LGC 0;PTN 2;MRK 1"
660     !
670    OUTPUT Ad1;"CLK 1;RES 1;FRQ 3000"
680    OUTPUT Ad1;"OFS 0;SPD 0"
690    OUTPUT Ad1;"DAP 1;DOS 1"
700    OUTPUT Ad1;"CDL 100;CAP 1;COS 1"
710    OUTPUT Ad1;"EAD 1"
720     !
730    OUTPUT Ad2;"CPL 0"
740    OUTPUT Ad2;"SRH 0"
750    OUTPUT Ad2;"SYN 1"
760     !
770    RETURN
780     !
790 Mode: !------------------------------------ Measurement mode
800     !
810    OUTPUT Ad2;"BCL"           ! Measurement buffer clear
820    OUTPUT Ad2;"PRN 1"         ! Printer ON
830    OUTPUT Ad2;"FMT 0"         ! Data Output:Standard format
840    OUTPUT Ad2;"OSC 0"         ! One second data output : OFF
850    OUTPUT Ad2;"THR 1"         ! Threshold output : ON
860    OUTPUT Ad2;"EPF 0"         ! Error performance Output:OFF
870     !
880    OUTPUT Ad2;"PRD 0,0,0,5"   ! Measurement period : 5sec
890    OUTPUT Ad2;"MOD 1"         ! Measurement mode : SINGLE
900    OUTPUT Ad2;"TIM 3"         ! Display TIMED
910    OUTPUT Ad2;"DMS 0"         ! Display ERROR RATIO
920     !
930    RETURN
940     !
950     !
960 Clock: !--------------------------- Check connection,Set data
970     !
980    LOOP
990      GOSUB Connect            ! Display Device connection
1000      !
1010     OUTPUT Ad2;"CLI?"        ! REQUEST Clock loss ?
1020     ENTER Ad2;Cli$
1030      !
1040     IF Cli$="CLI 1" THEN
1050       PRINT "** CLOCK LOSS **"
1060     END IF
1070      !
1080   EXIT IF Cli$="CLI 0"
1090   END LOOP
1100    !
1110   RETURN
1120    !
1130    !
1140 Srch:!------------------------------------------ Auto search
1150      !
1160   OUTPUT Ad2;"SRH 1"                          ! AUTO SEARCH : ON
1170    !
1180   LOOP
1190      !
1200     OUTPUT Ad2;"SRH?"
1210     ENTER Ad2;Srh$
1220      !
1230   EXIT IF Srh$="SRH 0" OR Srh$="SRH 2"
1240   END LOOP
```

```
1250   !
1260  IF Srh$="SRH 2" THEN
1270    PRINT "** Failed in AUTO SEARCH !  PROGRAM STOP **"
1280    STOP
1290     !
1300  END IF
1310   !
1320  RETURN
1330   !
1340   !
1350 S_poll:!------------------------------------ Serial Polling
1360   !
1370  LOOP
1380    A=SPOLL(Ad2)                                ! Serial Polling
1390     !
1400    IF BIT(A,4)=1 THEN                          ! BUSY bit=1
1410      GOSUB Item_read
1420    END IF
1430     !
1440  EXIT IF BIT(A,1)=1                            ! EXTENSION bit=1
1450    WAIT .1
1460  END LOOP
1470   !
1480  RETURN
1490   !
1500   !
1510 Item_read: !------------------------- Read & Print Item data
1520   !
1530  ENTER Ad2;Red$
1540  PRINT Red$
1550   !
1560  RETURN
1570   !
1580   !
1590 Result: !------------------------------------- Print Result
1600   !
1610  LOOP
1620    ENTER Ad2;Red$
1630    PRINT Red$
1640  EXIT IF Red$[1,3]="TH9"
1650  END LOOP
1660   !
1670  OUTPUT Ad2;"EER?"                  ! RESET Extension register
1680  ENTER Ad2;Eer$
1690   !
1700  RETURN
1710   !
1720   !
1730 Connect: !------------------------------- Display Connection
1740   !
1750  PEN 3
1760  VIEWPORT 70,140,50,100
1770  SHOW 0,70,0,50
1780   !
1790  CLIP 0,70,5,70
1800  FRAME
1810   !
1820  CSIZE 3,.4
1830  MOVE 25,45
1840  LABEL "<< CONNECTION >>"
1850   !
1860  CSIZE 3,.35
1870  MOVE 6,39
1880  LABEL "MP1701B/MP1608A/MP1650A     MP1702A/MP1609A/MP1651A"
```

```
1890  !
1900  MOVE 7,20
1910  RECTANGLE 25,18
1920   !
1930  MOVE 38,20
1940  RECTANGLE 25,18
1950   !
1960  MOVE 26,14
1970  IDRAW 0,9
1980   !
1990  FOR I=0 TO PI*2 STEP PI/12
2000    IDRAW .2*COS(I),.2*SIN(I)
2010  NEXT I
2020   !
2030  MOVE 26,14
2040  IDRAW 21,0
2050  IDRAW 0,9
2060   !
2070  FOR I=0 TO PI*2 STEP PI/12
2080    IDRAW .2*COS(I),.2*SIN(I)
2090  NEXT I
2100   !
2110  MOVE 21,17
2120  IDRAW 0,6
2130   !
2140  FOR I=0 TO PI*2 STEP PI/12
2150    IDRAW .2*COS(I),.2*SIN(I)
2160  NEXT I
2170   !
2180  MOVE 21,17
2190  IDRAW 21,0
2200  IDRAW 0,6
2210   !
2220  FOR I=0 TO PI*2 STEP PI/12
2230    IDRAW .2*COS(I),.2*SIN(I)
2240  NEXT I
2250   !
2260  MOVE 16,25
2270  CSIZE 2.3,.5
2280  LABEL "DATA  CLOCK1       DATA  CLOCK"
2290   !
2300  INPUT "Aer you ready ?  Press return key to start.",A
2310   !
2320  RETURN
2330   !
2340  END
```

EXECUTED RESULT

Output to GP-IB

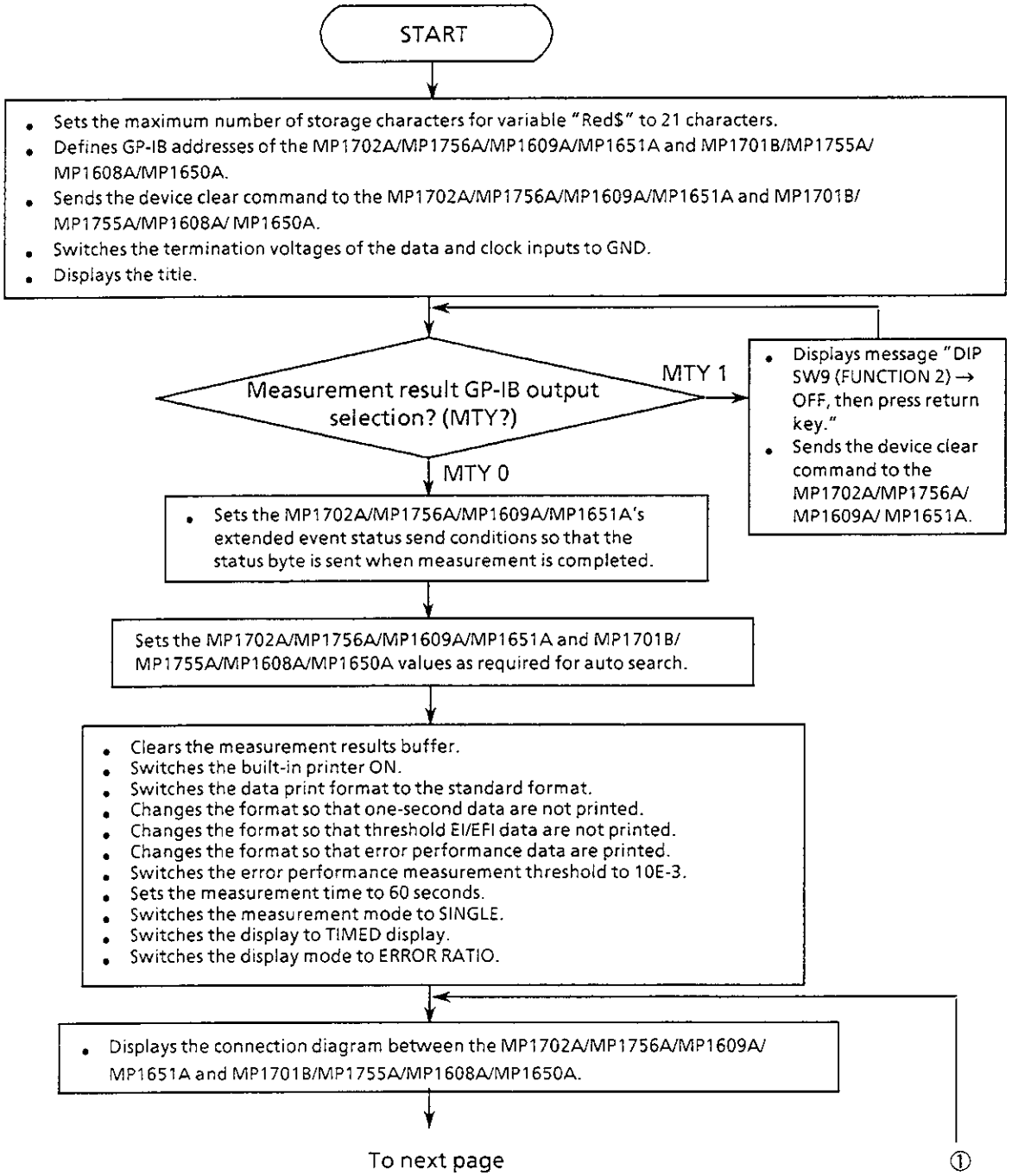
```
STA 90,02,26,11,56,44
STA 90,02,26,11,56,44
END 90,02,26,11,56,49
ELP      0,00,00,05
ER   1.0000E-04
EC      2500005
EI            5
EFI      0.0000
PFI           0
CLI           0
SLI           0
TH3           0,100.0000
TH4           5,   0.0000
TH5           5,   0.0000
TH6           5,   0.0000
TH7           5,   0.0000
TH8           5,   0.0000
TH9           0,100.0000
```

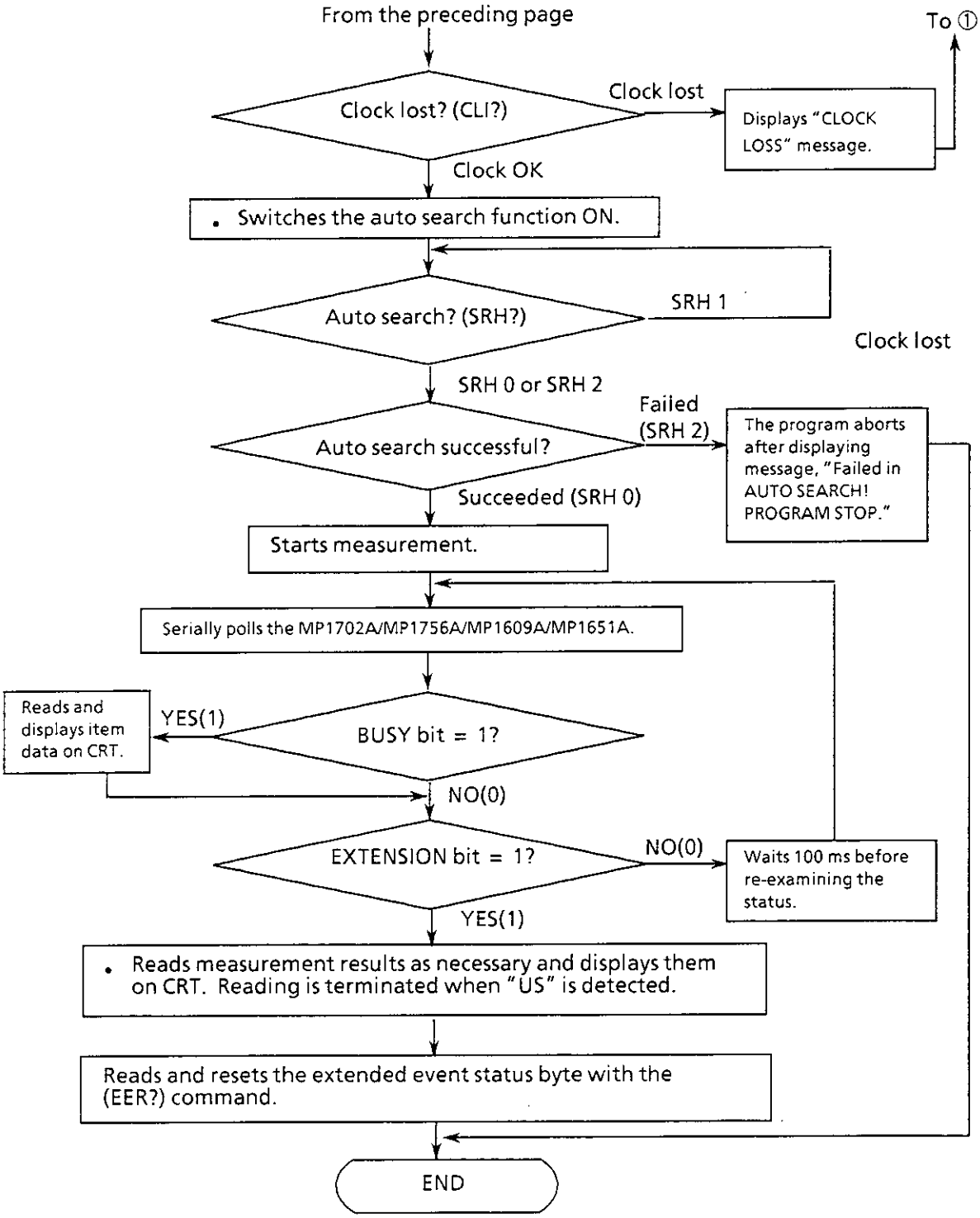
Printout from built-in printer

```
90-02-26/11:56:44 START

**************************
 START 90-02-26/11:56:44
 END   90-02-26/11:56:49
 ELP             0/00:00:05
 ==ERROR MEASUREMENT==
 ERROR RATIO 1.0000E-04
 ERROR COUNT    2500005
 EI                   5
 EFI                0.0000%
 PFI                  0
 CLI                  0
 SLI                  0
 == THRESHOLD EI,EFI ==
 ER    EI        EFI
 >-3          0 100.0000%
 >-4          5   0.0000%
 >-5          5   0.0000%
 >-6          5   0.0000%
 >-7          5   0.0000%
 >-8          5   0.0000%
 =<-8         0 100.0000%
**************************
```

(f) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Extended data (error performance measurement data) are output in standard format at the end of measurement, as determined by serial polling.




From the preceding page
To ①
Clock lost? (CLI?)
Clock lost
Displays "CLOCK LOSS" message.
Clock OK
• Switches the auto search function ON.
Auto search? (SRH?)
SRH 1
Clock lost
SRH 0 or SRH 2
Auto search successful?
Failed (SRH 2)
The program aborts after displaying message, "Failed in AUTO SEARCH! PROGRAM STOP."
Succeeded (SRH 0)
Starts measurement.
Serially polls the MP1702A/MP1756A/MP1609A/MP1651A.
Reads and displays item data on CRT.
YES(1)
BUSY bit = 1?
NO(0)
EXTENSION bit = 1?
NO(0)
Waits 100 ms before re-examining the status.
YES(1)
• Reads measurement results as necessary and displays them on CRT. Reading is terminated when "US" is detected.
Reads and resets the extended event status byte with the (EER?) command.
END

Program listing

```
10    !*************************************************************
20    !*                                                           *
30    !*     MP1702A/MP1609A/MP1651A  MEASUREMENT  SAMPLE SOFT     *
40    !*        -- Standard format , Extension output --           *
50    !*                  -- Error performance --                  *
60    !*                                                  ED_M6    *
70    !*************************************************************
80    !
90    !-----------------------------------------------------------!
100   !                         MAIN ROUTINE                      !
110   !-----------------------------------------------------------!
120   !
130   DIM Red$[21]
140   LET Ad1=700               ! Set Device Address (PPG)
150   LET Ad2=701               ! Set Device Address (ED)
160   CLEAR Ad1                 ! Device Clear (PPG)
170   CLEAR Ad2                 ! Device Clear (ED)
180   OUTPUT Ad2;"DTM 0;CTM 0"
190    !
200   CLEAR SCREEN
210    !
220   PRINT "** MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT **"
230   PRINT
240    !
250   GOSUB M_type              ! Check Measurement type
260    !
270   OUTPUT Ad2;"EES 64"       ! EES : MEASURE bit
280    !
290   GOSUB D_set               ! Data setting
300   GOSUB Mode                ! Measurement mode
310   GOSUB Clock               ! Check clock loss
320   GOSUB Srch                ! Auto search : ON
330    !
340   OUTPUT Ad2;"STA"          ! Measurement Start
350    !
360   GOSUB S_poll              ! Serial polling-->Extension bit=1
370   GOSUB Result              ! Display Result
380    !
390   STOP
400    !
410    !-----------------------------------------------------------!
420    !                          SUB ROUTINE                      !
430    !-----------------------------------------------------------!
440    !
450 M_type: !----------------------------- Check Measurement type
460   !
470   LOOP
480      !
490     OUTPUT Ad2;"MTY?"
500     ENTER Ad2;Mty$
510      !
520     IF Mty$="MTY 1" THEN
530       INPUT "DIP SW 9 (Function 2)-->OFF, then press return key",A
540       CLEAR Ad2
550     END IF
560      !
570   EXIT IF Mty$="MTY 0"
580   END LOOP
590    !
600   RETURN
610    !
```

```
620 D_set:  !------------------------------------------ Set Data
630     !
640    OUTPUT Ad1;"LGC 0;PTN 2;MRK 1"
650    OUTPUT Ad2;"LGC 0;PTN 2;MRK 1"
660     !
670    OUTPUT Ad1;"CLK 1;RES 1;FRQ 3000"
680    OUTPUT Ad1;"OFS 0;SPD 0"
690    OUTPUT Ad1;"DAP 1;DOS 1"
700    OUTPUT Ad1;"CDL 100;CAP 1;COS 1"
710    OUTPUT Ad1;"EAD 1"
720     !
730    OUTPUT Ad2;"CPL 0"
740    OUTPUT Ad2;"SRH 0"
750    OUTPUT Ad2;"SYN 1"
760     !
770    RETURN
780     !
790 Mode: !------------------------------------ Measurement mode
800     !
810    OUTPUT Ad2;"BCL"           ! Measurement buffer clear
820    OUTPUT Ad2;"PRN 1"         ! Printer ON
830    OUTPUT Ad2;"FMT 0"         ! Data Output:Standard format
840    OUTPUT Ad2;"OSC 0"         ! One second data output : OFF
850    OUTPUT Ad2;"THR 0"         ! Threshold EI/EFI : OFF
860    OUTPUT Ad2;"EPF 1"         ! Error performance Output:ON
870    OUTPUT Ad2;"ETH 0"         ! Error performance threshold : 10E-3
880     !
890    OUTPUT Ad2;"PRD 0,0,1,0"   ! Measurement period : 60sec
900    OUTPUT Ad2;"MOD 1"         ! Measurement mode : SINGLE
910    OUTPUT Ad2;"TIM 3"         ! Display TIMED
920    OUTPUT Ad2;"DMS 0"         ! Display ERROR RATIO
930     !
940    RETURN
950     !
960     !
970 Clock: !------------------------------------ Check connection
980     !
990    LOOP
1000     GOSUB Connect            ! Display Device connection
1010      !
1020     OUTPUT Ad2;"CLI?"        ! REQUEST Clock loss ?
1030     ENTER Ad2;Cli$
1040      !
1050     IF Cli$="CLI 1" THEN
1060       PRINT "** CLOCK LOSS **"
1070     END IF
1080      !
1090   EXIT IF Cli$="CLI 0"
1100   END LOOP
1110    !
1120   RETURN
1130    !
1140    !
1150 Srch:!------------------------------------------ Auto search
1160       !
1170   OUTPUT Ad2;"SRH 1"                          ! AUTO SEARCH : ON
1180    !
1190   LOOP
1200      !
1210     OUTPUT Ad2;"SRH?"
1220     ENTER Ad2;Srh$
1230      !
1240   EXIT IF Srh$="SRH 0" OR Srh$="SRH 2"
1250   END LOOP
```

```
1260   !
1270   IF Srh$="SRH 2" THEN
1280     PRINT "** Failed in AUTO SEARCH !  PROGRAM STOP **"
1290     STOP
1300     !
1310   END IF
1320   !
1330   RETURN
1340   !
1350   !
1360 S_poll:!------------------------------------- Serial Polling
1370   !
1380   LOOP
1390     A=SPOLL(Ad2)                                ! Serial Polling
1400     !
1410     IF BIT(A,4)=1 THEN                          ! BUSY bit=1
1420       GOSUB Item_read
1430     END IF
1440     !
1450   EXIT IF BIT(A,1)=1                            ! EXTENSION bit=1
1460     WAIT .1
1470   END LOOP
1480   !
1490   RETURN
1500   !
1510   !
1520 Item_read: !-------------------------- Read & Print Item data
1530   !
1540   ENTER Ad2;Red$
1550   PRINT Red$
1560   !
1570   RETURN
1580   !
1590   !
1600 Result: !------------------------------------- Print Result
1610   !
1620   LOOP
1630     ENTER Ad2;Red$
1640     PRINT Red$
1650   EXIT IF Red$[1,2]="US"
1660   END LOOP
1670   !
1680   OUTPUT Ad2;"EER?"                  ! RESET Extension register
1690   ENTER Ad2;Eer$
1700   !
1710   RETURN
1720   !
1730   !
1740 Connect: !------------------------------- Display Connection
1750   !
1760   PEN 3
1770   VIEWPORT 70,140,50,100
1780   SHOW 0,70,0,50
1790   !
1800   CLIP 0,70,5,70
1810   FRAME
1820   !
1830   CSIZE 3,.4
1840   MOVE 25,45
1850   LABEL "<< CONNECTION >>"
1860   !
1870   CSIZE 3,.35
1880   MOVE 6,39
1890   LABEL "MP1701B/MP1608A/MP1650A    MP1702A/MP1609A/MP1651A"
```

```
1900  !
1910  MOVE 7,20
1920  RECTANGLE 25,18
1930   !
1940  MOVE 38,20
1950  RECTANGLE 25,18
1960   !
1970  MOVE 26,14
1980  IDRAW 0,9
1990   !
2000  FOR I=0 TO PI*2 STEP PI/12
2010    IDRAW .2*COS(I),.2*SIN(I)
2020  NEXT I
2030   !
2040  MOVE 26,14
2050  IDRAW 21,0
2060  IDRAW 0,9
2070   !
2080  FOR I=0 TO PI*2 STEP PI/12
2090    IDRAW .2*COS(I),.2*SIN(I)
2100  NEXT I
2110   !
2120  MOVE 21,17
2130  IDRAW 0,6
2140   !
2150  FOR I=0 TO PI*2 STEP PI/12
2160    IDRAW .2*COS(I),.2*SIN(I)
2170  NEXT I
2180   !
2190  MOVE 21,17
2200  IDRAW 21,0
2210  IDRAW 0,6
2220   !
2230  FOR I=0 TO PI*2 STEP PI/12
2240    IDRAW .2*COS(I),.2*SIN(I)
2250  NEXT I
2260   !
2270  MOVE 16,25
2280  CSIZE 2.3,.5
2290  LABEL "DATA  CLOCK1       DATA  CLOCK"
2300   !
2310  INPUT "Aer you ready ?  Press return key to start.",A
2320   !
2330  RETURN
2340   !
2350  END
```

EXECUTED RESULT

Output to GP-IB

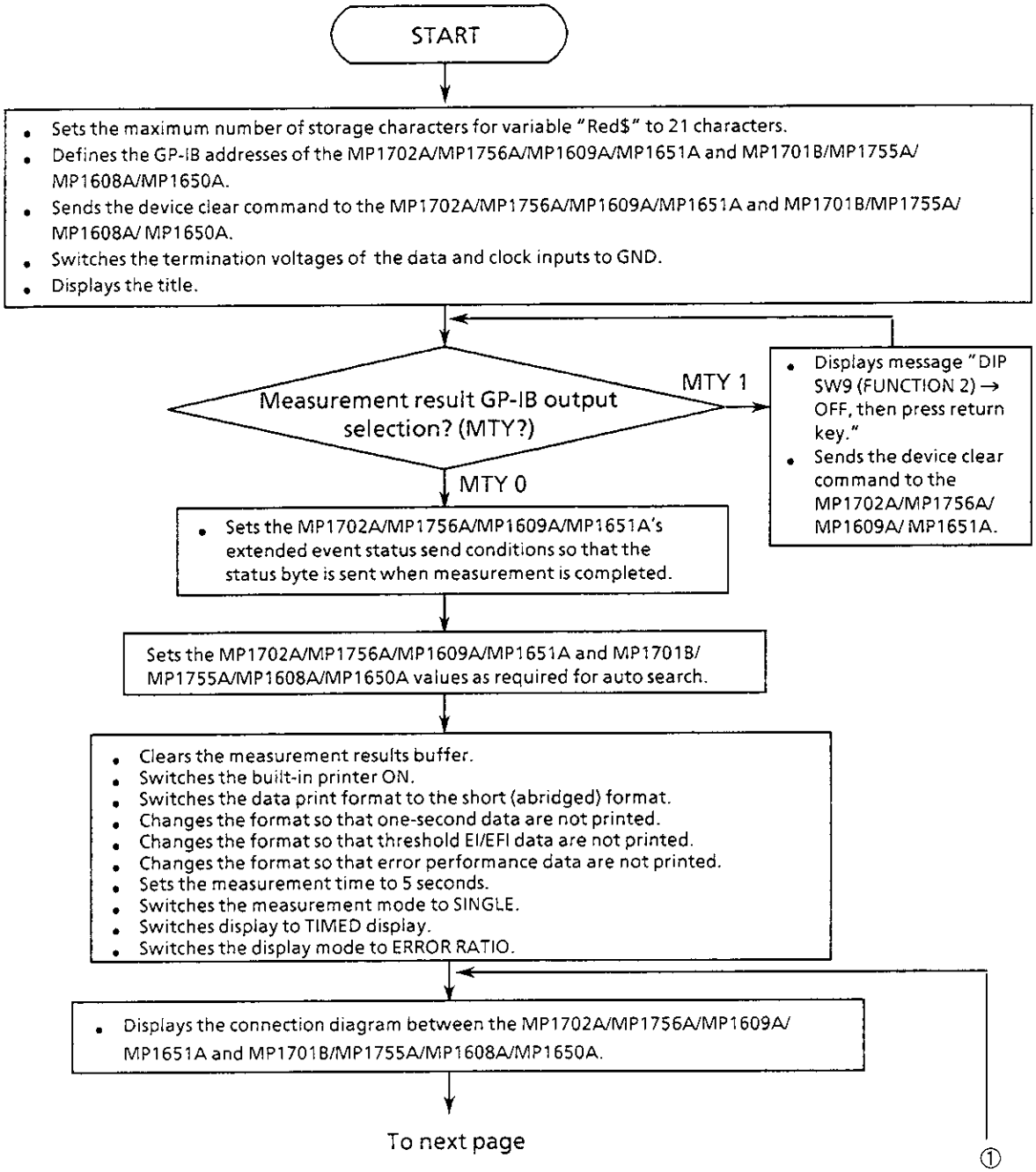
```
STA 90,02,26,11,59,37
STA 90,02,26,11,59,37
END 90,02,26,12,00,37
ELP      0,00,01,00
ER  1.0000E-04
EC    30000069
EI          60
EFI     0.0000
PFI          0
CLI          0
SLI          0
ES    100.0000
EFS     0.0000
SES     0.0000
DM    100.0000
US      0.0000
```

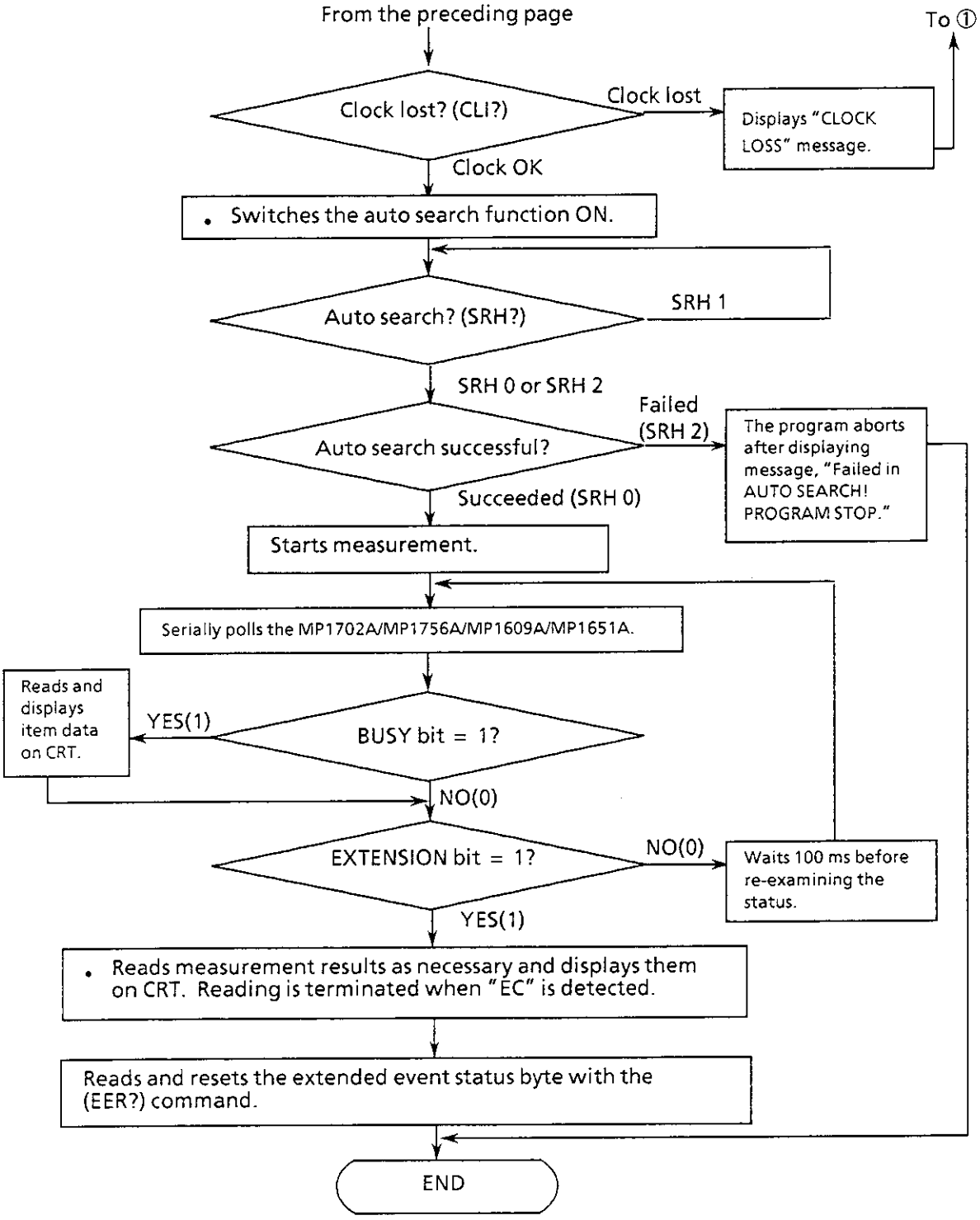
Printout from built-in printer

```
90-02-26/11:59:37 START

**********************
 START 90-02-26/11:59:37
 END   90-02-26/12:00:37
 ELP           0/00:01:00
 **ERROR MEASUREMENT**
 ERROR RATIO 1.0000E-04
 ERROR COUNT   30000069
 EI                  60
 EFI               0.0000%
 PFI                  0
 CLI                  0
 SLI                  0
 **ERROR PERFORMANCE**
 ERRORED SEC    100.0000%
 ER FREE SEC      0.0000%
 SES (E-3)        0.0000%
 DM  (E-6)      100.0000%
 UNAVAIL SEC      0.0000%
**********************
```

(g) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Standard data are output in short (abridged) format at the end of measurement, as determined by serial polling.




From the preceding page
To ①
Clock lost? (CLI?)
Clock lost
Displays "CLOCK LOSS" message.
Clock OK
• Switches the auto search function ON.
Auto search? (SRH?)
SRH 1
SRH 0 or SRH 2
Auto search successful?
Failed (SRH 2)
The program aborts after displaying message, "Failed in AUTO SEARCH! PROGRAM STOP."
Succeeded (SRH 0)
Starts measurement.
Serially polls the MP1702A/MP1756A/MP1609A/MP1651A.
Reads and displays item data on CRT.
YES(1)
BUSY bit = 1?
NO(0)
EXTENSION bit = 1?
NO(0)
Waits 100 ms before re-examining the status.
YES(1)
• Reads measurement results as necessary and displays them on CRT. Reading is terminated when "EC" is detected.
Reads and resets the extended event status byte with the (EER?) command.
END

Program listing

```
10    !*************************************************************
20    !*                                                           *
30    !*     MP1702A/MP1609A/MP1651A  MEASUREMENT  SAMPLE SOFT     *
40    !*          -- Abridged format , Standard output --          *
50    !*          -- Measurement end? >> Extension bit --          *
60    !*                                                  ED_M7    *
70    !*************************************************************
80    !
90    !-----------------------------------------------------------!
100   !                        MAIN ROUTINE                       !
110   !-----------------------------------------------------------!
120   !
130   DIM Red$[21]
140   LET Ad1=700               ! Set Device Address (PPG)
150   LET Ad2=701               ! Set Device Address (ED)
160   CLEAR Ad1                 ! Device Clear (PPG)
170   CLEAR Ad2                 ! Device Clear (ED)
180   OUTPUT Ad2;"DTM 0;CTM 0"
190    !
200   CLEAR SCREEN
210    !
220   PRINT "** MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT **"
230   PRINT
240    !
250   GOSUB M_type              ! Check Measurement type
260    !
270   OUTPUT Ad2;"EES 64"       ! EES : MEASURE bit
280    !
290   GOSUB D_set               ! Data setting
300   GOSUB Mode                ! Measurement mode
310   GOSUB Clock               ! Check clock loss
320   GOSUB Srch                ! Auto search : ON
330    !
340   OUTPUT Ad2;"STA"          ! Measurement Start
350    !
360   GOSUB S_poll              ! Serial polling-->Extension bit=1
370   GOSUB Result              ! Display Result
380    !
390   STOP
400    !
410    !-----------------------------------------------------------!
420    !                          SUB ROUTINE                      !
430    !-----------------------------------------------------------!
440    !
450 M_type: !---------------------------- Check Measurement type
460   !
470   LOOP
480      !
490     OUTPUT Ad2;"MTY?"
500     ENTER Ad2;Mty$
510      !
520     IF Mty$="MTY 1" THEN
530       INPUT "DIP SW 9 (Function 2)-->OFF, then press return key",A
540       CLEAR Ad2
550     END IF
560      !
570   EXIT IF Mty$="MTY 0"
580   END LOOP
590    !
600   RETURN
610    !
```

```
620 D_set:  !------------------------------------------ Set Data
630    !
640    OUTPUT Ad1;"LGC 0;PTN 2;MRK 1"
650    OUTPUT Ad2;"LGC 0;PTN 2;MRK 1"
660     !
670    OUTPUT Ad1;"CLK 1;RES 1;FRQ 3000"
680    OUTPUT Ad1;"OFS 0;SPD 0"
690    OUTPUT Ad1;"DAP 1;DOS 1"
700    OUTPUT Ad1;"CDL 100;CAP 1;COS 1"
710    OUTPUT Ad1;"EAD 1"
720     !
730    OUTPUT Ad2;"CPL 0"
740    OUTPUT Ad2;"SRH 0"
750    OUTPUT Ad2;"SYN 1"
760     !
770    RETURN
780     !
790 Mode: !------------------------------------ Measurement mode
800     !
810    OUTPUT Ad2;"BCL"          ! Measurement buffer clear
820    OUTPUT Ad2;"PRN 1"        ! Printer ON
830    OUTPUT Ad2;"FMT 1"        ! Data Output: Abridged format
840    OUTPUT Ad2;"OSC 0"        ! One second data output : OFF
850    OUTPUT Ad2;"THR 0"        ! Threshold output : OFF
860    OUTPUT Ad2;"EPF 0"        ! Error performance Output:OFF
870     !
880    OUTPUT Ad2;"PRD 0,0,0,5" ! Measurement period : 5sec
890    OUTPUT Ad2;"MOD 1"        ! Measurement mode : SINGLE
900    OUTPUT Ad2;"TIM 3"        ! Display TIMED
910    OUTPUT Ad2;"DMS 0"        ! Display ERROR RATIO
920     !
930    RETURN
940     !
950     !
960 Clock: !---------------------------- Check connection,Set data
970     !
980    LOOP
990      GOSUB Connect            ! Display Device connection
1000      !
1010     OUTPUT Ad2;"CLI?"        ! REQUEST Clock loss ?
1020     ENTER Ad2;Cli$
1030      !
1040     IF Cli$="CLI 1" THEN
1050       PRINT "** CLOCK LOSS **"
1060     END IF
1070      !
1080   EXIT IF Cli$="CLI 0"
1090   END LOOP
1100    !
1110   RETURN
1120    !
1130    !
1140 Srch:!------------------------------------------ Auto search
1150       !
1160   OUTPUT Ad2;"SRH 1"                          ! AUTO SEARCH : ON
1170    !
1180   LOOP
1190      !
1200     OUTPUT Ad2;"SRH?"
1210     ENTER Ad2;Srh$
1220      !
1230   EXIT IF Srh$="SRH 0" OR Srh$="SRH 2"
1240   END LOOP
1250    !
```

```
1260  IF Srh$="SRH 2" THEN
1270    PRINT "** Failed in AUTO SEARCH !  PROGRAM STOP **"
1280    STOP
1290     !
1300  END IF
1310   !
1320  RETURN
1330   !
1340   !
1350 S_poll:!------------------------------------ Serial Polling
1360   !
1370  LOOP
1380    A=SPOLL(Ad2)                                ! Serial Polling
1390     !
1400    IF BIT(A,4)=1 THEN                          ! BUSY bit=1
1410      GOSUB Item_read
1420    END IF
1430     !
1440  EXIT IF BIT(A,1)=1                            ! EXTENSION bit=1
1450    WAIT .1
1460  END LOOP
1470   !
1480  RETURN
1490   !
1500   !
1510 Item_read: !------------------------- Read & Print Item data
1520   !
1530  ENTER Ad2;Red$
1540  PRINT Red$
1550   !
1560  RETURN
1570   !
1580   !
1590 Result: !---------------------------------------- Print Result
1600   !
1610  LOOP
1620    ENTER Ad2;Red$
1630    PRINT Red$
1640  EXIT IF Red$[1,2]="EC"
1650  END LOOP
1660   !
1670  OUTPUT Ad2;"EER?"                  ! RESET Extension register
1680  ENTER Ad2;Eer$
1690   !
1700  RETURN
1710   !
1720   !
1730 Connect: !-------------------------------- Display Connection
1740   !
1750  PEN 3
1760  VIEWPORT 70,140,50,100
1770  SHOW 0,70,0,50
1780   !
1790  CLIP 0,70,5,70
1800  FRAME
1810   !
1820  CSIZE 3,.4
1830  MOVE 25,45
1840  LABEL "<< CONNECTION >>"
1850   !
1860  CSIZE 3,.35
1870  MOVE 6,39
1880  LABEL "MP1701B/MP1608A/MP1650A     MP1702A/MP1609A/MP1651A"
1890  !
```

```
1900  MOVE 7,20
1910  RECTANGLE 25,18
1920   !
1930  MOVE 38,20
1940  RECTANGLE 25,18
1950   !
1960  MOVE 26,14
1970  IDRAW 0,9
1980   !
1990  FOR I=0 TO PI*2 STEP PI/12
2000    IDRAW .2*COS(I),.2*SIN(I)
2010  NEXT I
2020   !
2030  MOVE 26,14
2040  IDRAW 21,0
2050  IDRAW 0,9
2060   !
2070  FOR I=0 TO PI*2 STEP PI/12
2080    IDRAW .2*COS(I),.2*SIN(I)
2090  NEXT I
2100   !
2110  MOVE 21,17
2120  IDRAW 0,6
2130   !
2140  FOR I=0 TO PI*2 STEP PI/12
2150    IDRAW .2*COS(I),.2*SIN(I)
2160  NEXT I
2170   !
2180  MOVE 21,17
2190  IDRAW 21,0
2200  IDRAW 0,6
2210   !
2220  FOR I=0 TO PI*2 STEP PI/12
2230    IDRAW .2*COS(I),.2*SIN(I)
2240  NEXT I
2250   !
2260  MOVE 16,25
2270  CSIZE 2.3,.5
2280  LABEL "DATA  CLOCK1       DATA  CLOCK"
2290   !
2300  INPUT "Aer you ready ?  Press return key to start.",A
2310   !
2320  RETURN
2330   !
2340  END
```

EXECUTED RESULT

Output to GP-IB

```
STA 90,02,26,12,01,21
END 90,02,26,12,01,26
ER   1.0000E-04
EC      2500006
```

Printout from built-in printer

```
90-02-26/12:01:21 START

 END   90-02-26/12:01:26
 ER 1.00E-04 EC  2500006
```

(5) Measurement (Measurement type: Outputs results each time a measurement is completed)

Intermediate measurement results are output each time an (IMD?) request command is issued.

The following programs, (a) to (e), are created for measurement where the measurement result GP-IB output selection is set for output to be made each time a measurement is completed.

Intermediate results are output each time an (IMD?) command is issued.

In these programs, measurement is initiated after connecting DATA to DATA and CLOCK1 to CLOCK between the MP1702A/MP1756A/MP1609A/MP1651A and MP1701B/MP1755A/ MP1608A/ MP1650A.

First, the program initializes conditions for auto search and for the appropriate measurement.

Next, it confirms that the clock is not lost and, after confirmation, switches the auto search function ON.

If auto search is successfully completed, measurement is begun.

Whenever the [RETURN] is pressed, the (IMD?) request command is issued, and measurement results are read and displayed on the CRT. The data read terminate conditions depend on the print format.

If auto search fails, the program is aborted.

Table 15-4 shows methods for determining the measurement result print format and the whether one-second data will be printed.

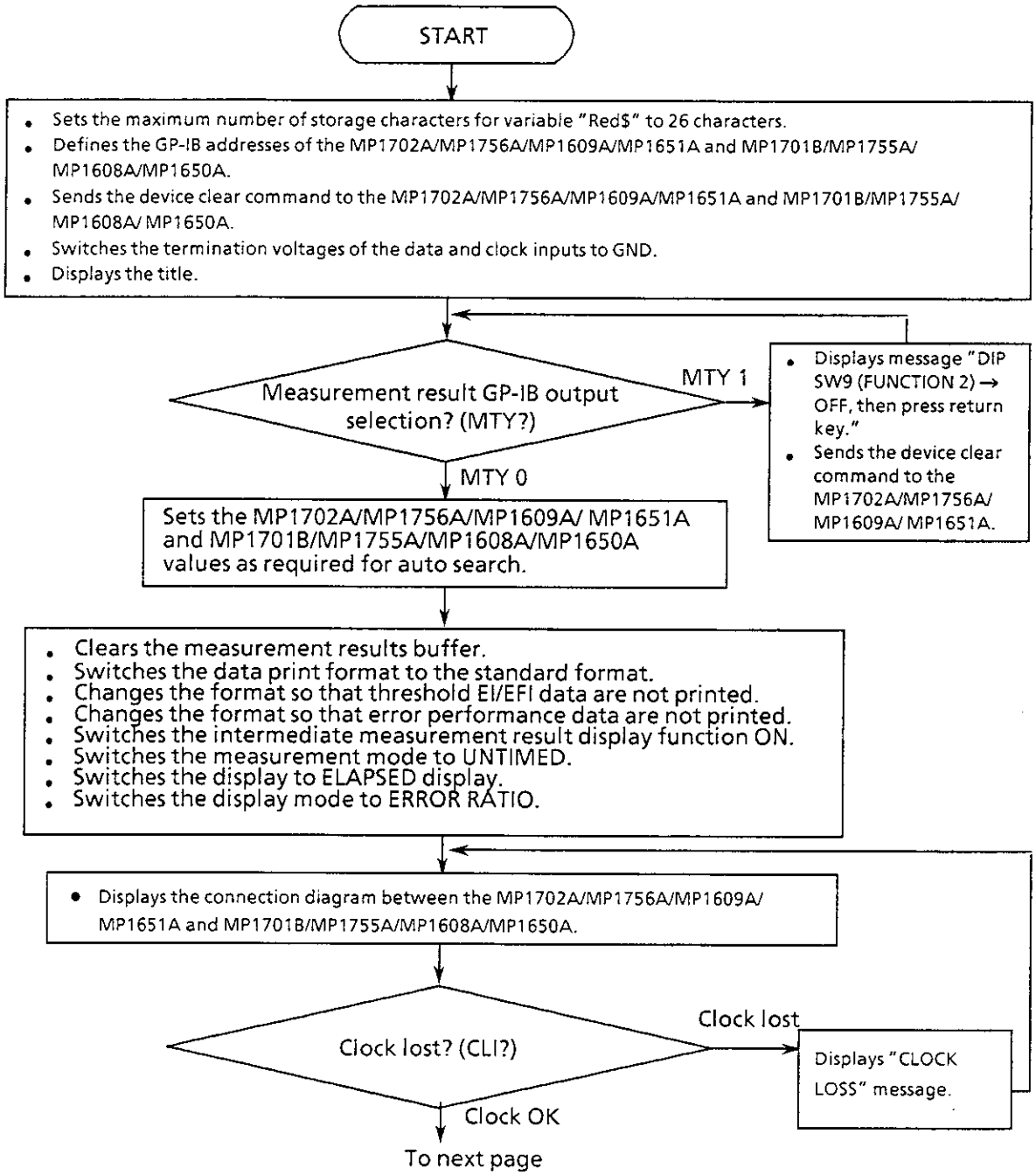
Table 15-4

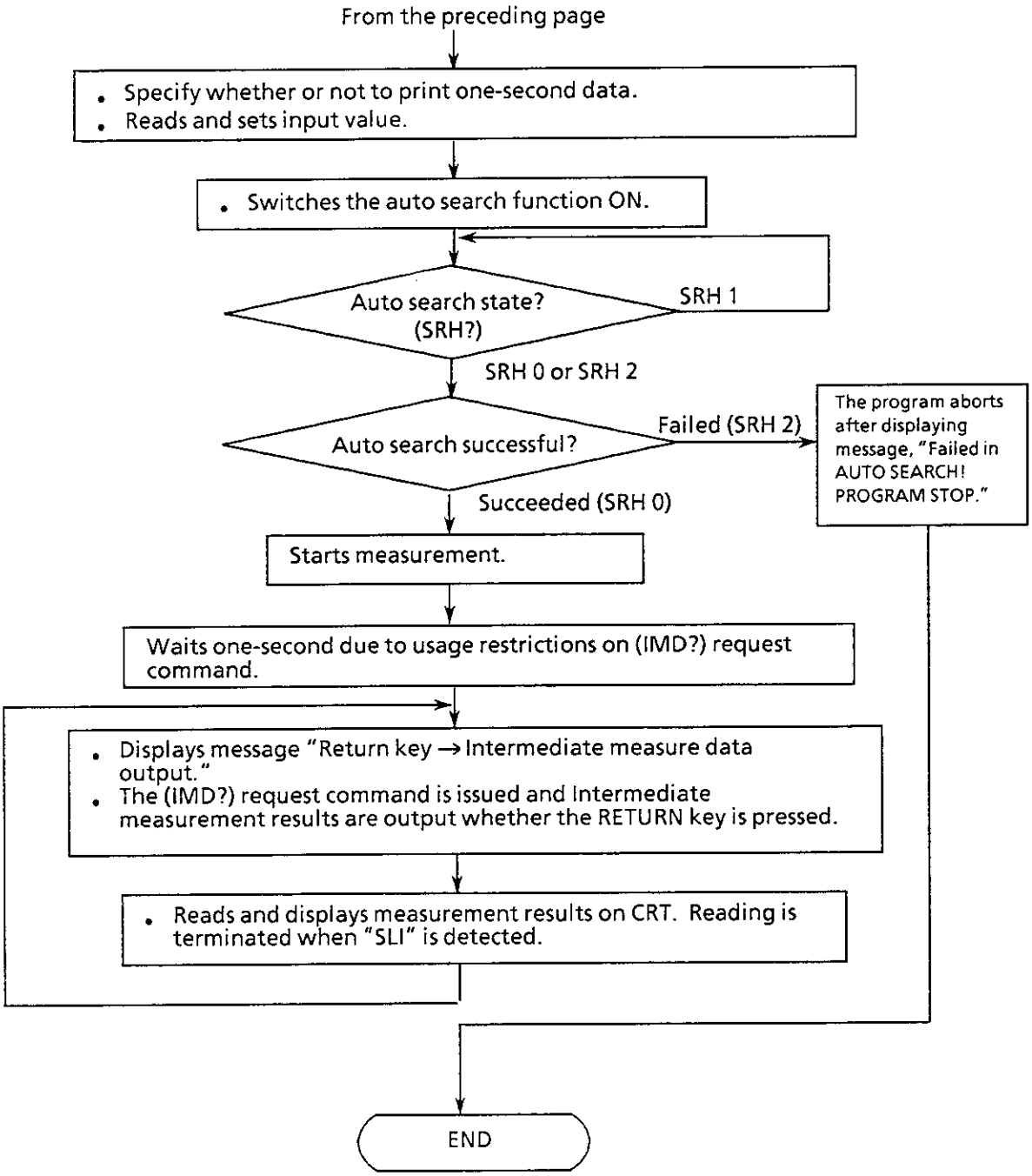
Paragraph	Data print format	Data output format (Standard or Extended)	Print one-second data
(a)	Standard	Standard	As selected
(b)	Standard	Extended: data including threshold and error performance data	No
(c)	Standard	Extended: data including threshold measurement data	No
(d)	Standard	Extended: data including error performance measurement data	No
(e)	Short (abridged)	Standard	No

Note: When using the MP1755A for the transmitter, internal clock frequency setting is ineffective. Set the frequency of the external clock source to 3000 MHz in advance.

(a) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Standard data are output in standard format; intermediate measurement results are output in response to (IMD?) request command.




From the preceding page
• Specify whether or not to print one-second data.
• Reads and sets input value.
• Switches the auto search function ON.
Auto search state? (SRH?)
SRH 1
SRH 0 or SRH 2
Auto search successful?
Failed (SRH 2)
The program aborts after displaying message, "Failed in AUTO SEARCH! PROGRAM STOP."
Succeeded (SRH 0)
Starts measurement.
Waits one-second due to usage restrictions on (IMD?) request command.
• Displays message "Return key → Intermediate measure data output."
• The (IMD?) request command is issued and Intermediate measurement results are output whether the RETURN key is pressed.
• Reads and displays measurement results on CRT. Reading is terminated when "SLI" is detected.
END

Program listing

```
10    !************************************************************
20    !*                                                          *
30    !*     MP1702A/MP1609A/MP1651A  MEASUREMENT  SAMPLE SOFT    *
40    !*          -- Standard format , Standard output --         *
50    !*              * Intermediate measure data *               *
60    !*                                                ED2_M1    *
70    !************************************************************
80    !
90    !-----------------------------------------------------------!
100   !                          MAIN ROUTINE                     !
110   !-----------------------------------------------------------!
120   !
130   DIM Red$[26]
140   LET Ad1=700               ! Set Device Address (PPG)
150   LET Ad2=701               ! Set Device Address (ED)
160   CLEAR Ad1                 ! Device Clear (PPG)
170   CLEAR Ad2                 ! Device Clear (ED)
180    !
190   OUTPUT Ad2;"DTM 0;CTM 0"
200    !
210   CLEAR SCREEN
220    !
230   PRINT "** MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT **"
240   PRINT
250    !
260   GOSUB M_type              ! Check Measurement type
270    !
280   GOSUB D_set               ! Data setting
290   GOSUB Mode                ! Measurement mode
300   GOSUB Clock               ! Check clock loss
310   GOSUB One_sec             ! One second data print ?
320   GOSUB Srch                ! Auto search : ON
330    !
340   OUTPUT Ad2;"STA"          ! Measurement Start
350    !
360   GOSUB Result              ! Display Result
370    !
380   STOP
390    !
400    !-----------------------------------------------------------!
410    !                          SUB ROUTINE                      !
420    !-----------------------------------------------------------!
430    !
440 M_type: !---------------------------- Check Measurement type
450   !
460   LOOP
470      !
480     OUTPUT Ad2;"MTY?"
490     ENTER Ad2;Mty$
500      !
510     IF Mty$="MTY 1" THEN
520       INPUT "DIP SW 9 (Function 2)-->OFF, then press return key",A
530       CLEAR Ad2
540     END IF
550      !
560   EXIT IF Mty$="MTY 0"
570   END LOOP
580    !
590   RETURN
600    !
```

```
610 D_set:  !------------------------------------------ Set Data .
620     !
630    OUTPUT Ad1;"LGC 0;PTN 2;MRK 1"
640    OUTPUT Ad2;"LGC 0;PTN 2;MRK 1"
650     !
660    OUTPUT Ad1;"CLK 1;RES 1;FRQ 3000"
670    OUTPUT Ad1;"OFS 0;SPD 0"
680    OUTPUT Ad1;"DAF 1;DOS 1"
690    OUTPUT Ad1;"CDL 100;CAF 1;COS 1"
700    OUTPUT Ad1;"EAD 1"
710     !
720    OUTPUT Ad2;"CPL 0"
730    OUTPUT Ad2;"SRH 0"
740    OUTPUT Ad2;"SYN 1"
750     !
760    RETURN
770     !
780 Mode: !------------------------------------ Measurement mode
790     !
800    OUTPUT Ad2;"BCL"          ! Measurement buffer clear
810    OUTPUT Ad2;"FMT 0"        ! Data Output:Standard format
820    OUTPUT Ad2;"THR 0"        ! Threshold output : OFF
830    OUTPUT Ad2;"EPF 0"        ! Error performance Output:OFF
840     !
850    OUTPUT Ad2;"CUR 1"        ! Current data : ON
860    OUTPUT Ad2;"MOD 2"        ! Measurement mode : UNTIMED
870    OUTPUT Ad2;"TIM 4"        ! Display ELAPSED
880    OUTPUT Ad2;"DMS 0"        ! Display ERROR RATIO
890     !
900    RETURN
910     !
920     !
930 Clock: !------------------------------------ Check connection
940     !
950    LOOP
960      GOSUB Connect           ! Display Device connection
970       !
980      OUTPUT Ad2;"CLI?"       ! REQUEST Clock loss ?
990      ENTER Ad2;Cli$
1000      !
1010     IF Cli$="CLI 1" THEN
1020       PRINT "** CLOCK LOSS **"
1030     END IF
1040      !
1050   EXIT IF Cli$="CLI 0"
1060   END LOOP
1070    !
1080   RETURN
1090    !
1100 One_sec: !--------------------------- One second data output ?
1110    !
1120   INPUT "One second data output [ OFF=0 , ON=1 ]",Osc$
1130   OUTPUT Ad2;"OSC "&Osc$
1140    !
1150   RETURN
1160    !
1170    !
1180 Srch:!------------------------------------------ Auto search
1190      !
1200   OUTPUT Ad2;"SRH 1"                          ! AUTO SEARCH : ON
1210    !
```

```
1220  LOOP
1230    !
1240    OUTPUT Ad2;"SRH?"
1250    ENTER Ad2;Srh$
1260    !
1270  EXIT IF Srh$="SRH 0" OR Srh$="SRH 2"
1280  END LOOP
1290   !
1300  IF Srh$="SRH 2" THEN
1310    PRINT "** Failed in AUTO SEARCH !  PROGRAM STOP **"
1320    STOP
1330    !
1340  END IF
1350   !
1360  RETURN
1370   !
1380   !
1390 Result: !------------------------------------ Print Result
1400   !
1410  WAIT 1
1420   !
1430  LOOP
1440    !
1450    INPUT "Return key --> Intermediate measure data output",A
1460    OUTPUT Ad2;"IMD?"
1470     !
1480    LOOP
1490      ENTER Ad2;Red$
1500      PRINT Red$
1510    EXIT IF Red$[1,3]="SLI"
1520    END LOOP
1530     !
1540  END LOOP
1550   !
1560  RETURN
1570   !
1580   !
1590 Connect: !-------------------------------- Display Connection
1600   !
1610  PEN 3
1620  VIEWPORT 70,140,50,100
1630  SHOW 0,70,0,50
1640   !
1650  CLIP 0,70,5,70
1660  FRAME
1670   !
1680  CSIZE 3,.4
1690  MOVE 25,45
1700  LABEL "<< CONNECTION >>"
1710   !
1720  CSIZE 3,.35
1730  MOVE 6,39
1740  LABEL "MP1701B/MP1608A/MP1650A    MP1702A/MP1609A/MP1651A"
1750  !
1760  MOVE 7,20
1770  RECTANGLE 25,18
1780   !
1790  MOVE 38,20
1800  RECTANGLE 25,18
1810   !
1820  MOVE 26,14
1830  IDRAW 0,9
```

```
1840   !
1850  FOR I=0 TO PI*2 STEP PI/12
1860    IDRAW .2*COS(I),.2*SIN(I)
1870  NEXT I
1880   !
1890  MOVE 26,14
1900  IDRAW 21,0
1910  IDRAW 0,9
1920   !
1930  FOR I=0 TO PI*2 STEP PI/12
1940    IDRAW .2*COS(I),.2*SIN(I)
1950  NEXT I
1960   !
1970  MOVE 21,17
1980  IDRAW 0,6
1990   !
2000  FOR I=0 TO PI*2 STEP PI/12
2010    IDRAW .2*COS(I),.2*SIN(I)
2020  NEXT I
2030   !
2040  MOVE 21,17
2050  IDRAW 21,0
2060  IDRAW 0,6
2070   !
2080  FOR I=0 TO PI*2 STEP PI/12
2090    IDRAW .2*COS(I),.2*SIN(I)
2100  NEXT I
2110   !
2120  MOVE 16,25
2130  CSIZE 2.3,.5
2140  LABEL "DATA  CLOCK1       DATA  CLOCK"
2150   !
2160  INPUT "Aer you ready ?  Press return key to start.",A
2170   !
2180  RETURN
2190   !
2200  END
```

EXECUTED RESULT

① When one-second data is not specified for printing

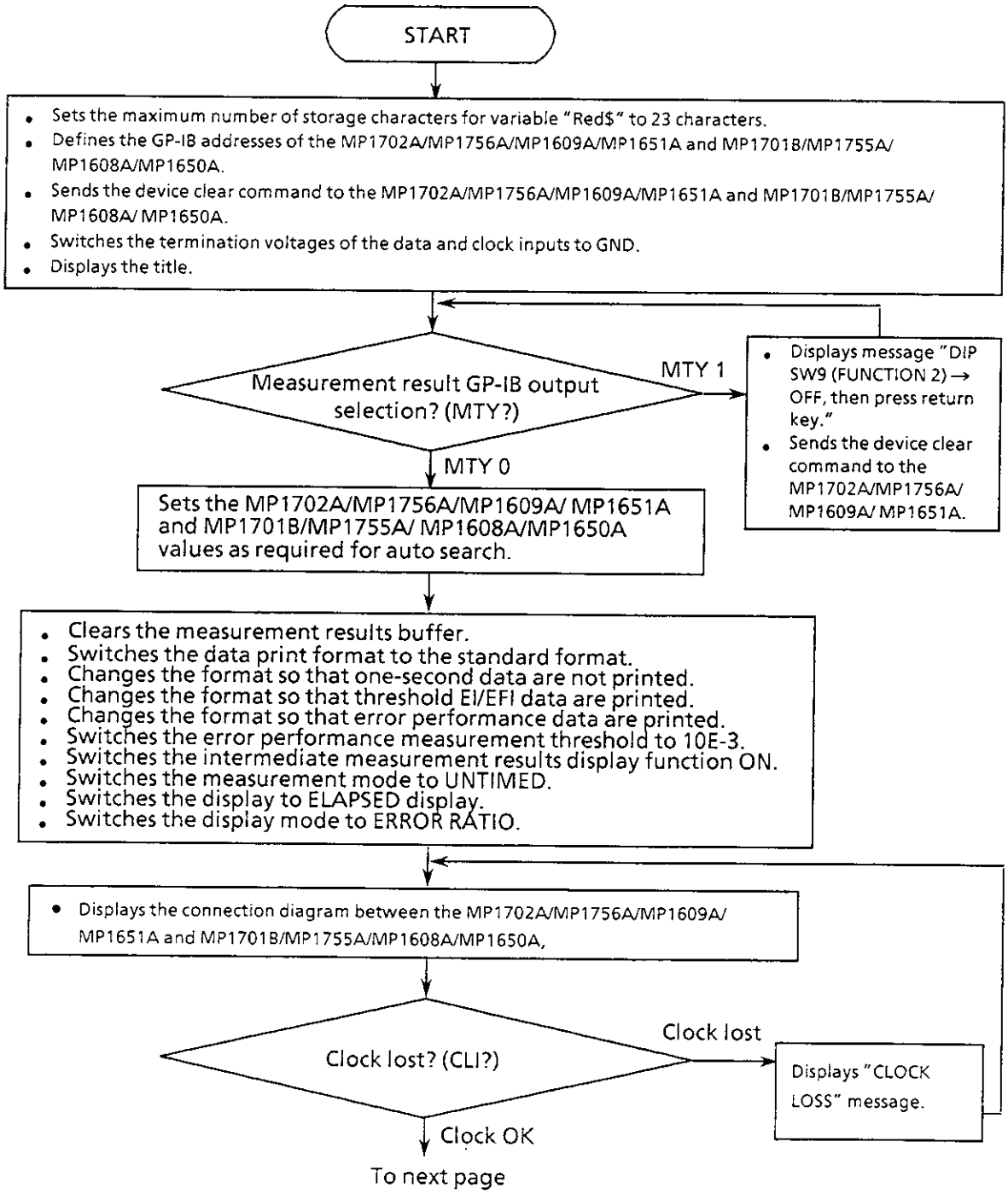
```
STA 90,02,26,12,35,27
STA 90,02,26,12,35,27
INT 90,02,26,12,35,37
ELP      0,00,00,10
ER  1.0000E-04
EC      5000011
EI           10
EFI      0.0000
PFI           0
CLI           0
SLI           0
STA 90,02,26,12,35,27
INT 90,02,26,12,35,42
ELP      0,00,00,15
ER  1.0000E-04
EC      7500016
EI           15
EFI      0.0000
PFI           0
CLI           0
SLI           0
```

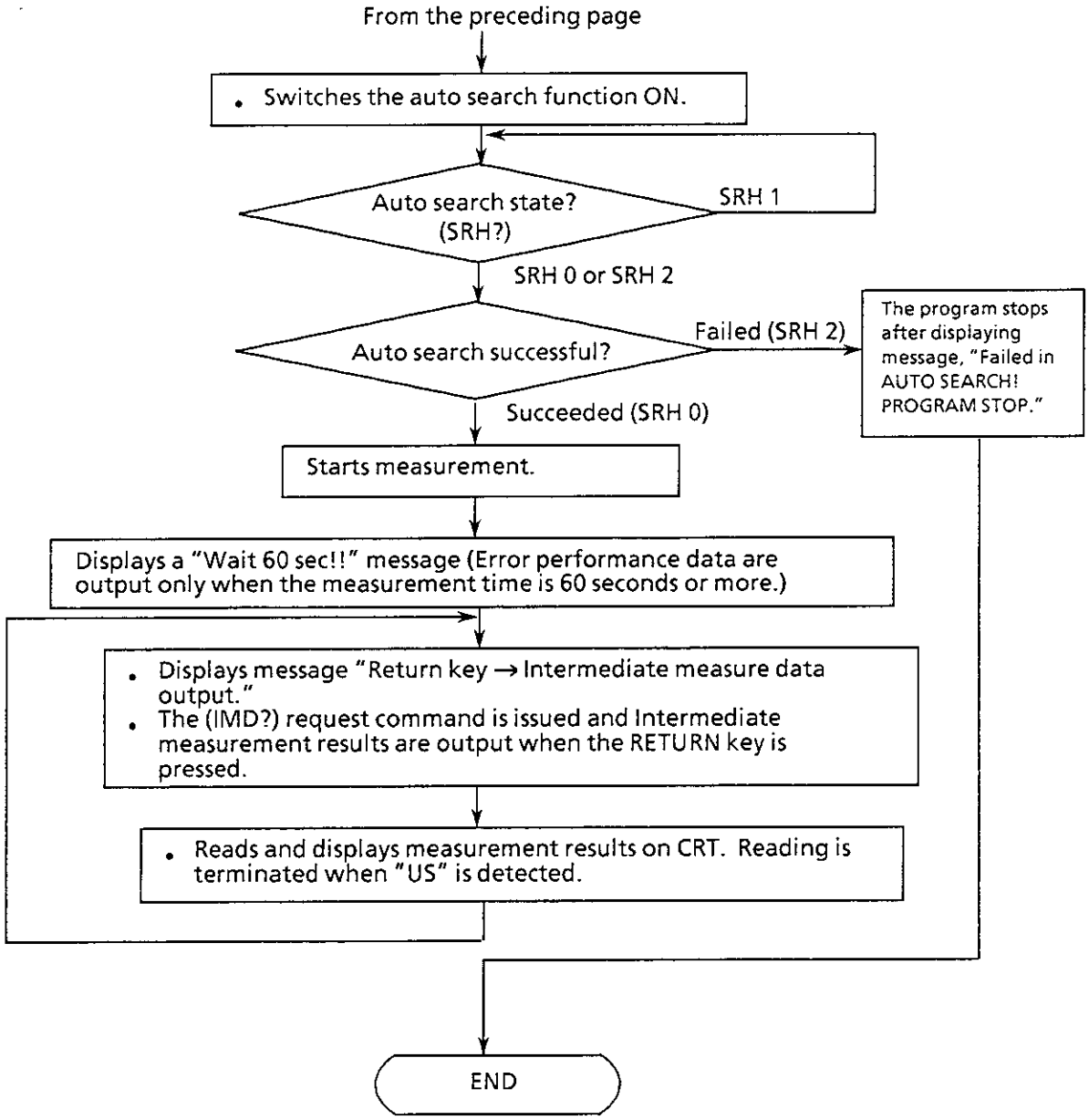
② When one-second data is specified for printing

```
STA 90,02,26,12,36,10
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
STA 90,02,26,12,36,10
INT 90,02,26,12,36,12
ELP      0,00,00,02
ER  1.0000E-04
EC      1000002
EI            2
EFI      0.0000
PFI           0
CLI           0
SLI           0
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
STA 90,02,26,12,36,10
INT 90,02,26,12,36,18
ELP      0,00,00,08
ER  1.0000E-04
EC      4000008
EI            8
EFI      0.0000
PFI           0
CLI           0
SLI           0
```

(b) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Extended data (threshold and error performance measurement data) are outut in standard format; intermediate measurement results are output when the (IMD?) request command is issued.




From the preceding page
• Switches the auto search function ON.
Auto search state? (SRH?)
SRH 1
SRH 0 or SRH 2
Auto search successful?
Failed (SRH 2)
The program stops after displaying message, "Failed in AUTO SEARCH! PROGRAM STOP."
Succeeded (SRH 0)
Starts measurement.
Displays a "Wait 60 sec!!" message (Error performance data are output only when the measurement time is 60 seconds or more.)
• Displays message "Return key → Intermediate measure data output."
• The (IMD?) request command is issued and Intermediate measurement results are output when the RETURN key is pressed.
• Reads and displays measurement results on CRT. Reading is terminated when "US" is detected.
END

Prigram listing

```
10    !*************************************************************
20    !*                                                           *
30    !*     MP1702A/MP1609A/MP1651A  MEASUREMENT  SAMPLE SOFT     *
40    !*        -- Standard format , Extension output --           *
50    !*       -- Threshold EI/EFI , Error performance --          *
60    !*            * Intermediate measure data *         ED2_M2  *
70    !*************************************************************
80    !
90    !-----------------------------------------------------------!
100   !                        MAIN ROUTINE                       !
110   !-----------------------------------------------------------!
120   !
130   DIM Red$[23]
140   LET Ad1=700               ! Set Device Address (PPG)
150   LET Ad2=701               ! Set Device Address (ED)
160   CLEAR Ad1                 ! Device Clear (PPG)
170   CLEAR Ad2                 ! Device Clear (ED)
180    !
190   OUTPUT Ad2;"DTM 0;CTM 0"
200    !
210   CLEAR SCREEN
220    !
230   PRINT "** MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT **"
240   PRINT
250    !
260   GOSUB M_type              ! Check Measurement type
270    !
280   GOSUB D_set               ! Data setting
290   GOSUB Mode                ! Measurement mode
300   GOSUB Clock               ! Check clock loss
310   GOSUB Srch                ! Auto search : ON
320    !
330   OUTPUT Ad2;"STA"          ! Measurement Start
340    !
350   GOSUB Result              ! Display Result
360    !
370   STOP
380    !
390    !-----------------------------------------------------------!
400    !                          SUB ROUTINE                      !
410    !-----------------------------------------------------------!
420    !
430 M_type: !------------------------------ Check Measurement type
440   !
450   LOOP
460      !
470     OUTPUT Ad2;"MTY?"
480     ENTER Ad2;Mty$
490      !
500     IF Mty$="MTY 1" THEN
510       INPUT "DIP SW 9 (Function 2)-->OFF, then press return key",A
520       CLEAR Ad2
530     END IF
540      !
550   EXIT IF Mty$="MTY 0"
560   END LOOP
570    !
580   RETURN
590    !
```

```
600 D_set:  !------------------------------------------ Set Data
610     !
620    OUTPUT Ad1;"LGC 0;PTN 2;MRK 1"
630    OUTPUT Ad2;"LGC 0;PTN 2;MRK 1"
640     !
650    OUTPUT Ad1;"CLK 1;RES 1;FRQ 3000"
660    OUTPUT Ad1;"OFS 0;SPD 0"
670    OUTPUT Ad1;"DAP 1;DOS 1"
680    OUTPUT Ad1;"CDL 100;CAP 1;COS 1"
690    OUTPUT Ad1;"EAD 1"
700     !
710    OUTPUT Ad2;"CPL 0"
720    OUTPUT Ad2;"SRH 0"
730    OUTPUT Ad2;"SYN 1"
740     !
750    RETURN
760     !
770 Mode: !------------------------------------ Measurement mode
780     !
790    OUTPUT Ad2;"BCL"           ! Measurement buffer clear
800    OUTPUT Ad2;"FMT 0"         ! Data Output:Standard format
810    OUTPUT Ad2;"OSC 0"         ! One second data output : OFF
820    OUTPUT Ad2;"THR 1"         ! Threshold output : ON
830    OUTPUT Ad2;"EPF 1"         ! Error performance Output:ON
840    OUTPUT Ad2;"ETH 0"         ! Error performance threshold : 10E-3
850     !
860    OUTPUT Ad2;"CUR 1"         ! Current data : ON
870    OUTPUT Ad2;"MOD 2"         ! Measurement mode : UNTIMED
880    OUTPUT Ad2;"TIM 4"         ! Display ILAPSED
890    OUTPUT Ad2;"DMS 0"         ! Display ERROR RATIO
900     !
910    RETURN
920     !
930     !
940 Clock: !------------------------------------ Check connection
950     !
960    LOOP
970      GOSUB Connect            ! Display Device connection
980       !
990      OUTPUT Ad2;"CLI?"        ! REQUEST Clock loss ?
1000     ENTER Ad2;Cli$
1010      !
1020     IF Cli$="CLI 1" THEN
1030       PRINT "** CLOCK LOSS **"
1040     END IF
1050      !
1060   EXIT IF Cli$="CLI 0"
1070   END LOOP
1080    !
1090   RETURN
1100    !
1110    !
1120 Srch:!------------------------------------------ Auto search
1130       !
1140   OUTPUT Ad2;"SRH 1"                          ! AUTO SEARCH : ON
1150    !
1160   LOOP
1170      !
1180     OUTPUT Ad2;"SRH?"
1190     ENTER Ad2;Srh$
1200      !
1210   EXIT IF Srh$="SRH 0" OR Srh$="SRH 2"
1220   END LOOP
```

```
1230   !
1240  IF Srh$="SRH 2" THEN
1250    PRINT "** Failed in AUTO SEARCH !  PROGRAM STOP **"
1260    STOP
1270     !
1280  END IF
1290   !
1300  RETURN
1310   !
1320   !
1330 Result: !------------------------------------ Print Result
1340   !
1350  PRINT
1360  PRINT "***** Wait 60 sec !!"
1370  PRINT "            To output Error performance data *****"
1380  PRINT
1390   !
1400  LOOP
1410     !
1420    INPUT "Return key --> Intermediate measure data output",A
1430    OUTPUT Ad2;"IMD?"
1440     !
1450    LOOP
1460      ENTER Ad2;Red$
1470      PRINT Red$
1480    EXIT IF Red$[1,2]="US"
1490    END LOOP
1500     !
1510  END LOOP
1520   !
1530  RETURN
1540   !
1550   !
1560 Connect: !-------------------------------- Display Connection
1570   !
1580  PEN 3
1590  VIEWPORT 70,140,50,100
1600  SHOW 0,70,0,50
1610   !
1620  CLIP 0,70,5,70
1630  FRAME
1640   !
1650  CSIZE 3,.4
1660  MOVE 25,45
1670  LABEL "<< CONNECTION >>"
1680   !
1690  CSIZE 3,.35
1700  MOVE 6,39
1710  LABEL "MP1701B/MP1608A/MP1650A    MP1702A/MP1609A/MP1651A"
1720  !
1730  MOVE 7,20
1740  RECTANGLE 25,18
1750   !
1760  MOVE 38,20
1770  RECTANGLE 25,18
1780   !
1790  MOVE 26,14
1800  IDRAW 0,9
1810   !
1820  FOR I=0 TO PI*2 STEP PI/12
1830    IDRAW .2*COS(I),.2*SIN(I)
1840  NEXT I
1850   !
```

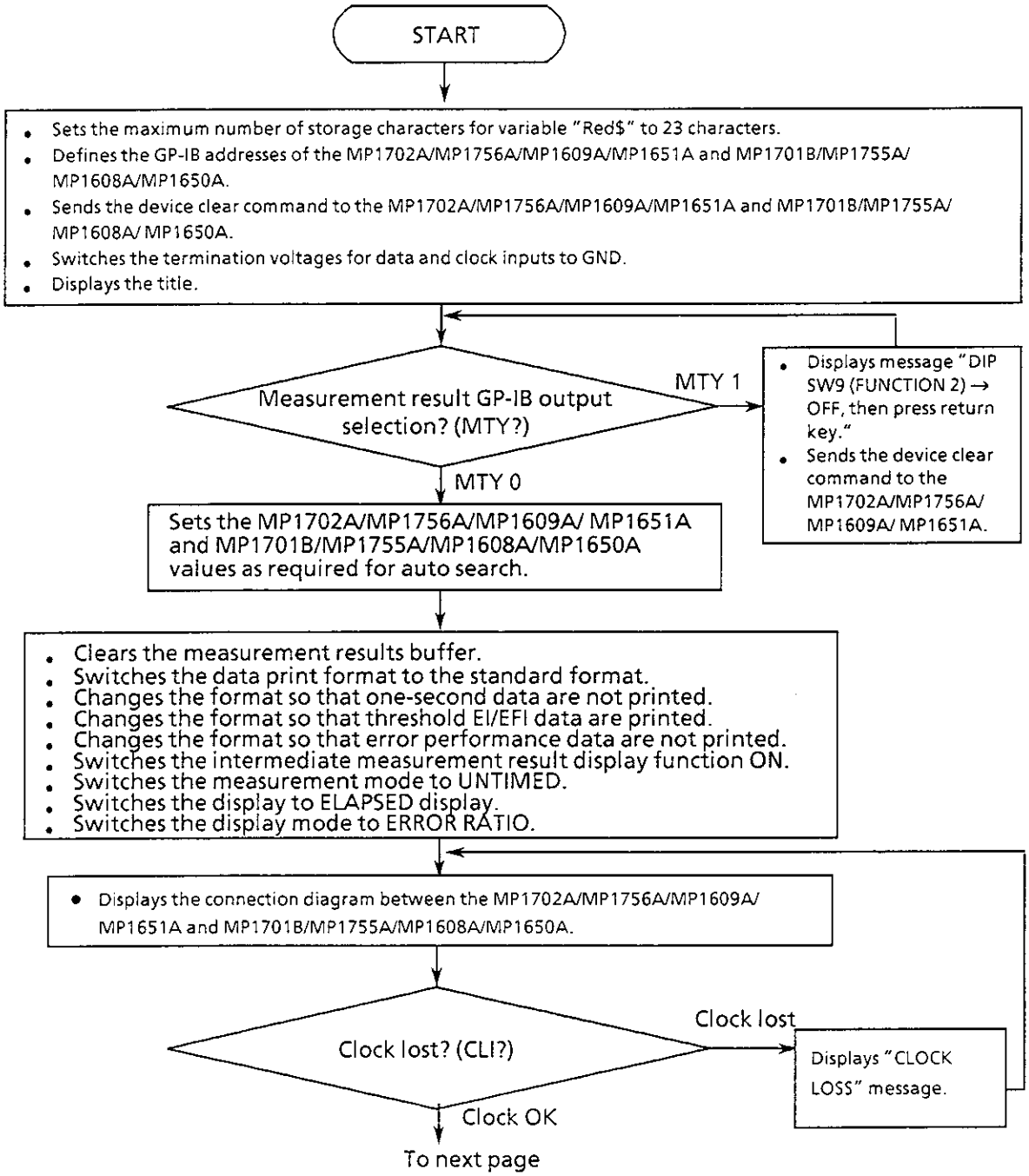
```
1860  MOVE 26,14
1870  IDRAW 21,0
1880  IDRAW 0,9
1890   !
1900  FOR I=0 TO PI*2 STEP PI/12
1910    IDRAW .2*COS(I),.2*SIN(I)
1920  NEXT I
1930   !
1940  MOVE 21,17
1950  IDRAW 0,6
1960   !
1970  FOR I=0 TO PI*2 STEP PI/12
1980    IDRAW .2*COS(I),.2*SIN(I)
1990  NEXT I
2000   !
2010  MOVE 21,17
2020  IDRAW 21,0
2030  IDRAW 0,6
2040   !
2050  FOR I=0 TO PI*2 STEP PI/12
2060    IDRAW .2*COS(I),.2*SIN(I)
2070  NEXT I
2080   !
2090  MOVE 16,25
2100  CSIZE 2.3,.5
2110  LABEL "DATA  CLOCK1       DATA  CLOCK"
2120   !
2130  INPUT "Aer you ready ?  Press return key to start.",A
2140   !
2150  RETURN
2160   !
2170  END
```

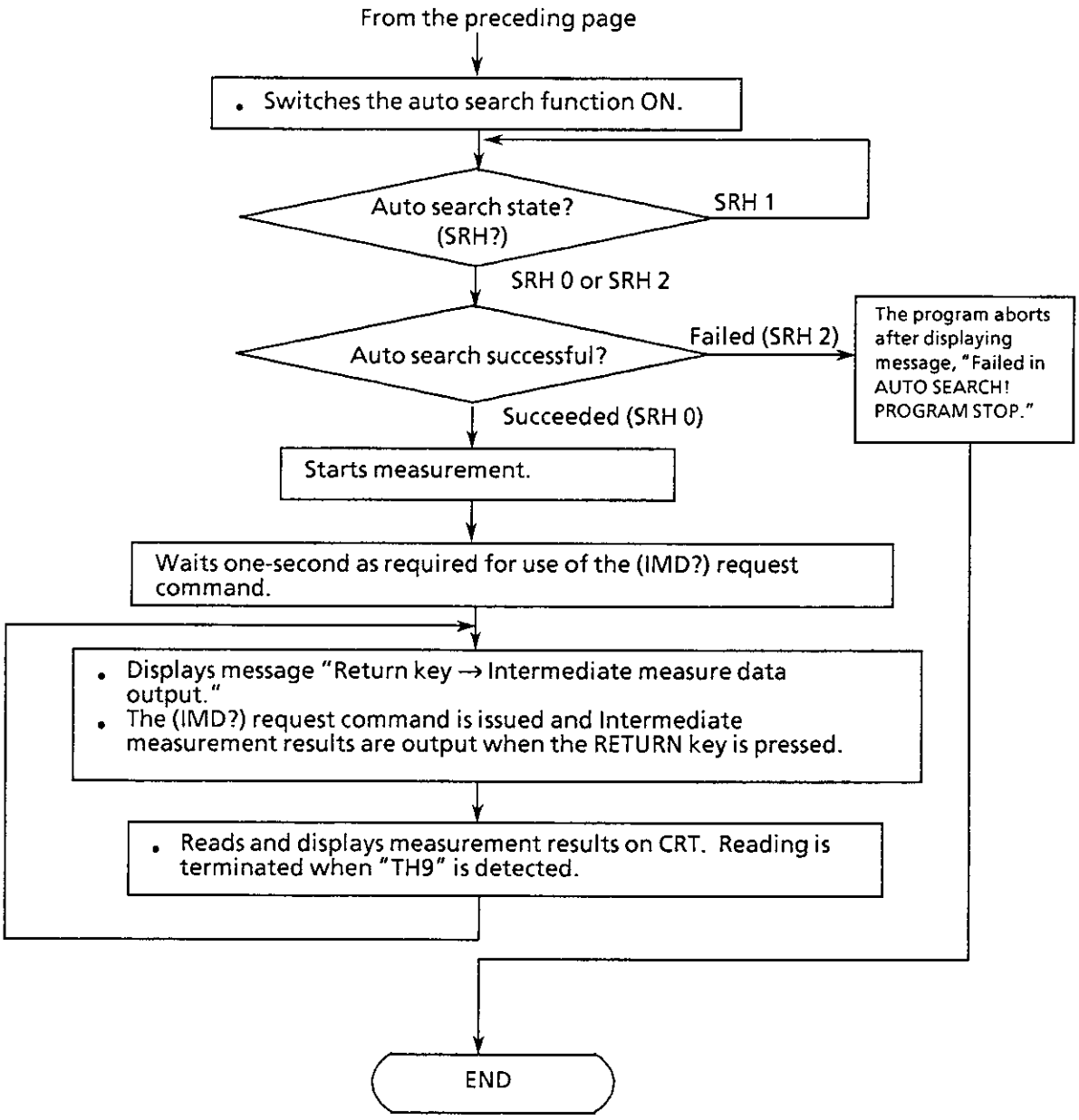
EXECUTED RESULT

```
STA 90,02,26,12,37,26
STA 90,02,26,12,37,26
INT 90,02,26,12,38,28
ELP     0,00,01,02
ER  1.0000E-04
EC    31000063
EI          62
EFI     0.0000
PFI          0
CLI          0
SLI          0
TH3          0,100.0000
TH4          2, 96.7741
TH5         62,  0.0000
TH6         62,  0.0000
TH7         62,  0.0000
TH8         62,  0.0000
TH9          0,100.0000
ES    100.0000
EFS     0.0000
SES     0.0000
DM    100.0000
US      0.0000
STA 90,02,26,12,37,26
INT 90,02,26,12,38,40
ELP     0,00,01,14
ER  1.0000E-04
EC    37000076
EI          74
EFI     0.0000
PFI          0
CLI          0
SLI          0
TH3          0,100.0000
TH4          3, 95.9459
TH5         74,  0.0000
TH6         74,  0.0000
TH7         74,  0.0000
TH8         74,  0.0000
TH9          0,100.0000
ES    100.0000
EFS     0.0000
SES     0.0000
DM    100.0000
US      0.0000
```

(c) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Extended data (threshold measurement data) are outut in standard format; intermediate measurement result are output in response to an (IMD?) request command.




From the preceding page
• Switches the auto search function ON.
Auto search state? (SRH?)
SRH 1
SRH 0 or SRH 2
Auto search successful?
Failed (SRH 2)
The program aborts after displaying message, "Failed in AUTO SEARCH! PROGRAM STOP."
Succeeded (SRH 0)
Starts measurement.
Waits one-second as required for use of the (IMD?) request command.
• Displays message "Return key → Intermediate measure data output."
• The (IMD?) request command is issued and Intermediate measurement results are output when the RETURN key is pressed.
• Reads and displays measurement results on CRT. Reading is terminated when "TH9" is detected.
END

Program listing

```
10    !*************************************************************
20    !*                                                           *
30    !*     MP1702A/MP1609A/MP1651A  MEASUREMENT  SAMPLE SOFT     *
40    !*        -- Standard format , Extension output --           *
50    !*                -- Threshold EI/EFI --                     *
60    !*            * Intermediate measure data *         ED2_M3  *
70    !*************************************************************
80    !
90    !-----------------------------------------------------------!
100   !                         MAIN ROUTINE                      !
110   !-----------------------------------------------------------!
120   !
130   DIM Red$[23]
140   LET Ad1=700               ! Set Device Address (PPG)
150   LET Ad2=701               ! Set Device Address (ED)
160   CLEAR Ad1                 ! Device Clear (PPG)
170   CLEAR Ad2                 ! Device Clear (ED)
180    !
190   OUTPUT Ad2;"DTM 0;CTM 0"
200    !
210   CLEAR SCREEN
220    !
230   PRINT "** MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT **"
240   PRINT
250    !
260   GOSUB M_type              ! Check Measurement type
270    !
280   GOSUB D_set               ! Data setting
290   GOSUB Mode                ! Measurement mode
300   GOSUB Clock               ! Check clock loss
310   GOSUB Srch                ! Auto search : ON
320    !
330   OUTPUT Ad2;"STA"          ! Measurement Start
340    !
350   GOSUB Result              ! Display Result
360    !
370   STOP
380    !
390    !-----------------------------------------------------------!
400    !                          SUB ROUTINE                      !
410    !-----------------------------------------------------------!
420    !
430 M_type: !----------------------------- Check Measurement type
440   !
450   LOOP
460      !
470     OUTPUT Ad2;"MTY?"
480     ENTER Ad2;Mty$
490      !
500     IF Mty$="MTY 1" THEN
510       INPUT "DIP SW 9 (Function 2)-->OFF, then press return key",A
520       CLEAR Ad2
530     END IF
540      !
550   EXIT IF Mty$="MTY 0"
560   END LOOP
570    !
580   RETURN
590    !
```

```
600 D_set:  !------------------------------------------ Set Data
610     !
620    OUTPUT Ad1;"LGC 0;PTN 2;MRK 1"
630    OUTPUT Ad2;"LGC 0;PTN 2;MRK 1"
640     !
650    OUTPUT Ad1;"CLK 1;RES 1;FRQ 3000"
660    OUTPUT Ad1;"OFS 0;SPD 0"
670    OUTPUT Ad1;"DAP 1;DOS 1"
680    OUTPUT Ad1;"CDL 100;CAP 1;COS 1"
690    OUTPUT Ad1;"EAD 1"
700     !
710    OUTPUT Ad2;"CPL 0"
720    OUTPUT Ad2;"SRH 0"
730    OUTPUT Ad2;"SYN 1"
740     !
750    RETURN
760     !
770 Mode: !------------------------------------ Measurement mode
780     !
790    OUTPUT Ad2;"BCL"            ! Measurement buffer clear
800    OUTPUT Ad2;"FMT 0"          ! Data Output:Standard format
810    OUTPUT Ad2;"OSC 0"          ! One second data output : OFF
820    OUTPUT Ad2;"THR 1"          ! Threshold output : ON
830    OUTPUT Ad2;"EPF 0"          ! Error performance Output:OFF
840     !
850    OUTPUT Ad2;"CUR 1"          ! Current data : ON
860    OUTPUT Ad2;"MOD 2"          ! Measurement mode : UNTIMED
870    OUTPUT Ad2;"TIM 4"          ! Display ILAPSED
880    OUTPUT Ad2;"DMS 0"          ! Display ERROR RATIO
890     !
900    RETURN
910     !
920     !
930 Clock: !------------------------------------ Check connection
940     !
950    LOOP
960      GOSUB Connect             ! Display Device connection
970       !
980      OUTPUT Ad2;"CLI?"         ! REQUEST Clock loss ?
990      ENTER Ad2;Cli$
1000      !
1010     IF Cli$="CLI 1" THEN
1020       PRINT "** CLOCK LOSS **"
1030     END IF
1040      !
1050   EXIT IF Cli$="CLI 0"
1060   END LOOP
1070    !
1080   RETURN
1090    !
1100    !
1110 Srch:!------------------------------------------ Auto search
1120       !
1130   OUTPUT Ad2;"SRH 1"                          ! AUTO SEARCH : ON
1140    !
1150   LOOP
1160      !
1170     OUTPUT Ad2;"SRH?"
1180     ENTER Ad2;Srh$
1190      !
1200   EXIT IF Srh$="SRH 0" OR Srh$="SRH 2"
1210   END LOOP
```

```
1220   !
1230  IF Srh$="SRH 2" THEN
1240    PRINT "** Failed in AUTO SEARCH !  PROGRAM STOP **"
1250    STOP
1260     !
1270  END IF
1280   !
1290  RETURN
1300   !
1310   !
1320 Result: !-------------------------------------- Print Result
1330   !
1340  WAIT 1
1350   !
1360  LOOP
1370     !
1380    INPUT "Return key --> Intermediate measure data output",A
1390    OUTPUT Ad2;"IMD?"
1400     !
1410    LOOP
1420      ENTER Ad2;Red$
1430      PRINT Red$
1440    EXIT IF Red$[1,3]="TH9"
1450    END LOOP
1460     !
1470  END LOOP
1480   !
1490  RETURN
1500   !
1510   !
1520 Connect: !-------------------------------- Display Connection
1530   !
1540  PEN 3
1550  VIEWPORT 70,140,50,100
1560  SHOW 0,70,0,50
1570   !
1580  CLIP 0,70,5,70
1590  FRAME
1600   !
1610  CSIZE 3,.4
1620  MOVE 25,45
1630  LABEL "<< CONNECTION >>"
1640   !
1650  CSIZE 3,.35
1660  MOVE 6,39
1670  LABEL "MP1701B/MP1608A/MP1650A    MP1702A/MP1609A/MP1651A"
1680  !
1690  MOVE 7,20
1700  RECTANGLE 25,18
1710   !
1720  MOVE 38,20
1730  RECTANGLE 25,18
1740   !
1750  MOVE 26,14
1760  IDRAW 0,9
1770   !
1780  FOR I=0 TO PI*2 STEP PI/12
1790    IDRAW .2*COS(I),.2*SIN(I)
1800  NEXT I
1810   !
```

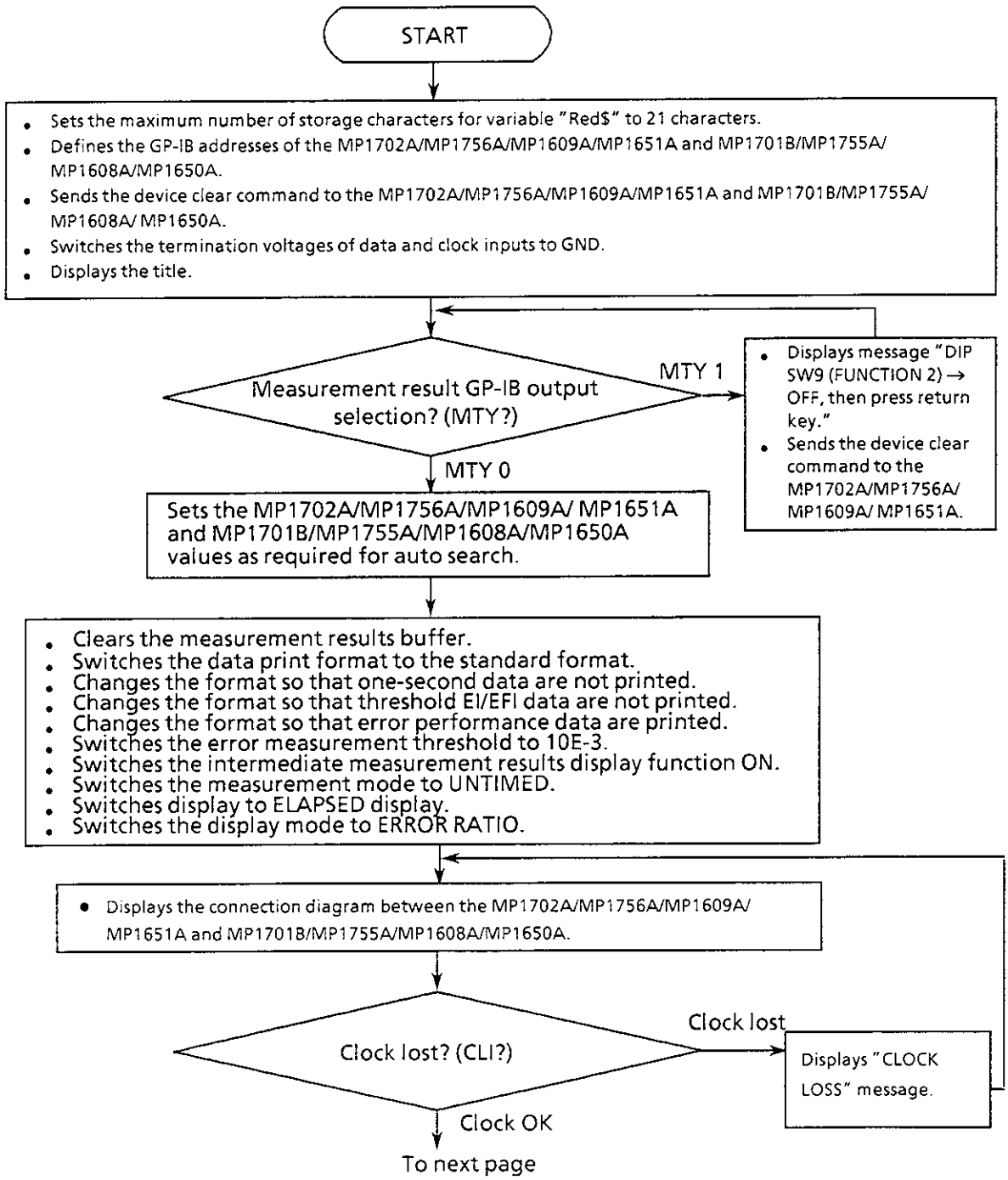
```
1820  MOVE 26,14
1830  IDRAW 21,0
1840  IDRAW 0,9
1850   !
1860  FOR I=0 TO PI*2 STEP PI/12
1870    IDRAW .2*COS(I),.2*SIN(I)
1880  NEXT I
1890   !
1900  MOVE 21,17
1910  IDRAW 0,6
1920   !
1930  FOR I=0 TO PI*2 STEP PI/12
1940    IDRAW .2*COS(I),.2*SIN(I)
1950  NEXT I
1960   !
1970  MOVE 21,17
1980  IDRAW 21,0
1990  IDRAW 0,6
2000   !
2010  FOR I=0 TO PI*2 STEP PI/12
2020    IDRAW .2*COS(I),.2*SIN(I)
2030  NEXT I
2040   !
2050  MOVE 16,25
2060  CSIZE 2.3,.5
2070  LABEL "DATA  CLOCK1       DATA  CLOCK"
2080   !
2090  INPUT "Aer you ready ?  Press return key to start.",A
2100   !
2110  RETURN
2120   !
2130  END
```

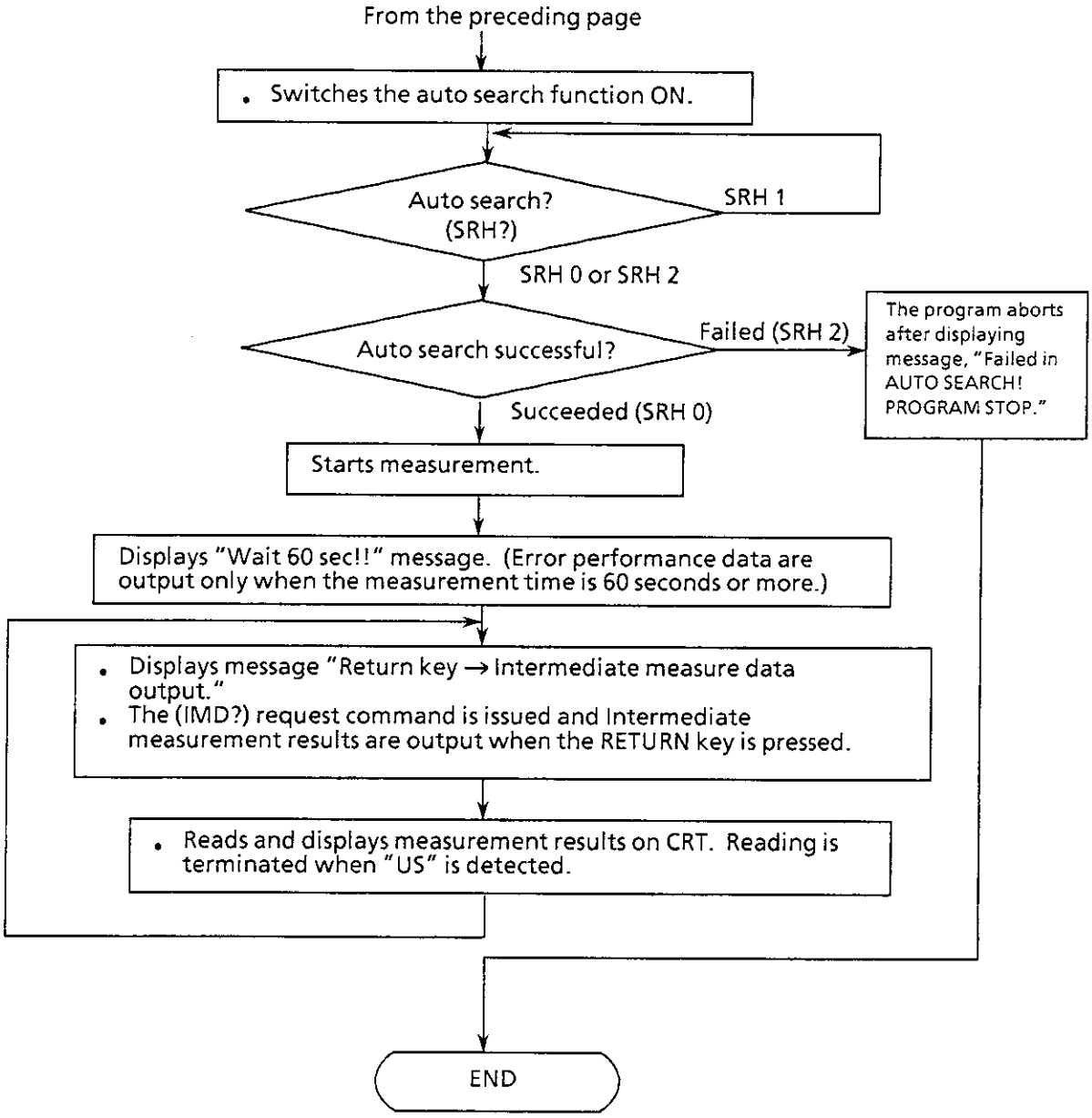
EXECUTED RESULT

```
STA 90,02,26,12,39,14
STA 90,02,26,12,39,14
INT 90,02,26,12,39,16
ELP      0,00,00,02
ER  1.0000E-04
EC      1000002
EI            2
EFI      0.0000
PFI           0
CLI           0
SLI           0
TH3           0,100.0000
TH4           0,100.0000
TH5           2,   0.0000
TH6           2,   0.0000
TH7           2,   0.0000
TH8           2,   0.0000
TH9           0,100.0000
STA 90,02,26,12,39,14
INT 90,02,26,12,39,24
ELP      0,00,00,10
ER  1.0000E-04
EC      5000010
EI           10
EFI      0.0000
PFI           0
CLI           0
SLI           0
TH3           0,100.0000
TH4           0,100.0000
TH5          10,   0.0000
TH6          10,   0.0000
TH7          10,   0.0000
TH8          10,   0.0000
TH9           0,100.0000
```

(d) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Extended data (error performance measurement data) are output in standard formats; intermediate measurement results are output in response to the (IMD?) request command.




From the preceding page
• Switches the auto search function ON.
Auto search? (SRH?)
SRH 1
SRH 0 or SRH 2
Auto search successful?
Failed (SRH 2)
The program aborts after displaying message, "Failed in AUTO SEARCH! PROGRAM STOP."
Succeeded (SRH 0)
Starts measurement.
Displays "Wait 60 sec!!" message. (Error performance data are output only when the measurement time is 60 seconds or more.)
• Displays message "Return key → Intermediate measure data output."
• The (IMD?) request command is issued and Intermediate measurement results are output when the RETURN key is pressed.
• Reads and displays measurement results on CRT. Reading is terminated when "US" is detected.
END

Program listing

```
10    !*************************************************************
20    !*                                                           *
30    !*      MP1702A/MP1609A/MP1651A  MEASUREMENT  SAMPLE SOFT    *
40    !*         -- Standard format , Extension output --          *
50    !*                -- Error performance --                    *
60    !*             * Intermediate measure data *       ED2_M4   *
70    !*************************************************************
80    !
90    !-----------------------------------------------------------!
100   !                         MAIN ROUTINE                      !
110   !-----------------------------------------------------------!
120   !
130   DIM Red$[21]
140   LET Ad1=700               ! Set Device Address (PPG)
150   LET Ad2=701               ! Set Device Address (ED)
160   CLEAR Ad1                 ! Device Clear (PPG)
170   CLEAR Ad2                 ! Device Clear (ED)
180    !
190   OUTPUT Ad2;"DTM 0;CTM 0"
200    !
210   CLEAR SCREEN
220    !
230   PRINT "** MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT **"
240   PRINT
250    !
260   GOSUB M_type              ! Check Measurement type
270    !
280   GOSUB D_set               ! Data setting
290   GOSUB Mode                ! Measurement mode
300   GOSUB Clock               ! Check clock loss
310   GOSUB Srch                ! Auto search : ON
320    !
330   OUTPUT Ad2;"STA"          ! Measurement Start
340    !
350   GOSUB Result              ! Display Result
360    !
370   STOP
380    !
390    !-----------------------------------------------------------!
400    !                          SUB ROUTINE                      !
410    !-----------------------------------------------------------!
420    !
430 M_type: !------------------------------ Check Measurement type
440   !
450   LOOP
460      !
470     OUTPUT Ad2;"MTY?"
480     ENTER Ad2;Mty$
490      !
500     IF Mty$="MTY 1" THEN
510       INPUT "DIP SW 9 (Function 2)-->OFF, then press return key",A
520       CLEAR Ad2
530     END IF
540      !
550   EXIT IF Mty$="MTY 0"
560   END LOOP
570    !
580   RETURN
590    !
```

```
600 D_set:  !------------------------------------------ Set Data
610     !
620    OUTPUT Ad1;"LGC 0;PTN 2;MRK 1"
630    OUTPUT Ad2;"LGC 0;PTN 2;MRK 1"
640     !
650    OUTPUT Ad1;"CLK 1;RES 1;FRQ 3000"
660    OUTPUT Ad1;"OFS 0;SPD 0"
670    OUTPUT Ad1;"DAP 1;DOS 1"
680    OUTPUT Ad1;"CDL 100;CAP 1;COS 1"
690    OUTPUT Ad1;"EAD 1"
700     !
710    OUTPUT Ad2;"CPL 0"
720    OUTPUT Ad2;"SRH 0"
730    OUTPUT Ad2;"SYN 1"
740     !
750    RETURN
760     !
770 Mode: !------------------------------------ Measurement mode
780     !
790    OUTPUT Ad2;"BCL"           ! Measurement buffer clear
800    OUTPUT Ad2;"FMT 0"         ! Data Output:Standard format
810    OUTPUT Ad2;"OSC 0"         ! One second data output : OFF
820    OUTPUT Ad2;"THR 0"         ! Threshold EI/EFI : OFF
830    OUTPUT Ad2;"EPF 1"         ! Error performance Output:ON
840    OUTPUT Ad2;"ETH 0"         ! Error performance threshold : 10E-3
850     !
860    OUTPUT Ad2;"CUR 1"         ! Current data : ON
870    OUTPUT Ad2;"MOD 2"         ! Measurement mode : UNTIMED
880    OUTPUT Ad2;"TIM 4"         ! Display ILAPSED
890    OUTPUT Ad2;"DMS 0"         ! Display ERROR RATIO
900     !
910    RETURN
920     !
930     !
940 Clock: !------------------------------------ Check connection
950     !
960    LOOP
970      GOSUB Connect            ! Display Device connection
980       !
990      OUTPUT Ad2;"CLI?"        ! REQUEST Clock loss ?
1000     ENTER Ad2;Cli$
1010      !
1020     IF Cli$="CLI 1" THEN
1030       PRINT "** CLOCK LOSS **"
1040     END IF
1050      !
1060   EXIT IF Cli$="CLI 0"
1070   END LOOP
1080    !
1090   RETURN
1100    !
1110    !
1120 Srch:!------------------------------------------ Auto search
1130       !
1140   OUTPUT Ad2;"SRH 1"                          ! AUTO SEARCH : ON
1150    !
1160   LOOP
1170      !
1180     OUTPUT Ad2;"SRH?"
1190     ENTER Ad2;Srh$
1200      !
1210   EXIT IF Srh$="SRH 0" OR Srh$="SRH 2"
1220   END LOOP
```

```
1230   !
1240  IF Srh$="SRH 2" THEN
1250    PRINT "** Failed in AUTO SEARCH !  PROGRAM STOP **"
1260    STOP
1270     !
1280  END IF
1290   !
1300  RETURN
1310   !
1320   !
1330 Result: !------------------------------------ Print Result
1340   !
1350  PRINT
1360  PRINT "***** Wait 60 sec !!"
1370  PRINT "            To output Error performance data *****"
1380  PRINT
1390   !
1400  LOOP
1410     !
1420    INPUT "Return key --> Intermediate measure data output",A
1430    OUTPUT Ad2;"IMD?"
1440     !
1450    LOOP
1460      ENTER Ad2;Red$
1470      PRINT Red$
1480    EXIT IF Red$[1,2]="US"
1490    END LOOP
1500     !
1510  END LOOP
1520   !
1530  RETURN
1540   !
1550   !
1560 Connect: !-------------------------------- Display Connection
1570   !
1580  PEN 3
1590  VIEWPORT 70,140,50,100
1600  SHOW 0,70,0,50
1610   !
1620  CLIP 0,70,5,70
1630  FRAME
1640   !
1650  CSIZE 3,.4
1660  MOVE 25,45
1670  LABEL "<< CONNECTION >>"
1680   !
1690  CSIZE 3,.35
1700  MOVE 6,39
1710  LABEL "MP1701B/MP1608A/MP1650A    MP1702A/MP1609A/MP1651A"
1720  !
1730  MOVE 7,20
1740  RECTANGLE 25,18
1750   !
1760  MOVE 38,20
1770  RECTANGLE 25,18
1780   !
1790  MOVE 26,14
1800  IDRAW 0,9
1810   !
1820  FOR I=0 TO PI*2 STEP PI/12
1830    IDRAW .2*COS(I),.2*SIN(I)
1840  NEXT I
1850   !
```

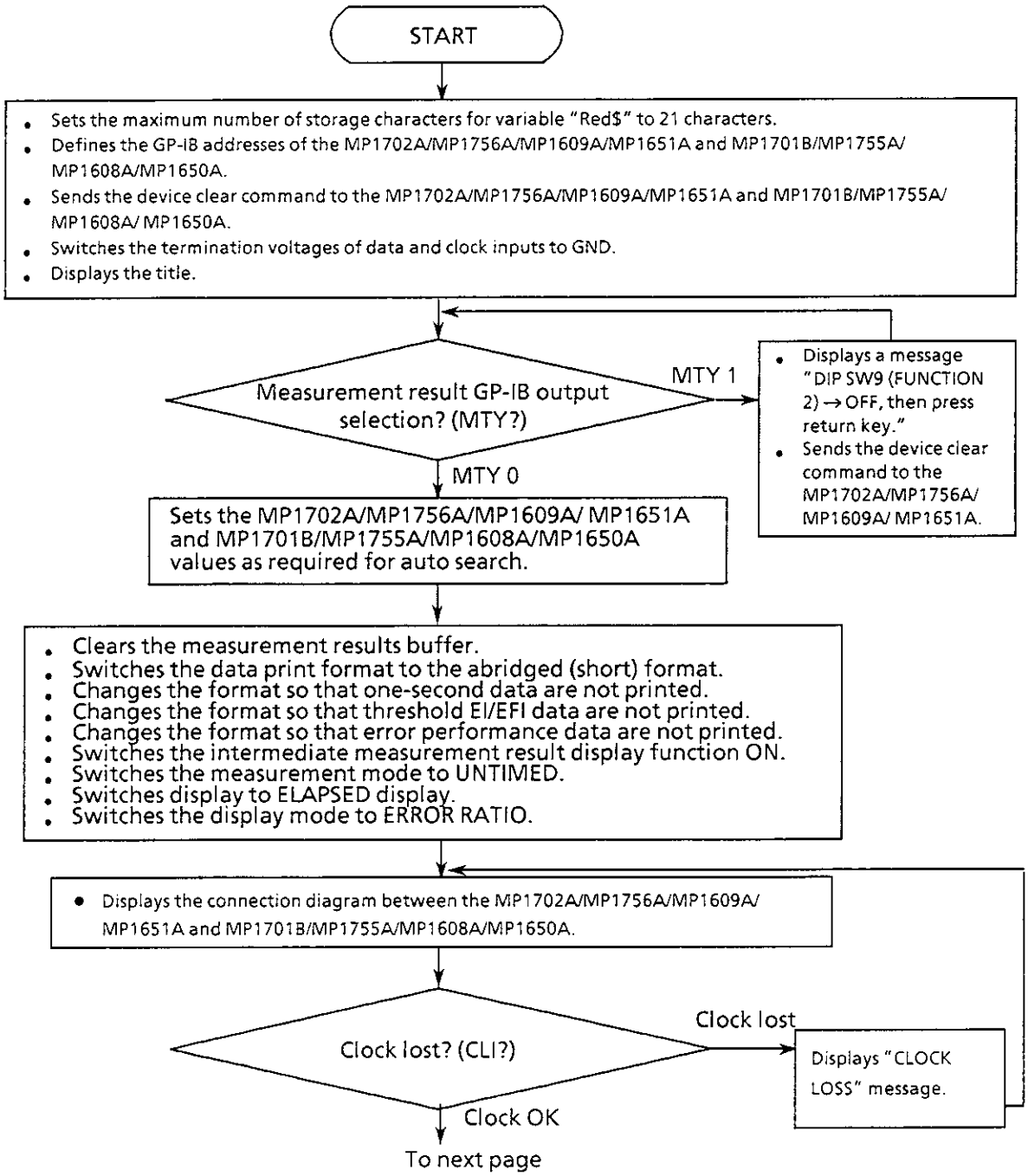
```
1860  MOVE 26,14
1870  IDRAW 21,0
1880  IDRAW 0,9
1890   !
1900  FOR I=0 TO PI*2 STEP PI/12
1910    IDRAW .2*COS(I),.2*SIN(I)
1920  NEXT I
1930   !
1940  MOVE 21,17
1950  IDRAW 0,6
1960   !
1970  FOR I=0 TO PI*2 STEP PI/12
1980    IDRAW .2*COS(I),.2*SIN(I)
1990  NEXT I
2000   !
2010  MOVE 21,17
2020  IDRAW 21,0
2030  IDRAW 0,6
2040   !
2050  FOR I=0 TO PI*2 STEP PI/12
2060    IDRAW .2*COS(I),.2*SIN(I)
2070  NEXT I
2080   !
2090  MOVE 16,25
2100  CSIZE 2.3,.5
2110  LABEL "DATA  CLOCK1       DATA  CLOCK"
2120   !
2130  INPUT "Aer you ready ?  Press return key to start.",A
2140   !
2150  RETURN
2160   !
2170  END
```

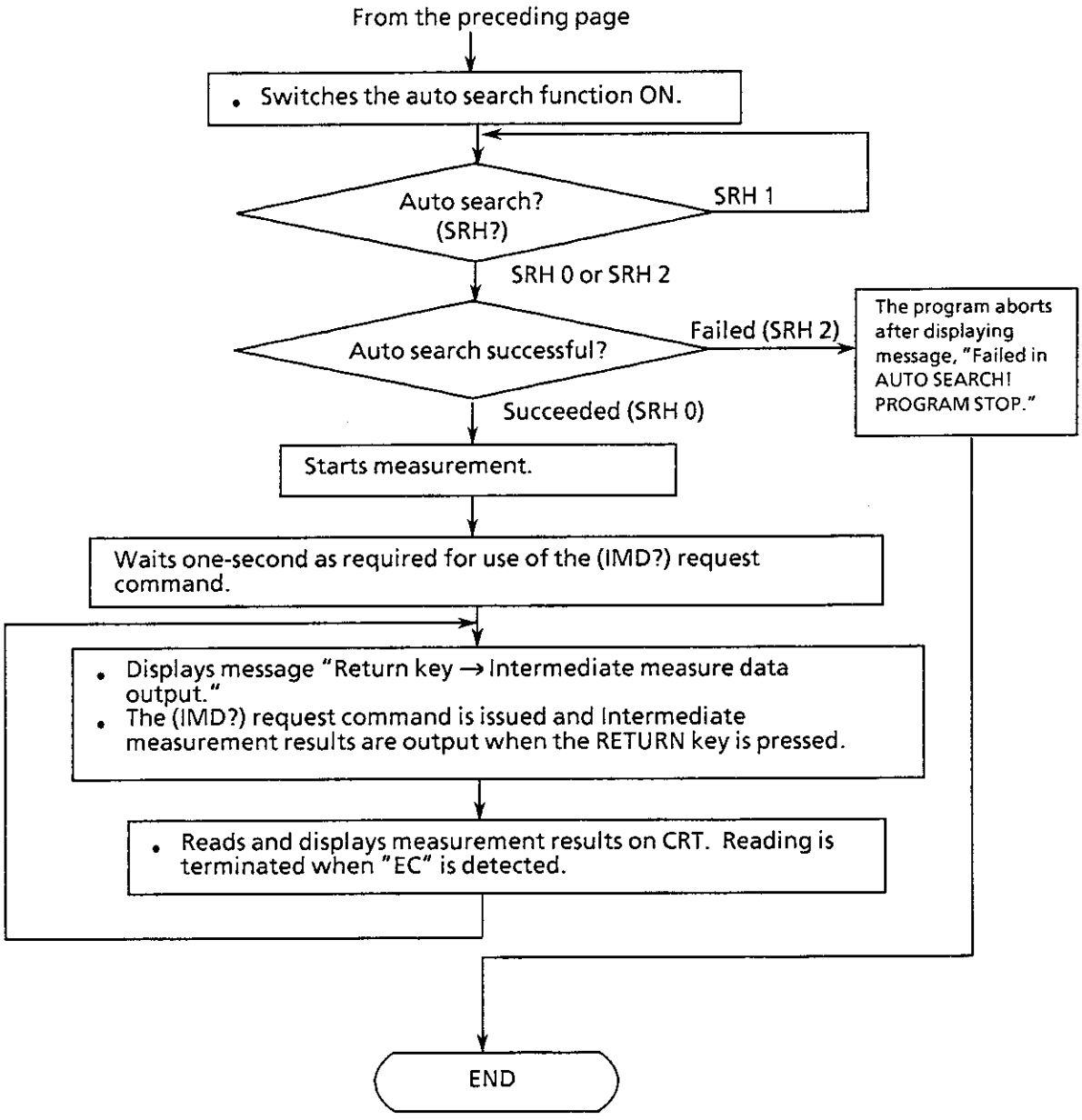
EXECUTED RESULT

```
STA 90,02,26,12,40,37
STA 90,02,26,12,40,37
INT 90,02,26,12,41,40
ELP      0,00,01,03
ER  1.0000E-04
EC     31500064
EI           63
EFI       0.0000
PFI            0
CLI            0
SLI            0
ES      100.0000
EFS       0.0000
SES       0.0000
DM      100.0000
US        0.0000
STA 90,02,26,12,40,37
INT 90,02,26,12,41,46
ELP      0,00,01,09
ER  1.0000E-04
EC     34500070
EI           69
EFI       0.0000
PFI            0
CLI            0
SLI            0
ES      100.0000
EFS       0.0000
SES       0.0000
DM      100.0000
US        0.0000
```

(e) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Standard data are output in short (abridged) format; intermediate measurement results are output in response to (IMD?) request command.




From the preceding page
• Switches the auto search function ON.
Auto search? (SRH?)
SRH 1
SRH 0 or SRH 2
Auto search successful?
Failed (SRH 2)
The program aborts after displaying message, "Failed in AUTO SEARCH! PROGRAM STOP."
Succeeded (SRH 0)
Starts measurement.
Waits one-second as required for use of the (IMD?) request command.
• Displays message "Return key → Intermediate measure data output."
• The (IMD?) request command is issued and Intermediate measurement results are output when the RETURN key is pressed.
• Reads and displays measurement results on CRT. Reading is terminated when "EC" is detected.
END

Program listing

```
10    !*************************************************************
20    !*                                                           *
30    !*     MP1702A/MP1609A/MP1651A  MEASUREMENT  SAMPLE SOFT     *
40    !*          -- Abridged format , Standard output --          *
50    !*             * Intermediate measure data *                 *
60    !*                                                  ED2_M5   *
70    !*************************************************************
80    !
90    !-----------------------------------------------------------!
100   !                        MAIN ROUTINE                       !
110   !-----------------------------------------------------------!
120   !
130   DIM Red$[21]
140   LET Ad1=700               ! Set Device Address (PPG)
150   LET Ad2=701               ! Set Device Address (ED)
160   CLEAR Ad1                 ! Device Clear (PPG)
170   CLEAR Ad2                 ! Device Clear (ED)
180    !
190   OUTPUT Ad2;"DTM 0;CTM 0"
200    !
210   CLEAR SCREEN
220    !
230   PRINT "*** MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT ***"
240   PRINT
250    !
260   GOSUB M_type              ! Check Measurement type
270    !
280   GOSUB D_set               ! Data setting
290   GOSUB Mode                ! Measurement mode
300   GOSUB Clock               ! Check clock loss
310   GOSUB Srch                ! Auto search : ON
320    !
330   OUTPUT Ad2;"STA"          ! Measurement Start
340    !
350   GOSUB Result              ! Display Result
360    !
370   STOP
380    !
390    !-----------------------------------------------------------!
400    !                        SUB ROUTINE                        !
410    !-----------------------------------------------------------!
420    !
430 M_type: !---------------------------- Check Measurement type
440   !
450   LOOP
460      !
470     OUTPUT Ad2;"MTY?"
480     ENTER Ad2;Mty$
490      !
500     IF Mty$="MTY 1" THEN
510       INPUT "DIP SW 9 (Function 2)-->OFF, then press return key",A
520       CLEAR Ad2
530     END IF
540      !
550   EXIT IF Mty$="MTY 0"
560   END LOOP
570    !
580   RETURN
590    !
```

```
600     !
610 D_set:  !------------------------------------------- Set Data
620     !
630    OUTPUT Ad1;"LGC 0;PTN 2;MRK 1"
640    OUTPUT Ad2;"LGC 0;PTN 2;MRK 1"
650     !
660    OUTPUT Ad1;"CLK 1;RES 1;FRQ 3000"
670    OUTPUT Ad1;"OFS 0;SPD 0"
680    OUTPUT Ad1;"DAP 1;DOS 1"
690    OUTPUT Ad1;"CDL 100;CAP 1;COS 1"
700    OUTPUT Ad1;"EAD 1"
710     !
720    OUTPUT Ad2;"CPL 0"
730    OUTPUT Ad2;"SRH 0"
740    OUTPUT Ad2;"SYN 1"
750     !
760    RETURN
770     !
780     !
790 Mode: !------------------------------------ Measurement mode
800     !
810    OUTPUT Ad2;"BCL"          ! Measurement buffer clear
820    OUTPUT Ad2;"FMT 1"        ! Data Output: Abridged format
830    OUTPUT Ad2;"OSC 0"        ! One second data output : OFF
840    OUTPUT Ad2;"THR 0"        ! Threshold output : OFF
850    OUTPUT Ad2;"EPF 0"        ! Error performance Output:OFF
860     !
870    OUTPUT Ad2;"CUR 1"        ! Current data : ON
880    OUTPUT Ad2;"MOD 2"        ! Measurement mode : UNTIMED
890    OUTPUT Ad2;"TIM 4"        ! Display ILAPSED
900    OUTPUT Ad2;"DMS 0"        ! Display ERROR RATIO
910     !
920    RETURN
930     !
940     !
950 Clock: !------------------------------------ Check connection
960     !
970    LOOP
980      GOSUB Connect           ! Display Device connection
990       !
1000     OUTPUT Ad2;"CLI?"       ! REQUEST Clock loss ?
1010     ENTER Ad2;Cli$
1020      !
1030     IF Cli$="CLI 1" THEN
1040       PRINT "** CLOCK LOSS **"
1050     END IF
1060      !
1070   EXIT IF Cli$="CLI 0"
1080   END LOOP
1090    !
1100   RETURN
1110    !
1120    !
1130 Srch:!------------------------------------------- Auto search
1140    !
1150   OUTPUT Ad2;"SRH 1"                          ! AUTO SEARCH : ON
1160    !
1170   LOOP
1180      !
1190     OUTPUT Ad2;"SRH?"
1200     ENTER Ad2;Srh$
1210      !
1220   EXIT IF Srh$="SRH 0" OR Srh$="SRH 2"
1230   END LOOP
```

```
1240   !
1250  IF Srh$="SRH 2" THEN
1260    PRINT "** Failed in AUTO SEARCH !  PROGRAM STOP **"
1270    STOP
1280     !
1290  END IF
1300   !
1310  RETURN
1320   !
1330   !
1340 Result: !---------------------------------------- Print Result
1350   !
1360  WAIT 1
1370   !
1380  LOOP
1390     !
1400    INPUT "Return key --> Intermediate measure data output",A
1410    OUTPUT Ad2;"IMD?"
1420     !
1430    LOOP
1440      ENTER Ad2;Red$
1450      PRINT Red$
1460    EXIT IF Red$[1,2]="EC"
1470    END LOOP
1480     !
1490  END LOOP
1500   !
1510  RETURN
1520   !
1530   !
1540 Connect: !-------------------------------- Display Connection
1550   !
1560  PEN 3
1570  VIEWPORT 70,140,50,100
1580  SHOW 0,70,0,50
1590   !
1600  CLIP 0,70,5,70
1610  FRAME
1620   !
1630  CSIZE 3,.4
1640  MOVE 25,45
1650  LABEL "<< CONNECTION >>"
1660   !
1670  CSIZE 3,.35
1680  MOVE 6,39
1690  LABEL "MP1701B/MP1608A/MP1650A    MP1702A/MP1609A/MP1651A"
1700  !
1710  MOVE 7,20
1720  RECTANGLE 25,18
1730   !
1740  MOVE 38,20
1750  RECTANGLE 25,18
1760   !
1770  MOVE 26,14
1780  IDRAW 0,9
1790   !
1800  FOR I=0 TO PI*2 STEP PI/12
1810    IDRAW .2*COS(I),.2*SIN(I)
1820  NEXT I
1830   !
1840  MOVE 26,14
1850  IDRAW 21,0
1860  IDRAW 0,9
```

```
1870   !
1880  FOR I=0 TO PI*2 STEP PI/12
1890    IDRAW .2*COS(I),.2*SIN(I)
1900  NEXT I
1910   !
1920  MOVE 21,17
1930  IDRAW 0,6
1940   !
1950  FOR I=0 TO PI*2 STEP PI/12
1960    IDRAW .2*COS(I),.2*SIN(I)
1970  NEXT I
1980   !
1990  MOVE 21,17
2000  IDRAW 21,0
2010  IDRAW 0,6
2020   !
2030  FOR I=0 TO PI*2 STEP PI/12
2040    IDRAW .2*COS(I),.2*SIN(I)
2050  NEXT I
2060   !
2070  MOVE 16,25
2080  CSIZE 2.3,.5
2090  LABEL "DATA  CLOCK1       DATA  CLOCK"
2100   !
2110  INPUT "Aer you ready ?  Press return key to start.",A
2120   !
2130  RETURN
2140   !
2150  END
```

EXECUTED RESULT

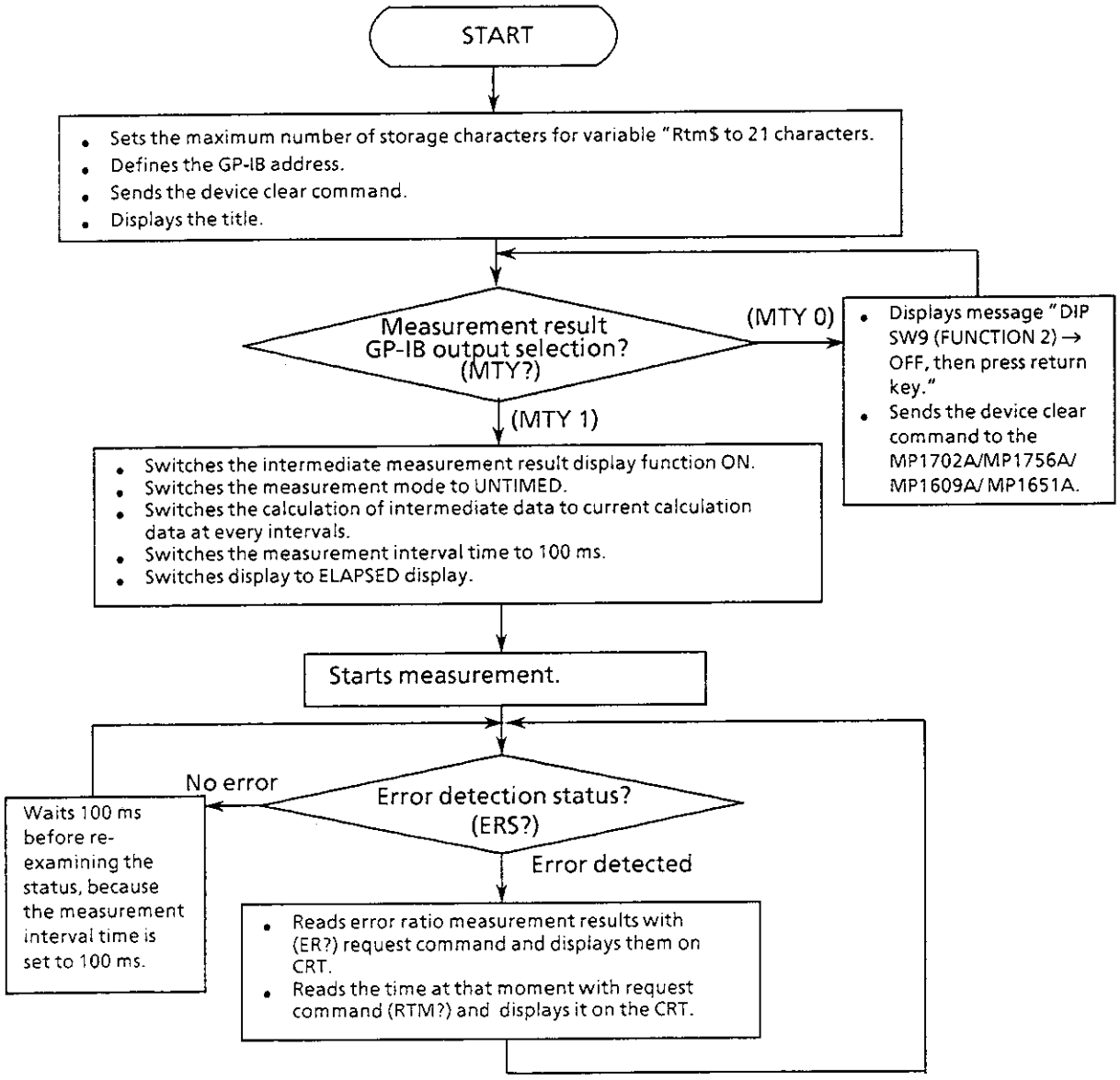
```
STA 90,02,26,12,42,21
INT 90,02,26,12,42,23
ER  1.0000E-04
EC      1000002
INT 90,02,26,12,42,28
ER  1.0000E-04
EC      3500007
```

(6) Measurement (Measurement type: Measurement results are output when a data request command is received.)

(a) Error ratio measurement

This program reads error ratio measurement results when measurement results GP-IB output selection is set for output by the data request command.

Measurement is started after the parameters are appropriately initialized. The program checks the error detection status with the (ERS?) request command. When an error is detected, the error ratio measurement results are read with the (ER?) request command and displayed on CRT along with the time.



Program listing

```
10    !*************************************************************
20    !*                                                           *
30    !*     MP1702A/MP1609A/MP1651A  ERROR RATIO  SAMPLE SOFT     *
40    !*       -- Result output >> Data request command --         *
50    !*                                                           *
60    !*                                                  ED_ERR   *
70    !*************************************************************
80    !
90    !-----------------------------------------------------------!
100   !                        MAIN ROUTINE                       !
110   !-----------------------------------------------------------!
120   !
130   DIM Rtm$[21]
140   LET Add=701               ! Set Device Address (ED)
150   CLEAR Add                 ! Device Clear (ED)
160    !
170   CLEAR SCREEN
180    !
190   PRINT "** MP1702A/MP1609A/MP1651A  ERROR RATIO SAMPLE SOFT **"
200   PRINT
210    !
220   GOSUB M_type              ! Measurement type ?
230    !
240   OUTPUT Add;"CUR 1"        ! Current data : ON
250   OUTPUT Add;"MOD 2"        ! Measurement mode : UNTIMED
260   OUTPUT Add;"CAL 1"        ! Current data caluculation :IMEDIATE
270   OUTPUT Add;"ITV 0"        ! Interval time : 100 ms
280   OUTPUT Add;"DMS 0"        ! Display ERROR RATIO
290    !
300   GOSUB Measure
310    !
320   STOP
330    !
340    !-----------------------------------------------------------!
350    !                          SUB ROUTINE                      !
360    !-----------------------------------------------------------!
370    !
380 M_type:  !--------------------------------- Measurement type ?
390    !
400   LOOP
410    !
420   OUTPUT Add;"MTY?"
430   ENTER Add;Mty$
440    !
450   IF Mty$="MTY 0" THEN
460     INPUT "DIP SW 9 (Function2)--> ON ,then press return key",A
470     CLEAR Add
480   END IF
490    !
500   EXIT IF Mty$="MTY 1"
510   END LOOP
520    !
530   RETURN
540    !
550    !
```

```
560 Measure:    !---------------------------------------- Measurement
570     !
580   OUTPUT Add;"STA"                                ! Measurement Start
590     !
600   LOOP
610       !
620     OUTPUT Add;"ERS?"
630     ENTER Add;Ers$
640       !
650     IF Ers$="ERS 1" THEN
660       GOSUB Result
670     ELSE
680       WAIT .1
690     END IF
700       !
710   END LOOP
720     !
730   RETURN
740     !
750     !
760 Result:   !---------------------------------------- Print Result
770     !
780   OUTPUT Add;"ER?"
790   ENTER Add;Er$
800   PRINT Er$
810     !
820   OUTPUT Add;"RTM?"
830   ENTER Add;Rtm$
840   PRINT Rtm$
850     !
860     !
870   RETURN
880     !
890     !
900   END
```

EXECUTED RESULT

```
ER  1.0000E-04
RTM 90,02,26,12,54,33
ER  1.0000E-04
RTM 90,02,26,12,54,33
ER  1.0000E-04
RTM 90,02,26,12,54,33
ER  1.0000E-04
RTM 90,02,26,12,54,33
ER  1.0000E-04
RTM 90,02,26,12,54,34
ER  1.0000E-04
RTM 90,02,26,12,54,34
ER  1.0000E-04
RTM 90,02,26,12,54,34
ER  1.0000E-04
RTM 90,02,26,12,54,34
ER  1.0000E-04
RTM 90,02,26,12,54,34
ER  1.0000E-04
RTM 90,02,26,12,54,35
ER  1.0000E-04
RTM 90,02,26,12,54,35
ER  1.0000E-04
RTM 90,02,26,12,54,35
ER  1.0000E-04
RTM 90,02,26,12,54,35
ER  1.0000E-04
RTM 90,02,26,12,54,35
ER  1.0000E-04
RTM 90,02,26,12,54,35
ER  1.0000E-04
RTM 90,02,26,12,54,36
```

(b) Measurement result output

This program was written for applications in which the measurement result GP-IB output selection is set for output by data request commands.

Measurement is initiated after connecting DATA to DATA and CLOCK1 to CLOCK between the MP1702A/MP1756A/MP1609A/MP1651A and MP1701B/MP1755A/MP1608A/MP1650A.

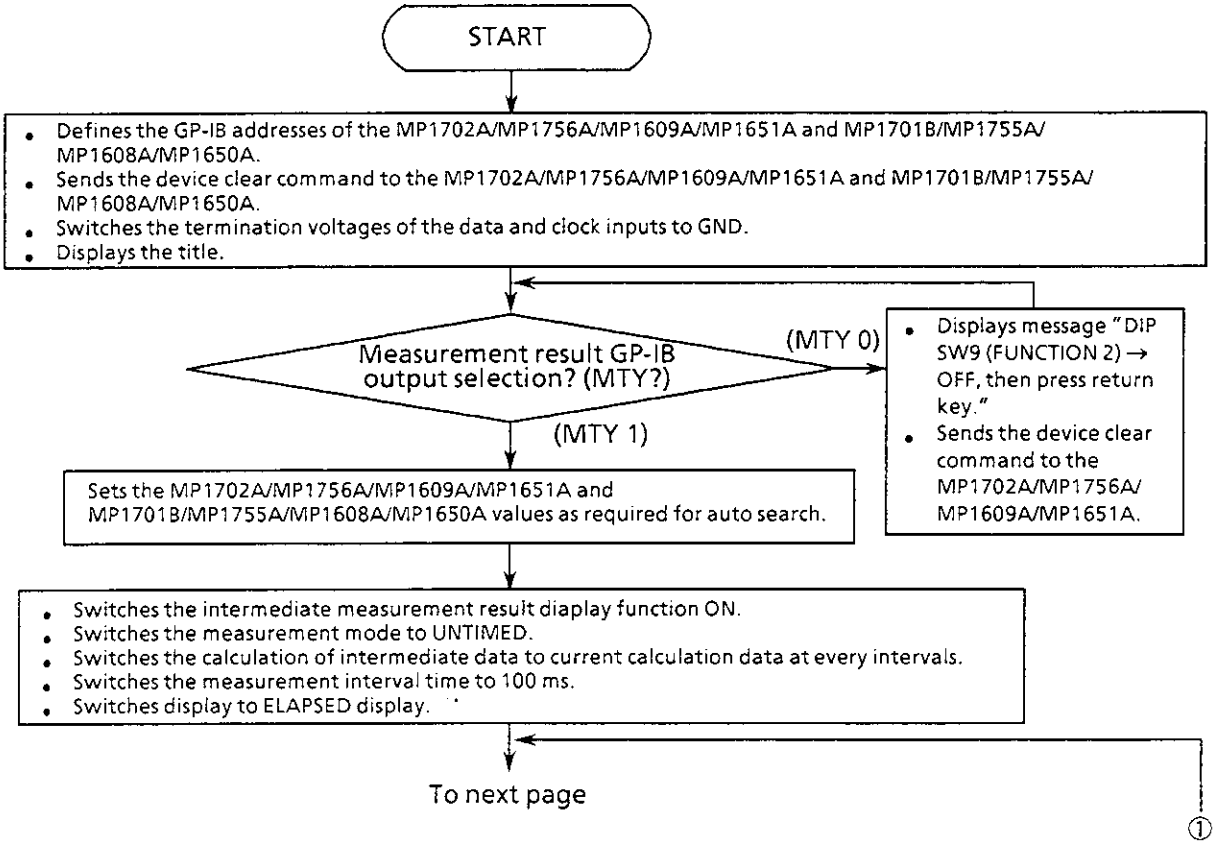
First, the program initializes the parameters as necessary for auto search and measurement.

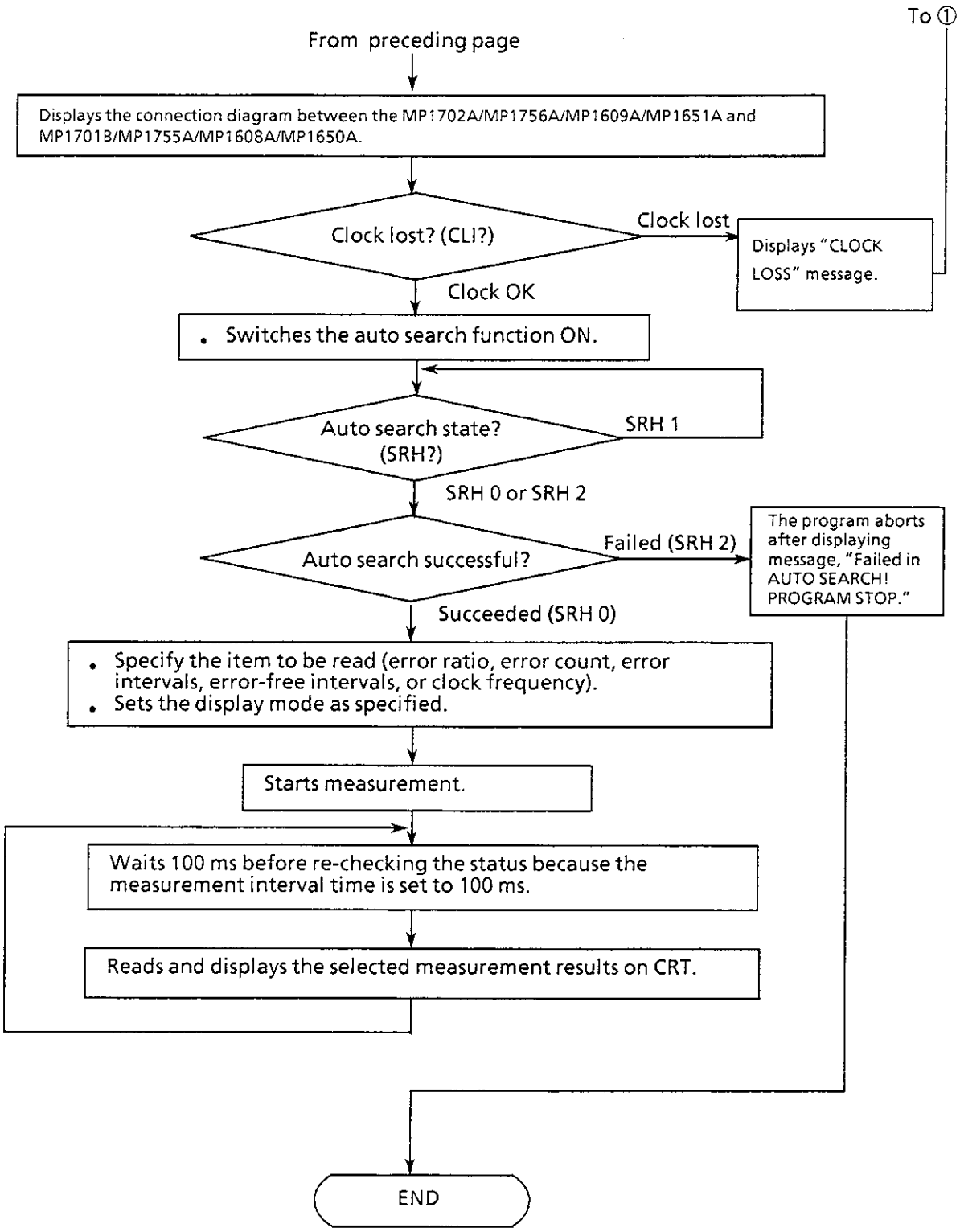
Next, it confirms that the clock is not lost and, after confirmation, it initiates an auto search.

If the auto search is completed successfully, the item to be read (error ratio, error count, error intervals, error-free intervals, or clock frequency) is selected and the program starts measurement after setting the display mode. After waiting 100 ms, the measurement results of the selected item are read via a request command and displayed on CRT.

On the other hand, if auto search fails, the program aborts.

Note: When using the MP1755A for the transmitter, internal clock frequency setting is ineffective. Set the frequency of the external clock source to 3000 MHz in advance.




To ①
From preceding page
Displays the connection diagram between the MP1702A/MP1756A/MP1609A/MP1651A and MP1701B/MP1755A/MP1608A/MP1650A.
Clock lost? (CLI?)
Clock lost
Displays "CLOCK LOSS" message.
Clock OK
• Switches the auto search function ON.
Auto search state? (SRH?)
SRH 1
SRH 0 or SRH 2
Auto search successful?
Failed (SRH 2)
The program aborts after displaying message, "Failed in AUTO SEARCH! PROGRAM STOP."
Succeeded (SRH 0)
• Specify the item to be read (error ratio, error count, error intervals, error-free intervals, or clock frequency).
• Sets the display mode as specified.
Starts measurement.
Waits 100 ms before re-checking the status because the measurement interval time is set to 100 ms.
Reads and displays the selected measurement results on CRT.
END

Program listing

```
10    !*************************************************************
20    !*                                                           *
30    !*     MP1702A/MP1609A/MP1651A  MEASUREMENT  SAMPLE SOFT     *
40    !*       -- Result output >> Data request command --         *
50    !*                                                           *
60    !*                                                  ED_MTY   *
70    !*************************************************************
80    !
90    !-----------------------------------------------------------!
100   !                        MAIN ROUTINE                       !
110   !-----------------------------------------------------------!
120   !
130   LET Ad1=700               ! Set Device Address (PPG)
140   LET Ad2=701               ! Set Device Address (ED)
150   CLEAR Ad1                 ! Device Clear (PPG)
160   CLEAR Ad2                 ! Device Clear (ED)
170    !
180   OUTPUT Ad2;"DTM 0;CTM 0"
190    !
200   CLEAR SCREEN
210    !
220   PRINT "** MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT **"
230   PRINT
240    !
250   GOSUB M_type              ! Check Measurement type
260   GOSUB D_set               ! Set data
270   GOSUB Mode                ! Measurement mode
280   GOSUB Clock               ! Check clock loss
290   GOSUB Srch                ! Auto search : ON
300   GOSUB Slct                ! Select Output Result
310    !
320   OUTPUT Ad2;"STA"          ! Measurement start
330    !
340   GOSUB Result              ! Display result
350    !
360   STOP
370    !
380    !-----------------------------------------------------------!
390    !                          SUB ROUTINE                      !
400    !-----------------------------------------------------------!
410    !
420 M_type:  !---------------------------- Check Measurement type
430    !
440   LOOP
450     OUTPUT Ad2;"MTY?"
460     ENTER Ad2;Mty$
470      !
480     IF Mty$="MTY 0" THEN
490       INPUT "DIP SW 9 (Function 2)--> ON ,then press return key",A
500       CLEAR Ad2
510     END IF
520      !
530   EXIT IF Mty$="MTY 1"
540   END LOOP
550    !
560   RETURN
570    !
```

```
580 D_set: !------------------------------------------- Set data
590    !
600    OUTPUT Ad1;"LGC 0;PTN 2;MRK 1"
610    OUTPUT Ad2;"LGC 0;PTN 2;MRK 1"
620    !
630    OUTPUT Ad1;"EAD 1"
640    OUTPUT Ad1;"CLK 1;RES 1;FRQ 3000"
650    OUTPUT Ad1;"OFS 0;SPD 0"
660    OUTPUT Ad1;"DAP 1;DOS 1"
670    OUTPUT Ad1;"CDL 100;CAP 1;COS 1"
680    !
690    OUTPUT Ad2;"CPL 0"
700    OUTPUT Ad2;"SRH 0"
710    OUTPUT Ad2;"SYN 1"
720    !
730    RETURN
740    !
750 Mode:!------------------------------------- Measurement mode
760    !
770    OUTPUT Ad2;"CUR 1"    ! Current data : ON
780    OUTPUT Ad2;"MOD 2"    ! Measurement mode : UNTIMED
790    OUTPUT Ad2;"CAL 1"    ! Current data caluculation : IMEDIATE
800    OUTPUT Ad2;"ITV 0"    ! Interval time : 100 ms
810    OUTPUT Ad2;"TIM 4"    ! Display elapsed time
820    !
830    RETURN
840    !
850 Clock:!------------------------------------- Check connection
860    !
870    LOOP
880      !
890      GOSUB Connect                    ! Display Device connection
900      !
910      OUTPUT Ad2;"CLI?"                ! REQUEST Clock loss ?
920      ENTER Ad2;Cli$
930      IF Cli$="CLI 1" THEN PRINT "** CLOCK LOSS **"
940      !
950    EXIT IF Cli$="CLI 0"
960    END LOOP
970    !
980    RETURN
990    !
1000 Srch:!------------------------------------------ Auto search
1010   !
1020   OUTPUT Ad2;"SRH 1"                          ! AUTO SEARCH : ON
1030   !
1040   LOOP
1050     !
1060     OUTPUT Ad2;"SRH?"
1070     ENTER Ad2;Srh$
1080     !
1090   EXIT IF Srh$="SRH 0" OR Srh$="SRH 2"
1100   END LOOP
1110   !
1120   IF Srh$="SRH 2" THEN
1130     PRINT "** Failed in AUTO SEARCH !  PROGRAM STOP **"
1140     STOP
1150   END IF
1160   !
1170   RETURN
1180   !
1190   !
```

```
1200 Slct: !---------------------------------- Select output result
1210   !
1220  PRINT "OUTPUT DATA SELECT  * ERROR RATIO = [0]"
1230  PRINT "                    * ERROR COUNT = [1]"
1240  PRINT "                    * ERROR INTERVALS = [2]"
1250  PRINT "                    * ERROR FREE INTERVALS = [3]"
1260  PRINT "                    * CLOCK FREQUENCY = [4]"
1270  PRINT
1280   !
1290  LOOP
1300    INPUT "SELECT OUTPUT DATA [ 0 ~ 4 ] ?",Dta
1310  EXIT IF Dta>=0 AND Dta<=4
1320  END LOOP
1330   !
1340  SELECT Dta
1350   !
1360  CASE 0
1370    Dms$="0"
1380    Wrt$="ER?"
1390     !
1400  CASE 1
1410    Dms$="1"
1420    Wrt$="EC?"
1430     !
1440  CASE 2
1450    Dms$="2"
1460    Wrt$="EI?"
1470     !
1480  CASE 3
1490    Dms$="3"
1500    Wrt$="EFI?"
1510     !
1520  CASE 4
1530    Dms$="4"
1540    Wrt$="FRQ?"
1550     !
1560  END SELECT
1570   !
1580  OUTPUT Ad2;"DMS "&Dms$              ! Display measurement mode
1590   !
1600  RETURN
1610   !
1620   !
1630 Result:  !------------------------------------ Print result
1640   !
1650  LOOP
1660     !
1670    WAIT .1
1680    OUTPUT Ad2;Wrt$
1690    ENTER Ad2;Red$
1700    PRINT Red$
1710     !
1720  END LOOP
1730   !
1740  RETURN
1750   !
1760   !
1770 Connect: !-------------------------------- Display Connection
1780   !
1790  PEN 3
1800  VIEWPORT 70,140,50,100
1810  SHOW 0,70,0,50
1820   !
```

```
1830  CLIP 0,70,0,70
1840  FRAME
1850   !
1860  CSIZE 3,.4
1870  MOVE 25,45
1880  LABEL "<< CONNECTION >>"
1890   !
1900  CSIZE 3,.35
1910  MOVE 6,39
1920  LABEL "MP1701B/MP1608A/MP1650A     MP1702A/MP1609A/MP1651A"
1930  !
1940  MOVE 7,20
1950  RECTANGLE 25,18
1960   !
1970  MOVE 38,20
1980  RECTANGLE 25,18
1990   !
2000  MOVE 26,14
2010  IDRAW 0,9
2020   !
2030  FOR I=0 TO PI*2 STEP PI/12
2040    IDRAW .2*COS(I),.2*SIN(I)
2050  NEXT I
2060   !
2070  MOVE 26,14
2080  IDRAW 21,0
2090  IDRAW 0,9
2100   !
2110  FOR I=0 TO PI*2 STEP PI/12
2120    IDRAW .2*COS(I),.2*SIN(I)
2130  NEXT I
2140   !
2150  MOVE 21,17
2160  IDRAW 0,6
2170   !
2180  FOR I=0 TO PI*2 STEP PI/12
2190    IDRAW .2*COS(I),.2*SIN(I)
2200  NEXT I
2210   !
2220  MOVE 21,17
2230  IDRAW 21,0
2240  IDRAW 0,6
2250   !
2260  FOR I=0 TO PI*2 STEP PI/12
2270    IDRAW .2*COS(I),.2*SIN(I)
2280  NEXT I
2290   !
2300  MOVE 16,25
2310  CSIZE 2.3,.5
2320  LABEL "DATA  CLOCK1       DATA  CLOCK"
2330   !
2340  INPUT "Aer you ready ?  Press return key to start.",A
2350   !
2360  RETURN
2370   !
2380  END
```

EXECUTED RESULT

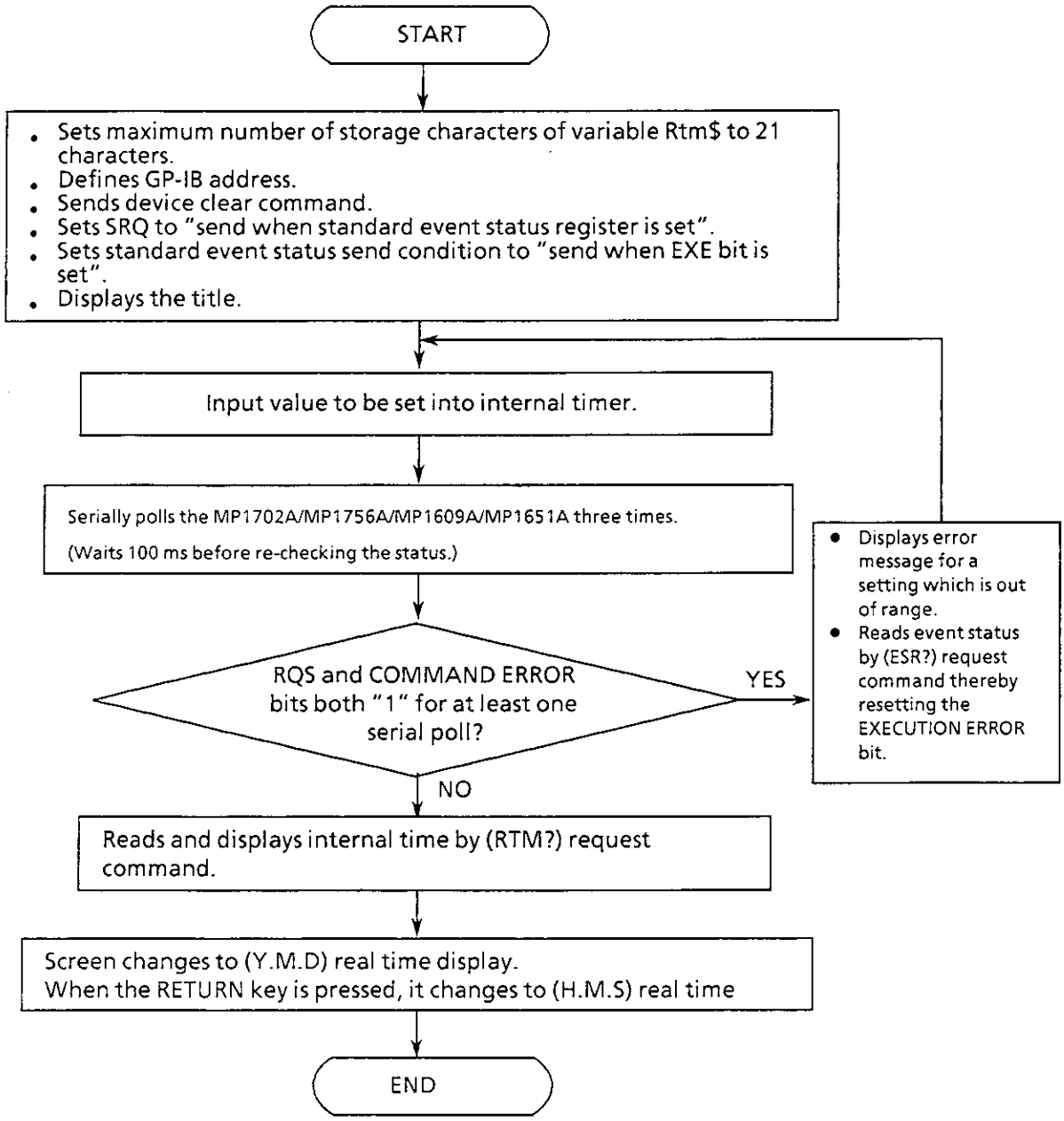
[ERROR RATIO]	[ERROR COUNT]	[ERROR INTERVALS]
ER 1.0000E-04	EC 50001	EI 1.0000E-99
ER 1.0000E-04	EC 50000	EI 1.0000E-99
ER 1.0000E-04	EC 50000	EI 1.0000E-99
ER 1.0000E-04	EC 50000	EI 1.0000E-99
ER 1.0000E-04	EC 50000	EI 1.0000E-99
ER 1.0000E-04	EC 50000	EI 1.0000E-99
ER 1.0000E-04	EC 50001	EI 1
ER 1.0000E-04	EC 50000	EI 1
ER 1.0000E-04	EC 50000	EI 1
ER 1.0000E-04	EC 50000	EI 1
ER 1.0000E-04	EC 50000	EI 1
ER 1.0000E-04	EC 50000	EI 1
ER 1.0000E-04	EC 50001	EI 2
ER 1.0000E-04	EC 50000	EI 2
ER 1.0000E-04	EC 50000	EI 2
ER 1.0000E-04	EC 50000	EI 2
ER 1.0000E-04	EC 50000	EI 2
ER 1.0000E-04	EC 50000	EI 2
ER 1.0000E-04	EC 50000	EI 2
		EI 3
		EI 3

[ERROR FREE INTERVALS]	[CLOCK FREQUENCY]
EFI 1.0000E-99	FRQ 3000.009
EFI 1.0000E-99	FRQ 3000.009
EFI 1.0000E-99	FRQ 3000.009
EFI 1.0000E-99	FRQ 3000.009
EFI 1.0000E-99	FRQ 3000.009
EFI 1.0000E-99	FRQ 3000.009
EFI 0.0000	FRQ 3000.009
EFI 0.0000	FRQ 3000.009
EFI 0.0000	FRQ 3000.009
EFI 0.0000	FRQ 3000.009
EFI 0.0000	FRQ 3000.009
EFI 0.0000	FRQ 3000.009
EFI 0.0000	FRQ 3000.009
EFI 0.0000	FRQ 3000.009
EFI 0.0000	FRQ 3000.009
EFI 0.0000	FRQ 3000.009
EFI 0.0000	FRQ 3000.009
EFI 0.0000	

(7) Internal timer setting

This program uses the (RTM?) data request command to read a value for the internal timer. This value is then set, and displayed on the CRT.

The range of the input value is checked via the standard event status byte.



Program listing

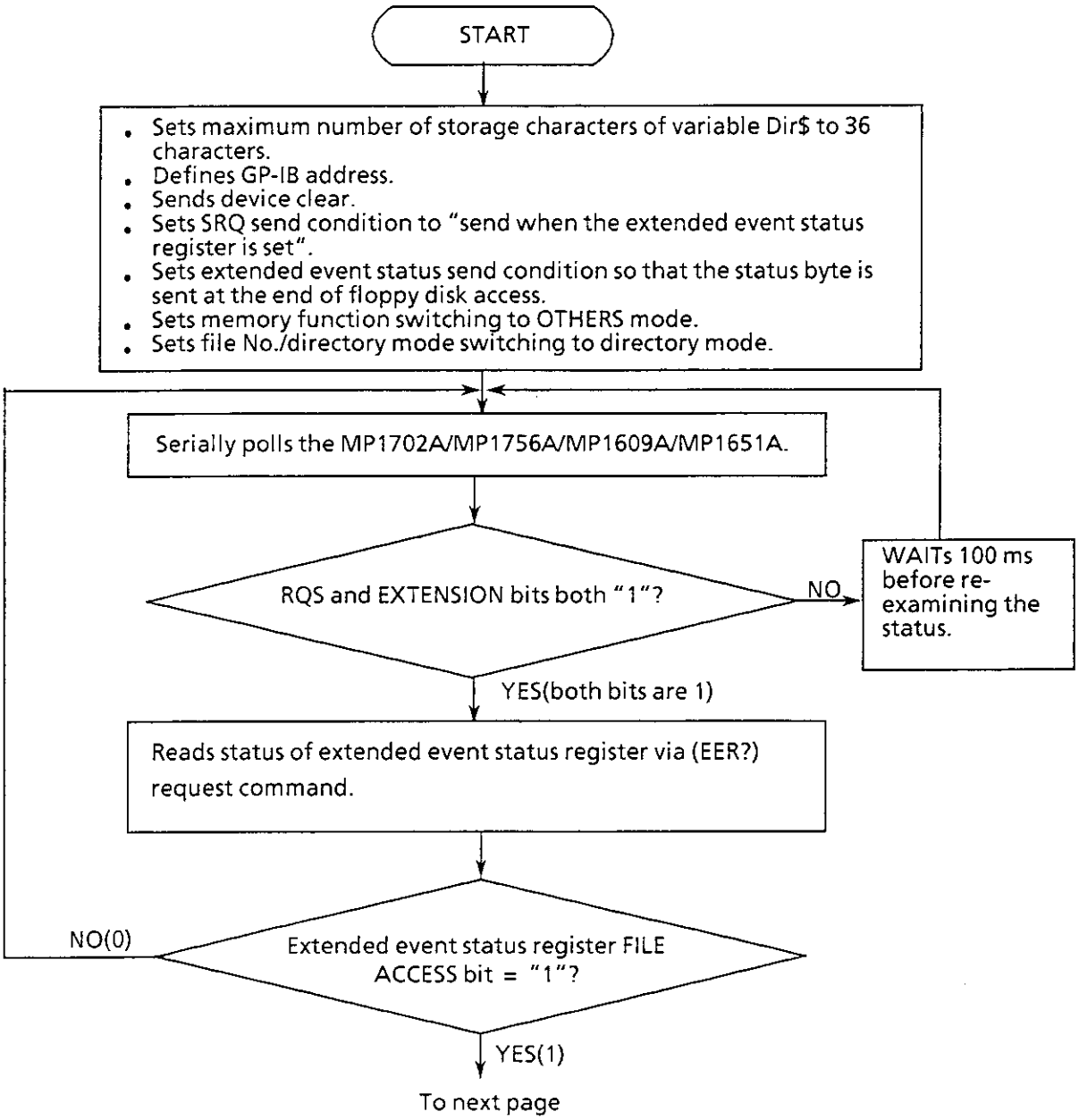
```
10    !*****************************************************************
20    !*                                                               *
30    !*  MP1702A/MP1609A/MP1651A   REAL TIME SETTING SAMPLE SOFT      *
40    !*                                                               *
50    !*                                                      ED_RTM   *
60    !*****************************************************************
70    !
80    !---------------------------------------------------------------!
90    !                          MAIN  ROUTINE                        !
100   !---------------------------------------------------------------!
110   !
120   DIM Rtm$[21]
130   LET Add=701               ! Set Device address
140   CLEAR Add                 ! Device clear
150   OUTPUT Add;"SRQ 32"       ! SQR : COMMAND ERROR bit
160   OUTPUT Add;"ESE 16"       ! Execution error
170    !
180   PRINT "* MP1702A/MP1609A/MP1651A  REAL TIME SETTING SAMPLE SOFT *"
190   PRINT
200    !
210   GOSUB D_set
220   GOSUB Dsply
230    !
240   STOP
250    !
260    !---------------------------------------------------------------!
270    !                          SUB  ROUTINE                         !
280    !---------------------------------------------------------------!
290    !
300 D_set:  !------------------------------------ Set real time
310    !
320   LOOP
330      !
340     INPUT "REAL TIME SETTING DATA  ** YEAR ** [ 0~99 ]",Y$
350     INPUT "REAL TIME SETTING DATA  ** MONTH ** [ 1~12 ]",M$
360     INPUT "REAL TIME SETTING DATA  ** DAY ** [ 1~31 ]",D$
370     INPUT "REAL TIME SETTING DATA  ** HOURE ** [ 0~23 ]",H$
380     INPUT "REAL TIME SETTING DATA  ** MINUTE ** [ 0~59 ]",Mi$
390     INPUT "REAL TIME SETTING DATA  ** SECOND ** [ 0~59 ]",S$
400      !
410     Rtm$=Y$&","&M$&","&D$&","&H$&","&Mi$&","&S$
420     OUTPUT Add;"RTM "&Rtm$
430      !
440     GOSUB Check
450      !
460   EXIT IF Result$="OK"
470    !
480   END LOOP
490    !
500   RETURN
510    !
520    !
530 Dsply:  !---------------------------------- Display real time
540    !
550   OUTPUT Add;"RTM?"
560    !
570   ENTER Add;Rtm$
580   PRINT Rtm$
590    !
```

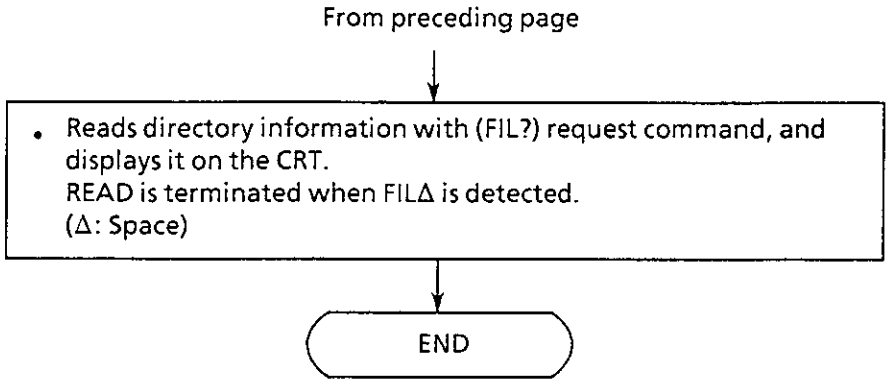
```
600   OUTPUT Add;"TIM 0"                  ! Real time display  Y,M,D
610   INPUT " Press return key  to display [ H,M,S ] ",A
620   OUTPUT Add;"TIM 1"                  ! Real time display  H,M,S
630    !
640   RETURN
650    !
660    !
670 Check:  !------------------------------------ Check Set Data
680    !
690   Byt=0
700    !
710   FOR I=0 TO 2
720      !
730     A=SPOLL(Add)                          ! Serial polling
740      !
750     IF BIT(A,6)=1 AND BIT(A,5)=1 THEN  ! RQS,COMMAND ERROR bit=1
760       Byt=A
770     END IF
780      !
790     WAIT .1
800      !
810   NEXT I
820    !
830   IF BIT(Byt,6)=1 AND BIT(Byt,5)=1 THEN
840     PRINT "EXECUTION ERROR !!   INPUT AGAIN "
850      !
860     OUTPUT Add;"ESR?"              ! RESET Execution error bit
870      !
880   ELSE
890     Result$="OK"
900      !
910   END IF
920    !
930   RETURN
940    !
950    !
960   END
```

(8) Floppy disk file information read
(Floppy disk access status check by serial polling)

This program checks the directory information of the files stored on the floppy disk, and displays it on the CRT. It also shows an example of actual output.

Floppy disk access status check is performed by serial polling and by data request command that checks the extended event status information.





EXECUTED RESULT

```
R00      ,PTN,    65664,89-12-14,21:05
R01      ,PTN,     1920,89-12-14,20:13
R01      ,OTH,      144,90-01-18,18:54
R03      ,PTN,      130,89-12-15,20:18
R03      ,OTH,      144,90-01-18,18:55
R04      ,PTN,     6016,90-01-23,10:59
R05      ,PTN,    65664,90-01-11,13:06
R05      ,OTH,      144,89-12-28,14:25
R06      ,PTN,    65664,90-01-11,13:08
R07      ,PTN,     2176,90-01-11,13:09
R08      ,PTN,     2176,90-01-11,13:10
R10      ,PTN,    65664,90-01-29,21:08
R50      ,PTN,      144,90-01-16,11:10
R50      ,OTH,      144,90-01-18,18:55
R80      ,PTN,      132,90-01-29,21:08
FIL 1
```

Program listing

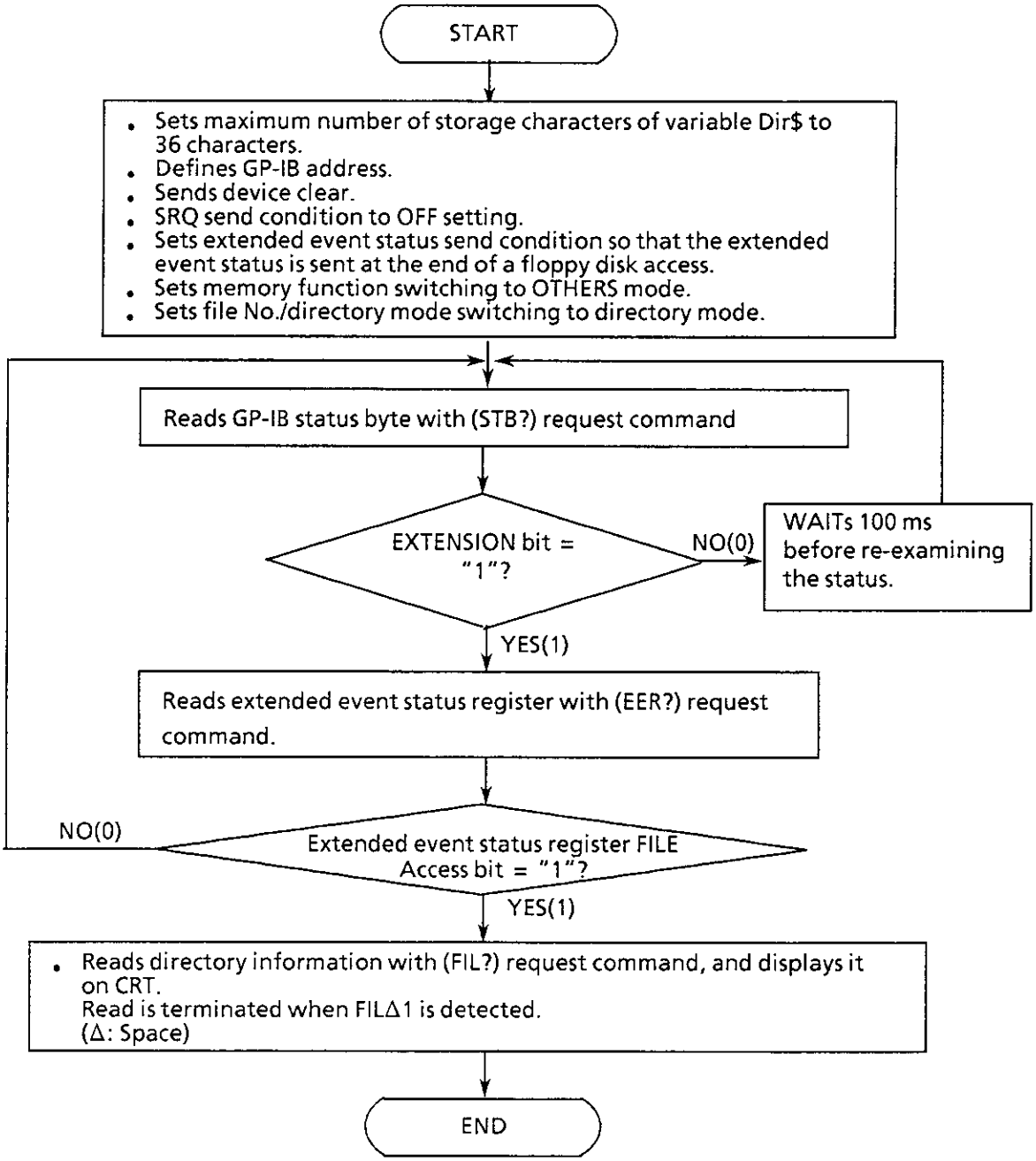
```
10    !*****************************************************************
20    !*                                                               *
30    !*                MP1702A / MP1609A / MP1651A                    *
40    !*            FILE DIRECTORY READ  SAMPLE SOFT_1                 *
50    !*                                                               *
60    !*                                                      ED_DIR   *
70    !*****************************************************************
80    !
90    !---------------------------------------------------------------!
100   !                          MAIN ROUTINE                         !
110   !---------------------------------------------------------------!
120   !
130   DIM Dir$[36]
140   LET Add=701               ! Set Device Address
150   CLEAR Add                 ! Device Clear
160    !
170   OUTPUT Add;"SRQ 2"        ! SRQ : Extension bit
180   OUTPUT Add;"EES 32"       ! EES : Floppy Access End
190   OUTPUT Add;"MEM 1"        ! Memory mode : OTHERS
200   OUTPUT Add;"FIL 1"        ! FIL : Directory Mode
210    !
220   GOSUB Spoll_eer
230   GOSUB File_dir
240    !
250   STOP
260    !
270    !---------------------------------------------------------------!
280    !                            SUB ROUTINE                        !
290    !---------------------------------------------------------------!
300    !
310 Spoll_eer:!---------------------------------- Check Status Byte
320    !
330   LOOP
340      !
350     LOOP
360        !
370       A=SPOLL(Add)          ! Send Serial Poll
380        !
390     EXIT IF BIT(A,6)=1 AND BIT(A,1)=1 ! RQS bit,Extension bit = 1
400       WAIT .1
410        !
420     END LOOP
430      !
440     OUTPUT Add;"EER?"       ! REQUEST Extension Event Register ?
450     ENTER Add;Eer$          ! READ    Extension Event Register
460      !
470   EXIT IF Eer$[9,9]="1"     ! File Access bit = 1
480    !
490   END LOOP
500    !
510   RETURN
520    !
530    !
540 File_dir:! ---------------------------------- Read File Directory
550    !
560   OUTPUT Add;"FIL?"         ! REQUEST Directory ?
570    !
580   LOOP
590      !
600     ENTER Add;Dir$          ! READ    Directory
610     PRINT Dir$
```

```
620       !
630   EXIT IF Dir$="FIL 1"
640     !
650   END LOOP
660     !
670   RETURN
680     !
690     !
700   END
```

(9) Floppy disk file information read
(Floppy disk access status check: by (STB?) request command)

This program reads and displays the directory information for files stored on floppy disk. It also shows an actual example of program output.

Floppy disk access is checked with data request commands that monitor the GP-IB status byte and the extended event staus byte.



Program listing

```
10    !*************************************************************
20    !*                                                           *
30    !*              MP1702A / MP1609A / MP1651A                  *
40    !*          FILE DIRECTORY READ  SAMPLE SOFT_2               *
50    !*                                                           *
60    !*                                                 ED_DIR2   *
70    !*************************************************************
80    !
90    !-----------------------------------------------------------!
100   !                          MAIN ROUTINE                     !
110   !-----------------------------------------------------------!
120    !
130   DIM Dir$[36]
140   LET Add=701                    ! Set Device Address
150   CLEAR Add                      ! Device Clear
160    !
170   OUTPUT Add;"SRQ 0"             ! SRQ : OFF
180   OUTPUT Add;"EES 32"            ! EES : Floppy Access End
190   OUTPUT Add;"MEM 1"             ! Memory mode : OTHERS
200   OUTPUT Add;"FIL 1"             ! FIL : Directory Mode
210    !
220   GOSUB Stb_reg
230   GOSUB File_dir
240    !
250   STOP
260    !
270    !
280    !-----------------------------------------------------------!
290    !                          SUB ROUTINE                      !
300    !-----------------------------------------------------------!
310    !
320 Stb_reg:!------------------------------------ Check Status Byte
330    !
340   LOOP
350      !
360     LOOP
370        !
380       OUTPUT Add;"STB?"           ! REQUEST Status Byte Register ?
390       ENTER Add;Stb$              ! READ    Status Byte Register
400        !
410     EXIT IF Stb$[13,13]="1"       ! Extention bit = 1
420       WAIT .1
430        !
440     END LOOP
450      !
460     OUTPUT Add;"EER?"             ! REQUEST Extension Event Register ?
470     ENTER Add;Eer$                ! READ    Extension Event Register
480      !
490   EXIT IF Eer$[9,9]="1"           ! File Access bit = 1
500    !
510   END LOOP
520    !
530   RETURN
540    !
550    !
560 File_dir:  ! ------------------------------ Read File Directory
570    !
580   OUTPUT Add;"FIL?"               ! REQUEST Directory
590    !
```

```
600   LOOP
610      !
620     ENTER Add;Dir$              ! READ    Directory
630     PRINT Dir$
640      !
650   EXIT IF Dir$="FIL 1"
660    !
670   END LOOP
680    !
690   RETURN
700    !
710   END
```

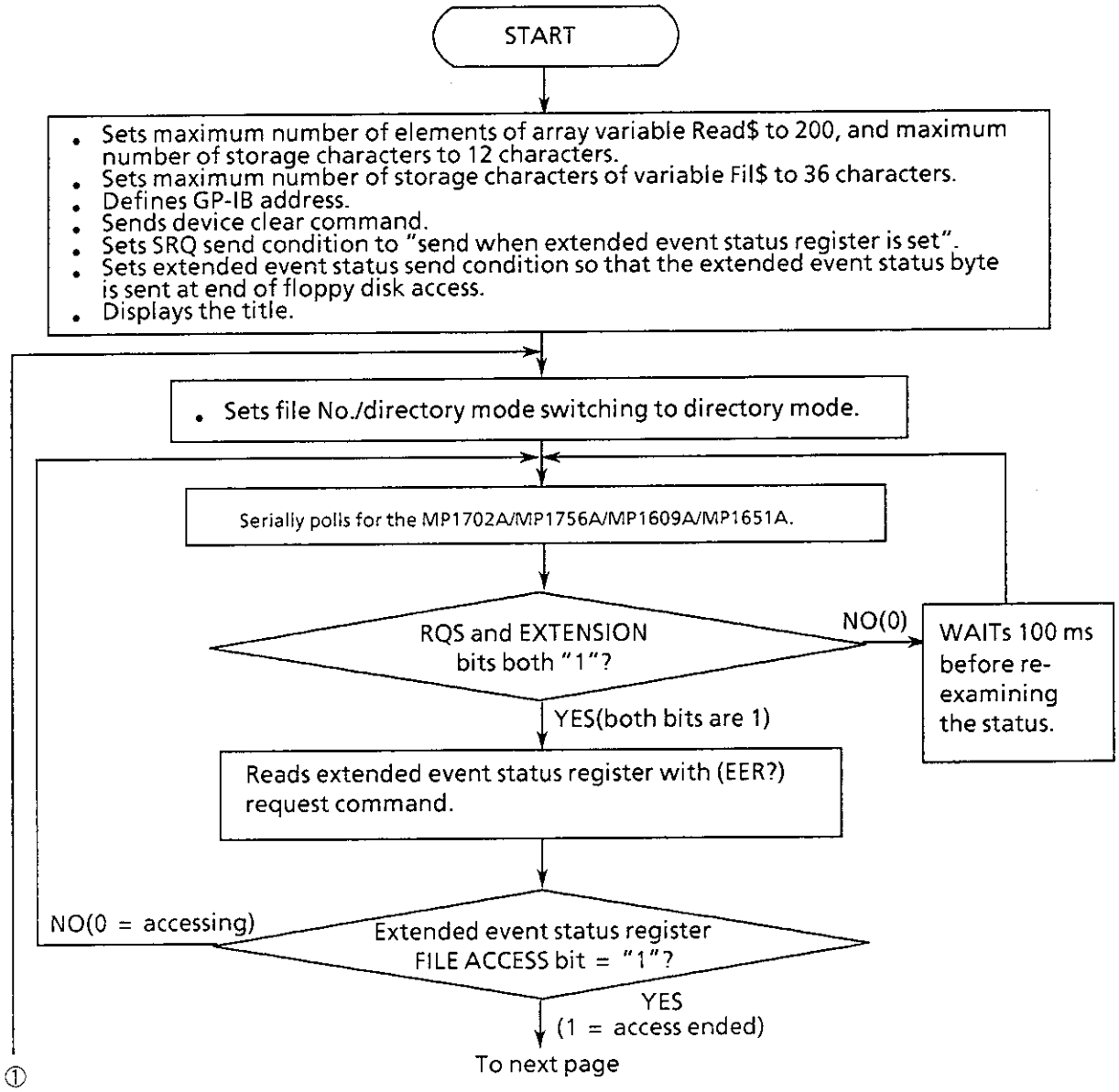
EXECUTED RESULTS

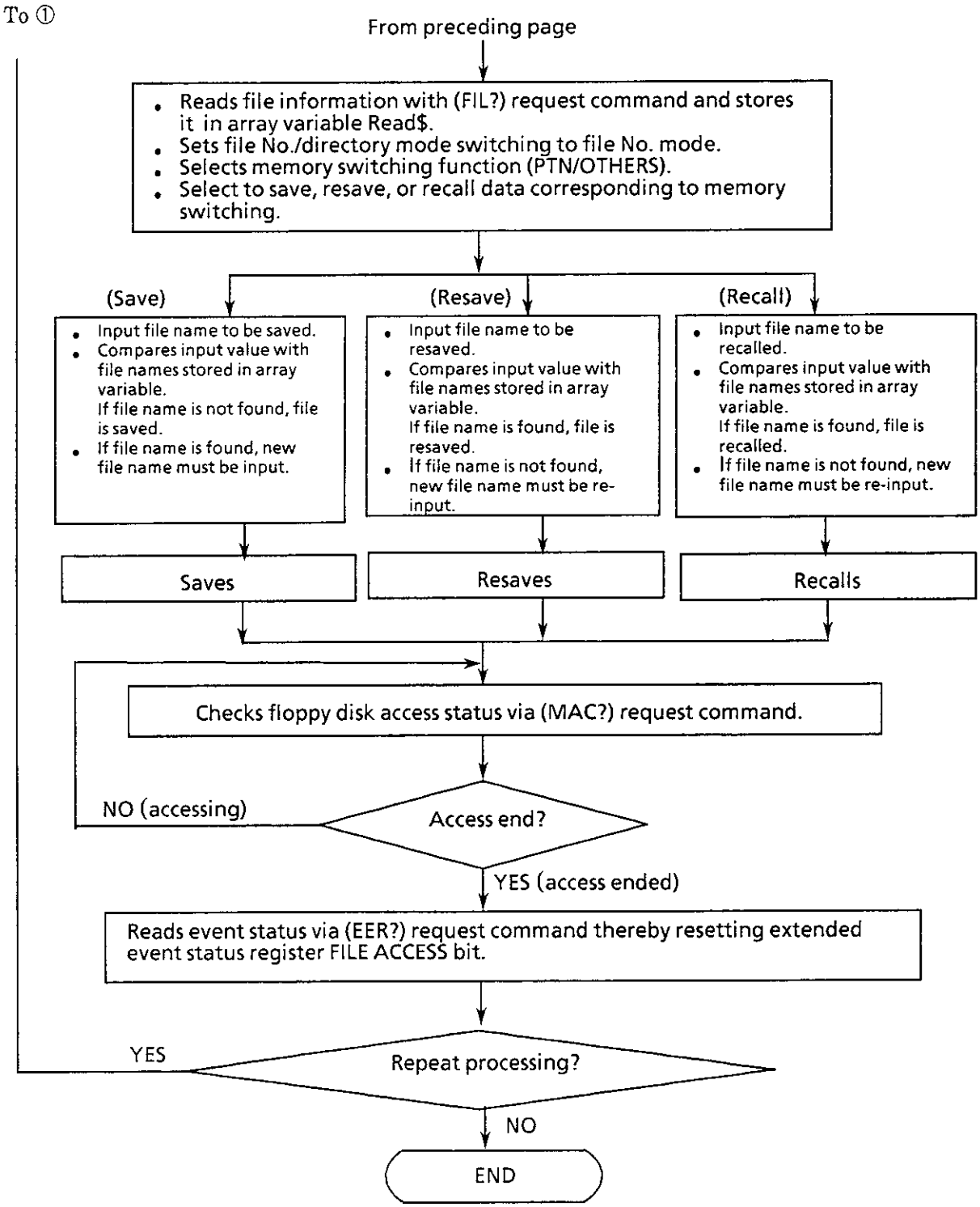
```
R00      ,PTN,    65664,89-12-14,21:05
R01      ,PTN,     1920,89-12-14,20:13
R01      ,OTH,      144,90-01-18,18:54
R03      ,PTN,      130,89-12-15,20:18
R03      ,OTH,      144,90-01-18,18:55
R04      ,PTN,     6016,90-01-23,10:59
R05      ,PTN,    65664,90-01-11,13:06
R05      ,OTH,      144,89-12-28,14:25
R06      ,PTN,    65664,90-01-11,13:08
R07      ,PTN,     2176,90-01-11,13:09
R08      ,PTN,     2176,90-01-11,13:10
R10      ,PTN,    65664,90-01-29,21:08
R50      ,PTN,      144,90-01-16,11:10
R50      ,OTH,      144,90-01-18,18:55
R80      ,PTN,      132,90-01-29,21:08
FIL 1
```

(10) Data save, resave, and recall

This program reads the directory information for files stored on floppy disk, saves or resaves the statuses set at the MP1702A/MP1756A/MP1609A/MP1651A corresponding to the memory switching functions 9PTN/OTHERS), and recalls files stored on floppy disk.

The access status of the floppy disk to obtain the directory information is monitored by serial polling and by data request command which checks the extended event status byte. During save, resave, and recall, the floppy disk access is monitored by request command MAC?.




To ①
From preceding page
• Reads file information with (FIL?) request command and stores it in array variable Read$.
• Sets file No./directory mode switching to file No. mode.
• Selects memory switching function (PTN/OTHERS).
• Select to save, resave, or recall data corresponding to memory switching.
(Save)
• Input file name to be saved.
• Compares input value with file names stored in array variable.
If file name is not found, file is saved.
• If file name is found, new file name must be input.
(Resave)
• Input file name to be resaved.
• Compares input value with file names stored in array variable.
If file name is found, file is resaved.
• If file name is not found, new file name must be re-input.
(Recall)
• Input file name to be recalled.
• Compares input value with file names stored in array variable.
If file name is found, file is recalled.
• If file name is not found, new file name must be re-input.
Saves
Resaves
Recalls
Checks floppy disk access status via (MAC?) request command.
NO (accessing)
Access end?
YES (access ended)
Reads event status via (EER?) request command thereby resetting extended event status register FILE ACCESS bit.
YES
Repeat processing?
NO
END

Program listing

```
10    !*****************************************************************
20    !*                                                               *
30    !*       MP1702A/MP1609A/MP1651A  MEMORY SAMPLE SOFT             *
40    !*                                                               *
50    !*                                                    ED_MEMO    *
60    !*****************************************************************
70    !
80    !
90    !///////////////////////////////////////////////////////////////////
100   !/                                                                 /
110   !/                          MAIN ROUTINE                           /
120   !/                                                                 /
130   !///////////////////////////////////////////////////////////////////
140   !
150   DIM Read$(199)[12]
160   DIM Fil$[36]
170   LET Add=701               ! Set Device Address
180   CLEAR Add                 ! Device Clear
190   OUTPUT Add;"SRQ 2"        ! SRQ : EXTENSION bit
200   OUTPUT Add;"EES 32"       ! Floppy Access End
210   !
220   PRINT "** MP1702A/MP1609A/MP1651A  MEMORY SAMPLE SOFT **"
230   PRINT
240    !
250   LOOP
260      !
270     GOSUB S_poll
280     GOSUB D_set
290     GOSUB Access
300      !
310     INPUT " NEXT DATA SET  [ YES=0 , NO=1 ]  ?",Loop$
320      !
330   EXIT IF Loop$="1"
340    !
350   END LOOP
360    !
370    !
380   STOP
390   !
400   !///////////////////////////////////////////////////////////////////
410   !/                                                                 /
420   !/                           SUB ROUTINE                           /
430   !/                                                                 /
440   !///////////////////////////////////////////////////////////////////
450   !
460   !-----------------------------------------------------------------!
470   !   Serial Polling                                                !
480   !                  RQS,Extension bit=1 , File Access bit=1        !
490   !-----------------------------------------------------------------!
500   !
510 S_poll:!
520   ! ------------------------------------------ Serial polling
530    !
540   OUTPUT Add;"FIL 1"
550    !
560   LOOP
570      !
580     LOOP
590        !
600       A=SPOLL(Add)                          ! Serial polling
610     EXIT IF BIT(A,6)=1 AND BIT(A,1)=1       ! RQS,Extension bit=1
```

```
620       WAIT .1
630        !
640     END LOOP
650      !
660     OUTPUT Add;"EER?"              ! REQUEST Extension register ?
670     ENTER Add;Eer$                 ! READ Extension register
680      !
690    EXIT IF Eer$[9,9]="1"           ! File access is end
700     !
710    END LOOP
720     !
730    RETURN
740     !
750     !
760     !-------------------------------------------------------------!
770     !     Read File directory , Set Memory mode                   !
780     !                  & Select  SAVE  or  RESAVE  or  RECALL     !
790     !-------------------------------------------------------------!
800     !
810 D_set;!
820     ! ---------------------------------------- Read File directory
830    I=0
840     !
850    OUTPUT Add;"FIL?"
860     !
870    LOOP
880     !
890     ENTER Add;Fil$
900     Read$(I)=Fil$[1,12]
910    EXIT IF Fil$="FIL 1"
920     !
930     I=I+1
940     !
950    END LOOP
960     !
970     ! ---------------------------------------- Set Memory mode
980    OUTPUT Add;"FIL 0"
990     !
1000   LOOP
1010     INPUT "MEMORY MODE SELECT [ PTN=0 , OTHERS=1 ] ?",Mem
1020   EXIT IF Mem=0 OR Mem=1
1030   END LOOP
1040    !
1050   OUTPUT Add;"MEM "&VAL$(Mem)
1060    !
1070   IF Mem=0 THEN
1080     Mem$="PTN"
1090   ELSE
1100     Mem$="OTH"
1110   END IF
1120    !
1130    ! ------------------------------  Select SAVE,RESAVE,RECALL
1140   LOOP
1150     INPUT "SELECT [ SAVE=0 , RESAVE=1 , RECALL=2 ] ?",Dta
1160   EXIT IF Dta=0 OR Dta=1 OR Dta=2
1170   END LOOP
1180    !
1190   IF Dta=0 THEN GOSUB Dsave
1200   IF Dta=1 THEN GOSUB Dresave
1210   IF Dta=2 THEN GOSUB Drecall
1220    !
1230   RETURN
1240    !
1250    !
```

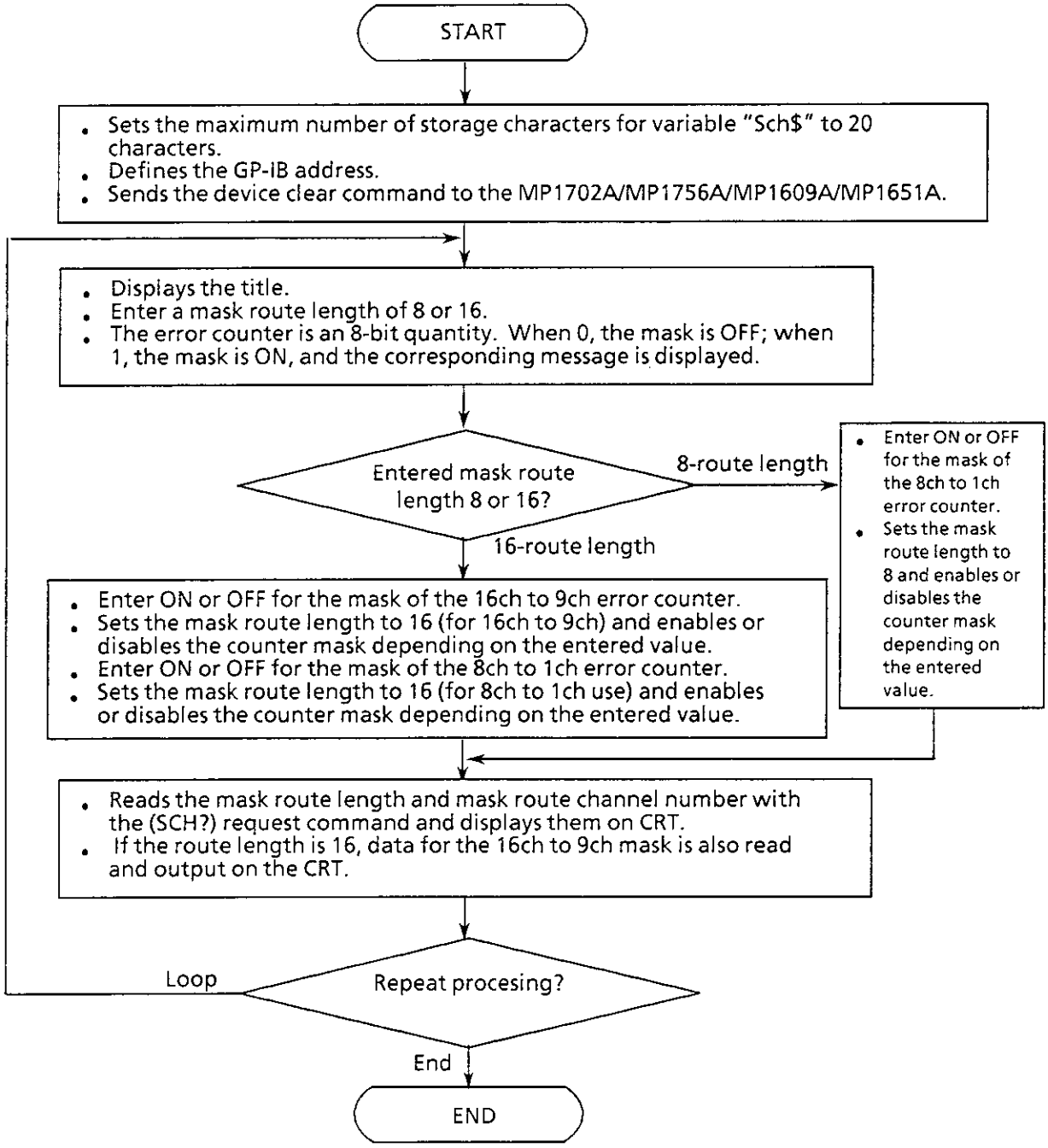
```
1260   !-------------------------------------------------------------!
1270   !      * DATA SAVE *                                          !
1280   !            SAME Memory mode & File name --> ERROR !         !
1290   !-------------------------------------------------------------!
1300   !
1310 Dsave:!
1320   !
1330  LOOP
1340     !
1350    I=0
1360     !
1370    INPUT "** DATA SAVE **  FILE NAME [ 0~99 ] ?",Nam
1380     !
1390    Result$="     "
1400    LOOP
1410       !
1420      IF Mem$=Read$(I)[10,12] AND Nam=VAL(Read$(I)[2,3]) THEN
1430        Result$="SAME"
1440      END IF
1450       !
1460      I=I+1
1470       !
1480    EXIT IF Read$(I)[1,5]="FIL 1"
1490     !
1500    END LOOP
1510     !
1520  EXIT IF Result$<>"SAME"
1530   !
1540  END LOOP
1550   !
1560  OUTPUT Add;"SAV "&VAL$(Nam)
1570   !
1580  RETURN
1590   !
1600   !
1610   !-------------------------------------------------------------!
1620   !      * DATA RESAVE *                                        !
1630   !              SAME Memory mode & File name --> OK !          !
1640   !-------------------------------------------------------------!
1650   !
1660 Dresave:!
1670       !
1680  LOOP
1690     !
1700    I=0
1710    INPUT "** DATA RESAVE **  FILE NAME [ 0~99 ] ?",Nam
1720     !
1730    Result$="     "
1740    LOOP
1750       !
1760      IF Mem$=Read$(I)[10,12] AND Nam=VAL(Read$(I)[2,3]) THEN
1770        Result$="SAME"
1780      END IF
1790       !
1800      I=I+1
1810       !
1820    EXIT IF Read$(I)[1,5]="FIL 1"
1830     !
1840    END LOOP
1850     !
1860  EXIT IF Result$="SAME"
1870  END LOOP
1880   !
1890  OUTPUT Add;"RSV "&VAL$(Nam)
1900   !
1910  RETURN
```

```
1920    !
1930    !
1940    !-------------------------------------------------------------!
1950    !      * DATA RECALL *                                        !
1960    !               SAME Memory mode $ File name --> OK !         !
1970    !-------------------------------------------------------------!
1980    !
1990  Drecall:!
2000    !
2010   LOOP
2020      !
2030     I=0
2040      !
2050     INPUT "** DATA RECALL **  FILE NAME [ 0~99 ] ?",Nam
2060      !
2070     Result$="     "
2080     LOOP
2090        !
2100       IF Mem$=Read$(I)[10,12] AND Nam=VAL(Read$(I)[2,3]) THEN
2110         Result$="SAME"
2120       END IF
2130        !
2140       I=I+1
2150        !
2160     EXIT IF Read$(I)[1,5]="FIL 1"
2170      !
2180     END LOOP
2190      !
2200   EXIT IF Result$="SAME"
2210   END LOOP
2220    !
2230   OUTPUT Add;"RCL "&VAL$(Nam)
2240    !
2250   RETURN
2260    !
2270    !
2280    !-------------------------------------------------------------!
2290    !       MEMORY  ACCESS  CONDITION ?                           !
2300    !                          &  RESET FILE ACCESS bit           !
2310    !-------------------------------------------------------------!
2320    !
2330  Access:!
2340    !
2350   LOOP
2360     OUTPUT Add;"MAC?"
2370     ENTER Add;Mac$
2380   EXIT IF Mac$="MAC 0"
2390   END LOOP
2400    !
2410   OUTPUT Add;"EER?"
2420   ENTER Add;Eer$
2430    !
2440   RETURN
2450    !
2460    !
2470   END
```

(11) Setting the mask route length and channel number

This program sets the mask route length and channel number.

First, the program selects the mask route length and turns the mask for each error counter ON or OFF depending on the selected mask route length. Then, it reads the mask route length value with (SCH?) request command and displays it on CRT.



Program listing

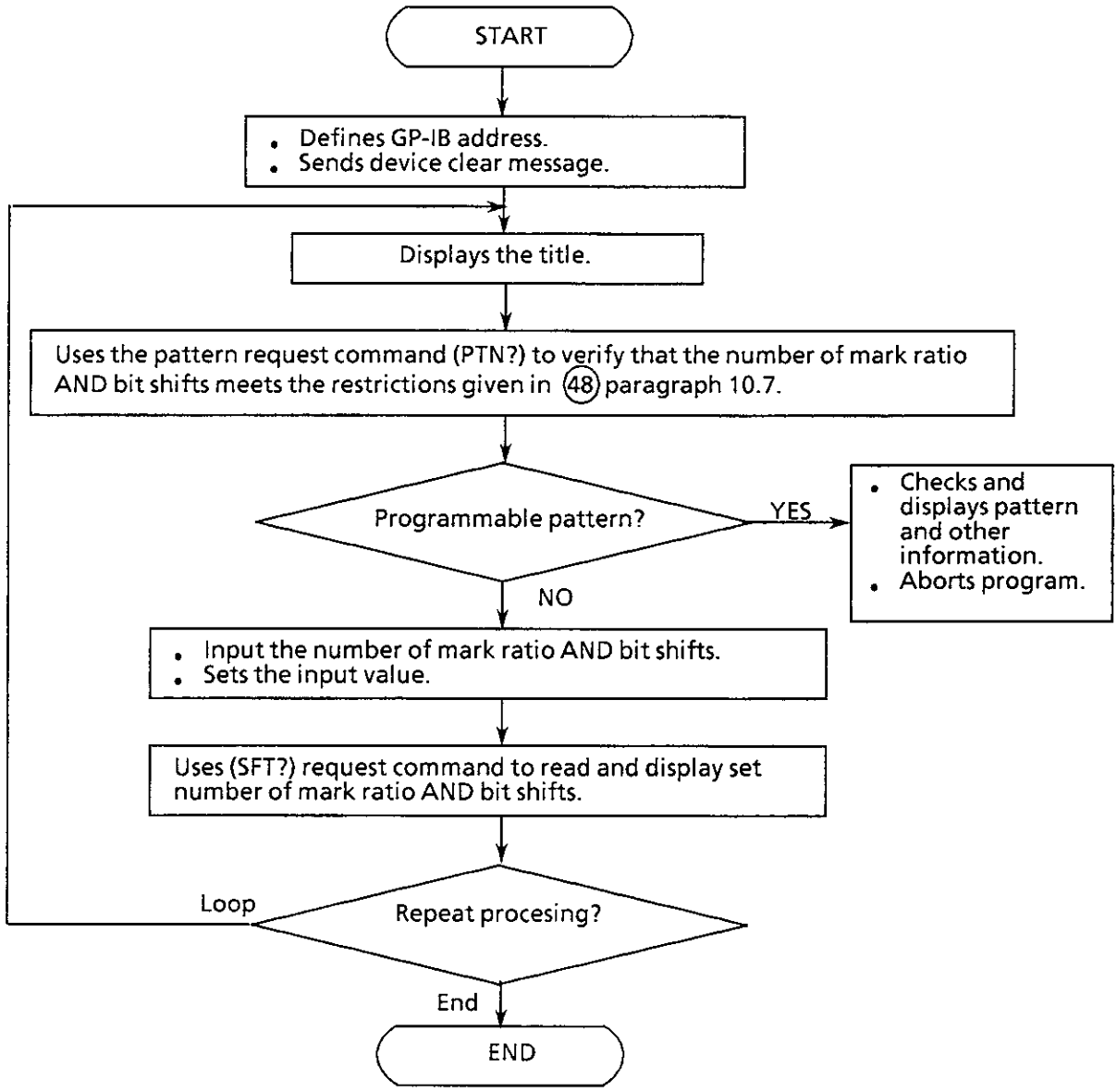
```
10     !*****************************************************************
20     !*                                                               *
30     !*   MP1702A/MP1609A/MP1651A  MEASURE CH MASK  SAMPLE SOFT       *
40     !*                                                               *
50     !*                                                      ED_MASK  *
60     !*****************************************************************
70     !
80     !
90     !---------------------------------------------------------------!
100    !                          MAIN ROUTINE                         !
110    !---------------------------------------------------------------!
120    !
130    DIM Sch$[20]
140    Add=701                        ! Device address
150    CLEAR Add                      ! Device Clear
160     !
170    LOOP
180       !
190      CLEAR SCREEN
200      PRINT "* MP1702A/MP1609A/MP1651A MEASURE CH MASK SAMPLE SOFT *"
210      PRINT
220       !
230      GOSUB M_route
240      GOSUB Dsply
250       !
260      INPUT " NEXT DATA SET  [ YES=0 , NO=1 ]  ?",Loop$
270       !
280    EXIT IF Loop$="1"
290     !
300    END LOOP
310     !
320     !
330    STOP
340     !
350     !---------------------------------------------------------------!
360     !                          SUB ROUTINE                          !
370     !---------------------------------------------------------------!
380     !
390  M_route: !-------------------- Set Mask route length & Channel
400     !
410    LOOP
420       !
430      INPUT "MASK ROUTE LENGTH [ 8 LENGTH=0 , 16 LENGTH=1 ] ?",Msk
440    EXIT IF Msk=0 OR Msk=1
450    END LOOP
460     !
470    PRINT "ERROR COUNTER = 8 bit [ 0=MASK OFF, 1=MASK ON ]"
480    PRINT
490     !
500    IF Msk=1 THEN
510      PRINT "   ** MASK ROUTE LENGTH =16 **"
520       !
530      INPUT "CHANNEL SET DATA BIT8 (16ch)-->BIT1 (9ch) [0/1]",Sch2$
540      OUTPUT Add;"MSK 2;SCH #B"&Sch2$
550       !
560      INPUT "CHANNEL SET DATA BIT8 (8ch)-->BIT1 (1ch) [0/1]",Sch1$
570      OUTPUT Add;"MSK 1;SCH #B"&Sch1$
580       !
590    ELSE
600      PRINT "   ** MASK ROUTE LENGTH =8 **"
610      INPUT "CHANNEL SET DATA BIT8 (8ch)-->BIT1 (1ch) [0/1]",Sch1$
```

```
620       !
630       OUTPUT Add;"MSK 0;SCH #B"&Sch1$
640       !
650     END IF
660     !
670     RETURN
680     !
690     !
700 Dsply: !------------------------------------ Display set data
710     !
720     OUTPUT Add;"SCH?"
730     !
740     ENTER Add;Sch$
750     PRINT Sch$,
760     PRINT "     (8ch~1ch)"
770     !
780     IF Sch$[5,5]="1" THEN
790       ENTER Add;Sch$
800       PRINT Sch$,
810       PRINT "     (16ch~9ch)"
820     END IF
830     !
840     PRINT
850     PRINT
860     !
870     RETURN
880     !
890     END
```

(12) Setting the number of mark ratio AND bit shifts

This program sets the number of mark ratio AND bit shifts.

The number of mark ratio AND bit shifts cannot be set for programmable patterns. If attempted, information about the pattern is displayed on the CRT and program execution is aborted.



Program listing

```
10    !*****************************************************************
20    !*                                                               *
30    !*               MP1702A / MP1609A / MP1651A                     *
40    !*          MARK RATIO AND BIT SHIFT  SAMPLE SOFT                *
50    !*                                                               *
60    !*                                                    ED_MARK    *
70    !*****************************************************************
80    !
90    !---------------------------------------------------------------!
100   !                          MAIN ROUTINE                         !
110   !---------------------------------------------------------------!
120   !
130   LET Add=701                    ! Set Device Address
140   CLEAR Add                      ! Device Clear
150    !
160   LOOP
170    !
180   CLEAR SCREEN
190   PRINT "** MP1702A/MP1609A/MP1651A  MARK RATIO AND BIT SHIFT **"
200   PRINT
210      !
220     GOSUB Ptn                   ! Check pattern mode
230     GOSUB Mark                  ! Mark ratio bit shift
240      !
250     INPUT " NEXT DATA SET  [ YES=0 , NO=1 ]  ?",Loop$
260      !
270   EXIT IF Loop$="1"
280    !
290   END LOOP
300    !
310    !
320   STOP
330   !
340   !---------------------------------------------------------------!
350   !                          SUB ROUTINE                          !
360   !---------------------------------------------------------------!
370      !
380 Ptn: !------------------------------------ Check pattern mode
390      !
400   OUTPUT Add;"PTN?"              ! REQUEST Pattern mode ?
410   ENTER Add;Ptn$                 ! READ Pattern mode
420    !
430   IF Ptn$[5,5]="0" OR Ptn$[5,5]="1" THEN    ! PROG.MODE-->ERROR
440     GOSUB Err
450     PRINT "** PROGRAM STOP !"
460     STOP
470      !
480   END IF
490    !
500   RETURN
510    !
520 Mark:!------------------------------ Set Mark ratio bit shift
530   !
540   LOOP
550      !
560     INPUT "MARK RATIO  [ 1 BIT SHIFT=0, 3 BIT SHIFT=1 ] ?",Sft
570   EXIT IF Sft=0 OR Sft=1
580    !
590   END LOOP
```

```
600     !
610    OUTPUT Add;"SFT "&VAL$(Sft)
620     !
630    OUTPUT Add;"SFT?"
640    ENTER Add;Sft$
650     !
660    IF Sft$[5,5]="0" THEN
670      Sft$="1"
680    ELSE
690      Sft$="3"
700    END IF
710     !
720    PRINT "MARK RATIO AND BIT SHIFT = "&Sft$
730     !
740    RETURN
750     !
760     !
770 Err: !---------------------------------- Display Pattern data
780     !
790    OUTPUT Add;"FRM?"
800    ENTER Add;Frm$
810     !
820    IF Ptn$[5,5]="0" THEN
830       !
840      PRINT "** ERROR !!   PATTERN MODE = PROG.WORD "
850      PRINT
860       !
870      IF Frm$="FRM 0" THEN
880         !
890        OUTPUT Add;"WNB?"
900        ENTER Add;Wnb$
910        PRINT "NUMBER OF WORD ="&Wnb$[4,9]
920         !
930        OUTPUT Add;"WLN?"
940        ENTER Add;Wln$
950        PRINT "WORD LENGTH ="&Wln$[4,6]
960         !
970      ELSE
980        GOSUB Fln
990         !
1000     END IF
1010      !
1020   ELSE
1030      !
1040     PRINT "** ERROR !!   PATTERN MODE = PROG.DATA "
1050     PRINT
1060      !
1070     OUTPUT Add;"FRM?"
1080     ENTER Add;Frm$
1090      !
1100     IF Frm$="FRM 0" THEN
1110        !
1120       OUTPUT Add;"DLN?"
1130       ENTER Add;Dln$
1140       PRINT "DATA LENGTH ="&Dln$[4,10]
1150        !
1160     ELSE
1170        !
1180       GOSUB Fln
1190     END IF
1200      !
1210   END IF
1220      !
```

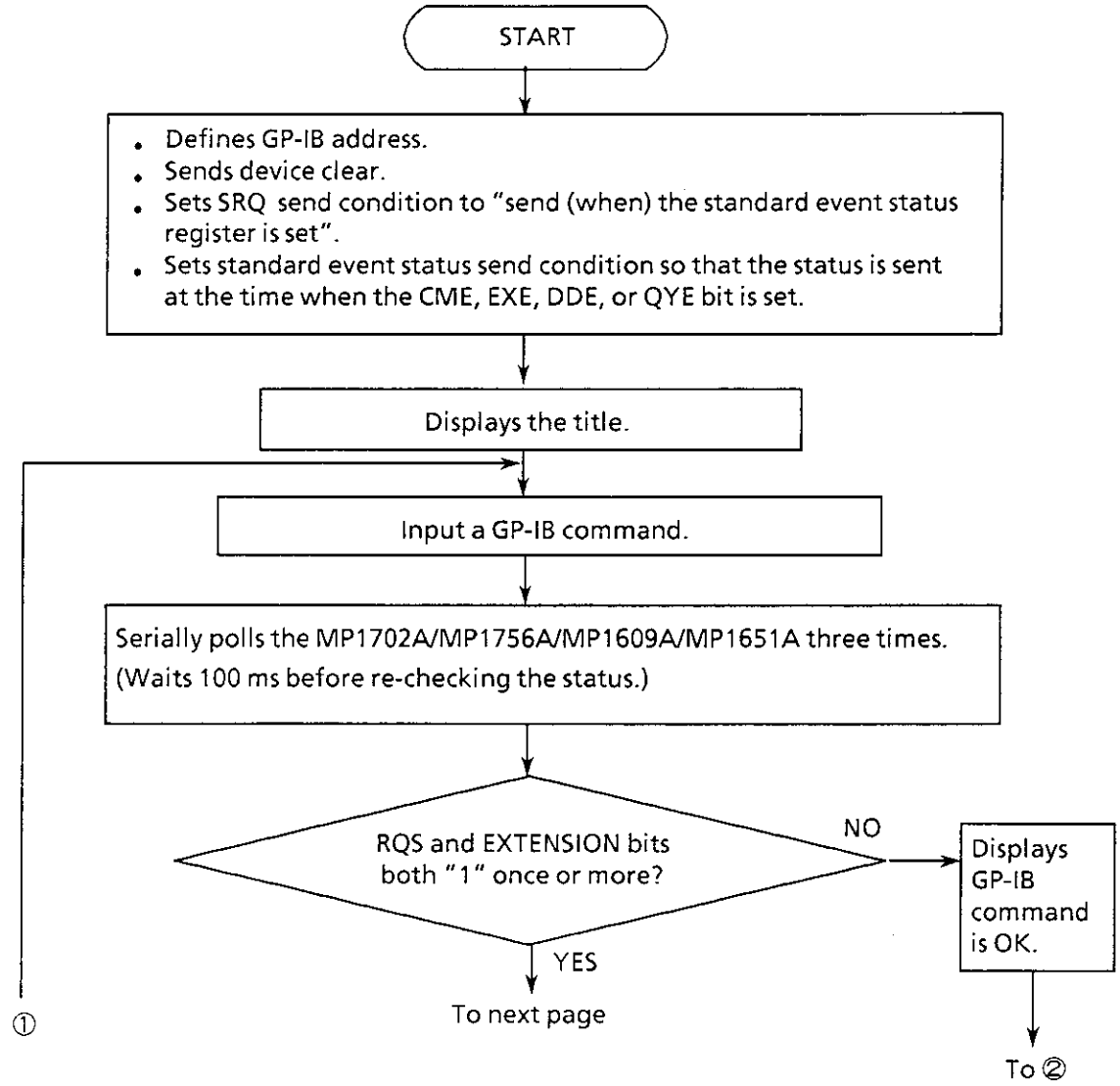
```
1230  OUTPUT Add;"PAG?"
1240  ENTER Add;Pag$
1250  PRINT "PAGE ="&Pag$[4,9]
1260  PRINT
1270   !
1280  RETURN
1290       !
1300 Fln: !------------------------------------ Frame length mode
1310       !
1320  OUTPUT Add;"FLN?"
1330  ENTER Add;Fln$
1340  PRINT "FRAME LENGTH ="&Fln$[4,6]
1350   !
1360  RETURN
1370   !
1380  END
```

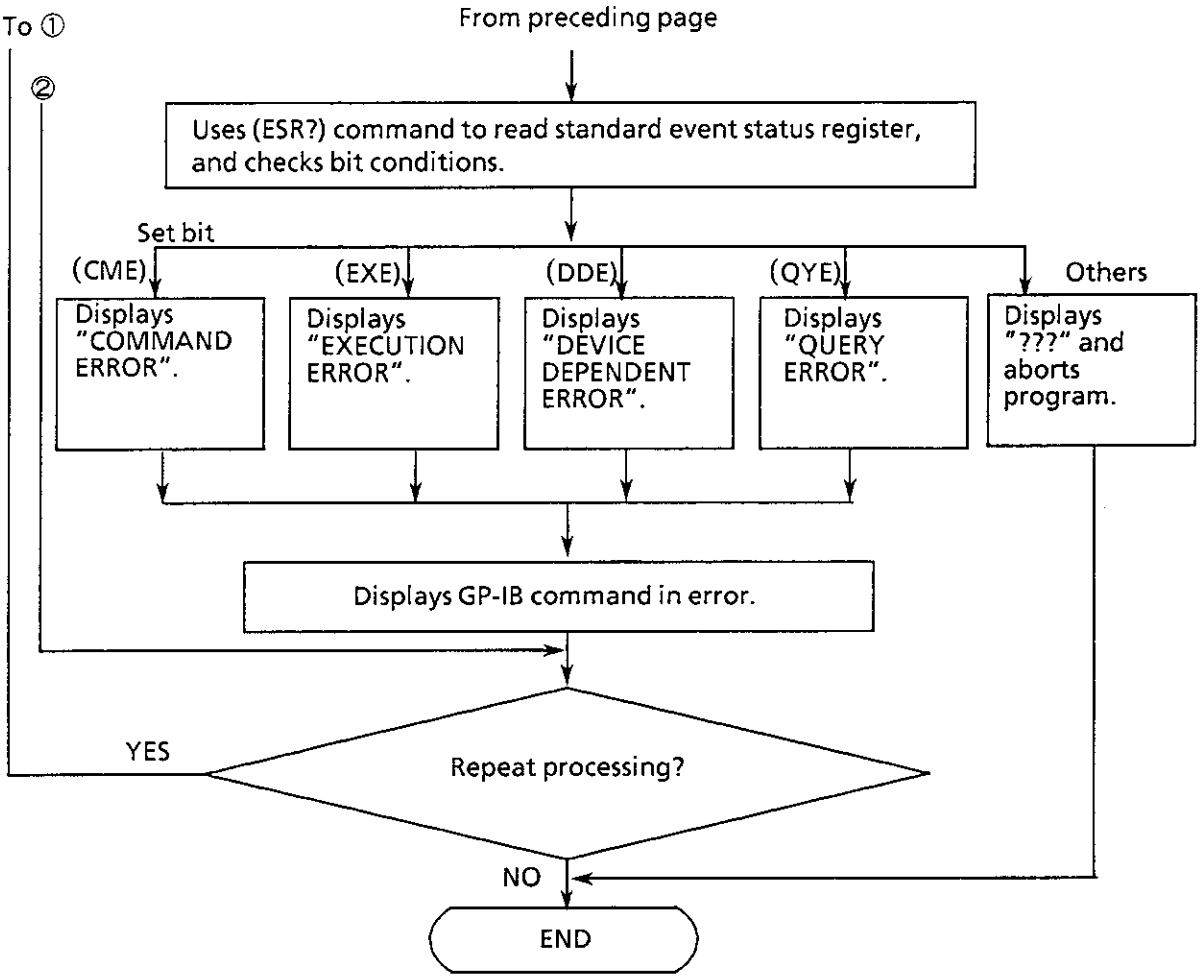
(13) Standard event status byte check (COMMAND ERROR bit check by serial polling)

This program checks the CME, EXE, DDE, and QYE bits of the standard event status byte, and indicates whether the input GP-IB command is correct or not.

When the command is incorrect, the cause of error is displayed.

The GP-IB status byte is monitored by serial polling and the status of the standard event status byte which is obtained by data request command.




To ①
From preceding page
②
Uses (ESR?) command to read standard event status register, and checks bit conditions.
Set bit
(CME)
(EXE)
(DDE)
(QYE)
Others
Displays "COMMAND ERROR".
Displays "EXECUTION ERROR".
Displays "DEVICE DEPENDENT ERROR".
Displays "QUERY ERROR".
Displays "???" and aborts program.
Displays GP-IB command in error.
YES
Repeat processing?
NO
END

Program listing

```
10    !*************************************************************
20    !*                                                           *
30    !*                  MP1702A / MP1609A / MP1651A              *
40    !*    STANDARD EVENT STATUS REGISTER CHECK SAMPLE SOFT        *
50    !*                                                  ED_ESE   *
60    !*************************************************************
70    !
80    !-----------------------------------------------------------!
90    !                              MAIN ROUTINE                 !
100   !-----------------------------------------------------------!
110   !
120   LET Add=701               ! Set Device Address
130   CLEAR Add                 ! Device Clear
140   OUTPUT Add;"SRQ 32"       ! SRQ : COMMAND ERROR bit
150   OUTPUT Add;"ESE 60"       ! CME,EXE,DDE,QYE bit
160    !
170   PRINT "      ** MP1702A/MP1609A/MP1651A **"
180   PRINT "**  STANDARD EVENT STATUS REGISTER CHECK **"
190   PRINT
200    !
210   LOOP
220      !
230     INPUT " INPUT ANY GP-IB COMMAND ?",Com$
240     OUTPUT Add;Com$
250      !
260     GOSUB S_poll
270      !
280     INPUT " NEXT COMMAND SET ?  [ YES=0 , NO=1 ] ?",Loop$
290      !
300   EXIT IF Loop$="1"
310    !
320   END LOOP
330    !
340   STOP
350   !
360   !-----------------------------------------------------------!
370   !                              SUB ROUTINE                  !
380   !-----------------------------------------------------------!
390   !
400 S_poll:!-------------------------------------  Check Status byte
410    !
420   Byt=0
430    !
440   FOR I=0 TO 2
450      !
460     A=SPOLL(Add)                                ! Serial polling
470      !
480     IF BIT(A,6)=1 AND BIT(A,5)=1 THEN Byt=A ! RQS,COMMAND ERROR bit=1
490      !
500     WAIT .1
510   NEXT I
520    !
530   IF BIT(Byt,6)=1 AND BIT(Byt,5)=1 THEN
540     GOSUB Err
550   ELSE
560     PRINT " GP-IB COMMAND IS OK "
570     PRINT
580   END IF
590    !
600   RETURN
610    !
```

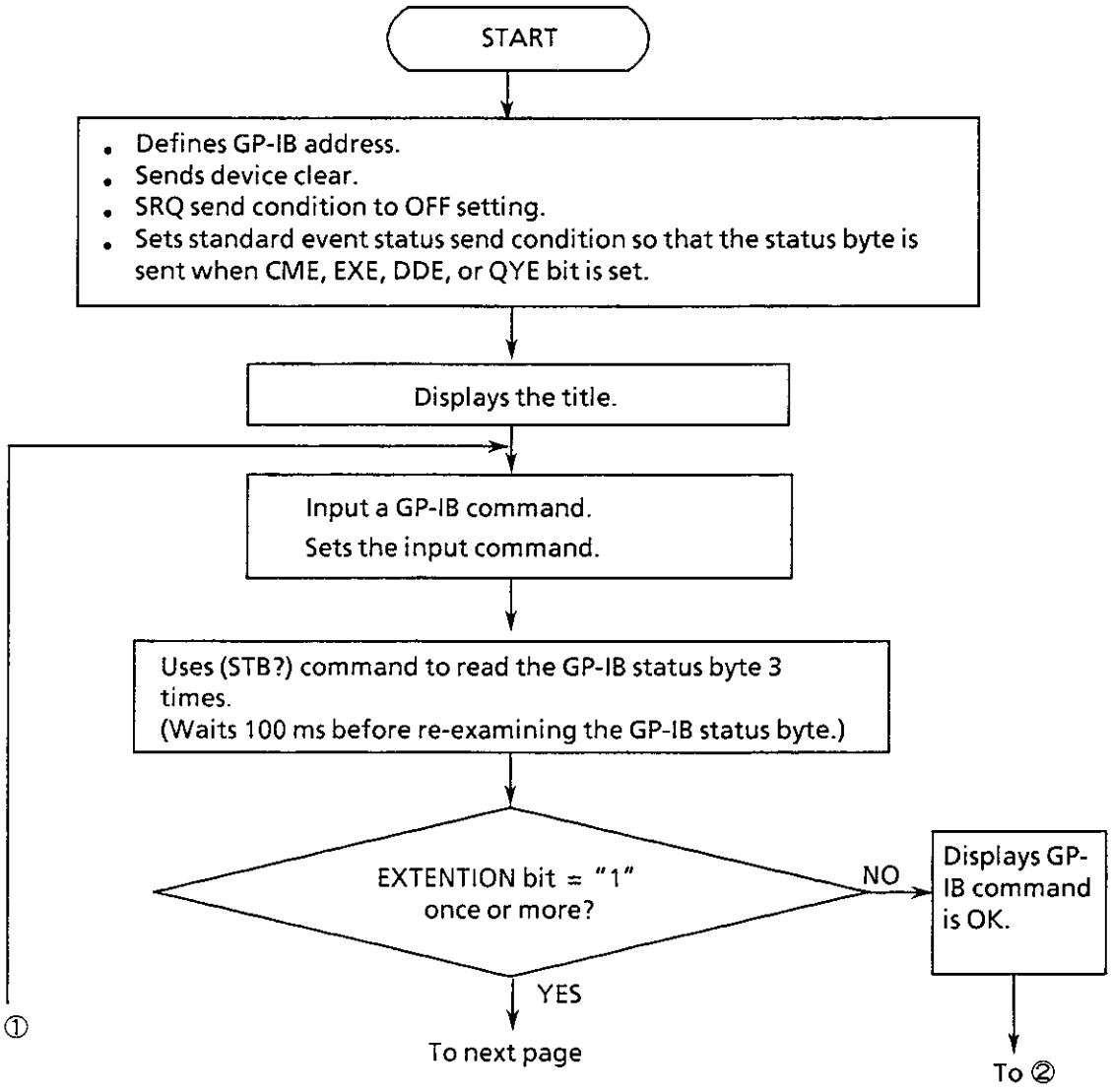
```
620     !
630 Err:!---------------------------------------- ERROR
640     !
650    OUTPUT Add;"ESR?"
660    ENTER Add;Er$
670     !
680    IF Er$[9,9]="1" THEN PRINT " COMMAND ERROR !!   "
690    IF Er$[10,10]="1" THEN PRINT " EXECUTION ERROR !! "
700    IF Er$[11,11]="1" THEN PRINT " DEVICE DEPENDENT ERROR !! "
710    IF Er$[12,12]="1" THEN PRINT " QUERY ERROR !! "
720     !
730    PRINT " INPUT COMMAND = "&Com$
740    PRINT
750     !
760   IF Er$[7,7]="1" OR Er$[8,8]="1" OR Er$[13,13]="1" OR Er$[14,14]="1" THEN
770      GOSUB Trap
780    END IF
790     !
800    RETURN
810     !
820     !
830 Trap:  !------------
840     !
850    PRINT "???"
860    STOP
870     !
880     !
890    END
```

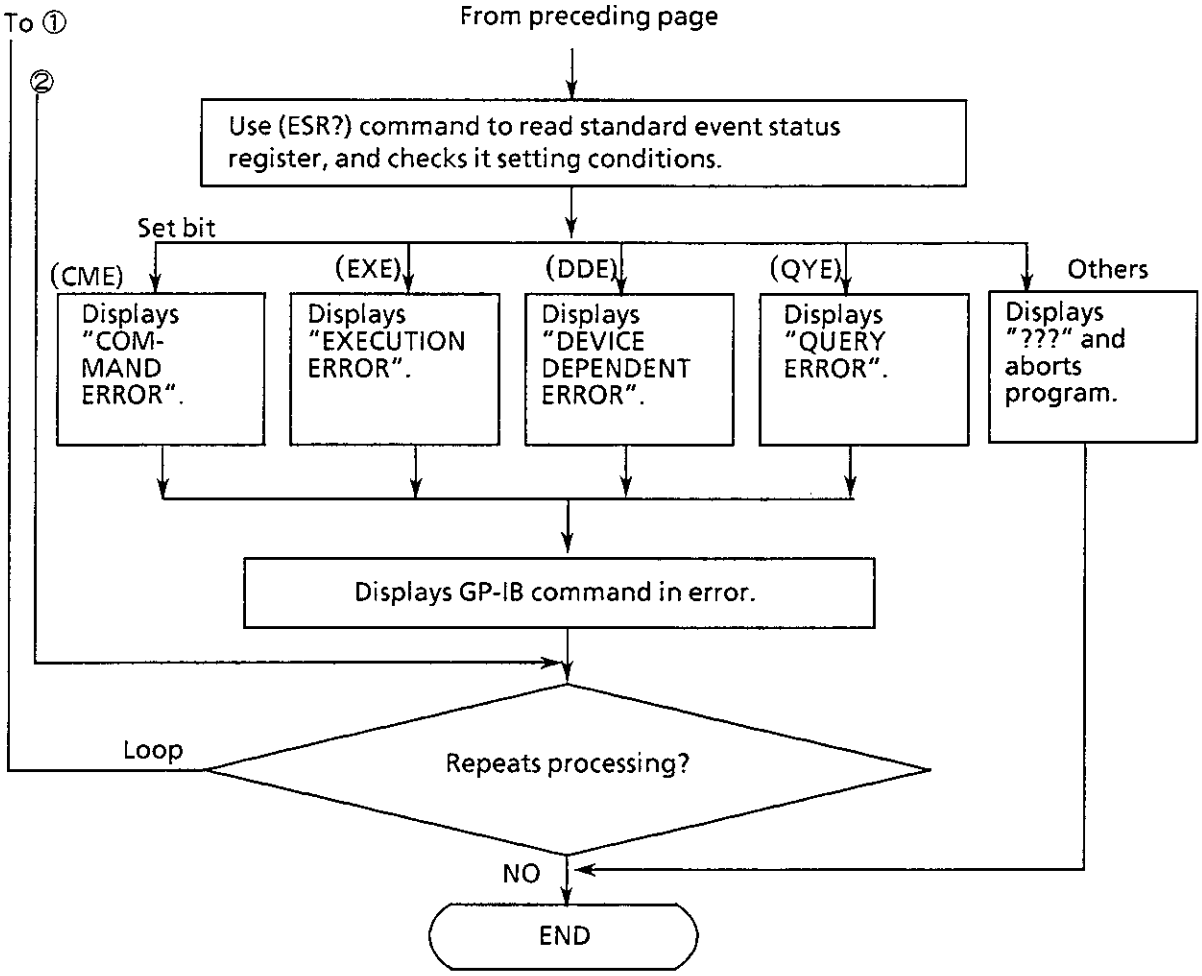
(14) Standard event status byte check (COMMAND ERROR bit check by (STB?) request command)

This program checks the CME, EXE, DDE, and QYE bits of the standard event status byte, and indicates whether the input GP-IB command is correct or not.

When the command is incorrect, the cause of error is displayed.

The GP-IB status byte and the standard event status byte are both checked by data request commands.




To ①
From preceding page
②
Use (ESR?) command to read standard event status register, and checks it setting conditions.
Set bit
(CME)
(EXE)
(DDE)
(QYE)
Others
Displays "COM-MAND ERROR".
Displays "EXECUTION ERROR".
Displays "DEVICE DEPENDENT ERROR".
Displays "QUERY ERROR".
Displays "???" and aborts program.
Displays GP-IB command in error.
Loop
Repeats processing?
NO
END

Program listing

```
10    !*************************************************************
20    !*                                                           *
30    !*                MP1702A / MP1609A / MP1651A                *
40    !*      STANDARD EVENT STATUS REGISTER CHECK SAMPLE SOFT      *
50    !*                                                  ED_ESE2  *
60    !*************************************************************
70    !
80    !-----------------------------------------------------------!
90    !                            MAIN ROUTINE                   !
100   !-----------------------------------------------------------!
110   !
120   Add=701                   ! Set Device address
130   CLEAR Add                 ! Device Clear
140   OUTPUT Add;"SRQ 0"        ! SRQ : OFF
150   OUTPUT Add;"ESE 60"       ! CME,EXE,DDE,QYE bit
160    !
170   PRINT "       ** MP1702A/MP1609A/MP1651A **"
180   PRINT "** STANDARD EVENT STATUS REGISTER CHECK **"
190   PRINT
200    !
210   LOOP
220      !
230     INPUT " INPUT ANY GP-IB COMMAND ?",Com$
240     OUTPUT Add;Com$
250      !
260     GOSUB Stb_reg
270      !
280     INPUT " NEXT COMMAND SET ?  [ YES=0 , NO=1 ] ?",Loop$
290      !
300   EXIT IF Loop$="1"
310    !
320   END LOOP
330    !
340   STOP
350   !
360   !-----------------------------------------------------------!
370   !                            SUB ROUTINE                    !
380   !-----------------------------------------------------------!
390   !
400 Stb_reg: ! ---------------------------------  Check Status byte
410    !
420   Byt$="                "
430    !
440   FOR I=0 TO 2
450      !
460     OUTPUT Add;"STB?"       ! REQUEST Status byte register ?
470     ENTER Add;Stb$          ! READ Status byte register
480      !
490     IF Stb$[9,9]="1" THEN    ! COMMAND ERROR bit=1
500       Byt$=Stb$
510     END IF
520      !
530     WAIT .1
540   NEXT I
550    !
560   IF Byt$[9,9]="1" THEN
570     GOSUB Err
580   ELSE
590     PRINT " GP-IB COMMAND IS OK"
600     PRINT
610   END IF
```

```
620     !
630   RETURN
640     !
650     !
660 Err:!------------------------------------ ERROR
670     !
680   OUTPUT Add;"ESR?"
690   ENTER Add;Er$
700     !
710   IF Er$[9,9]="1" THEN PRINT " COMMAND ERROR !! "
720   IF Er$[10,10]="1" THEN PRINT " EXECUTION ERROR !! "
730   IF Er$[11,11]="1" THEN PRINT " DEVICE DEPENDENT ERROR !! "
740   IF Er$[12,12]="1" THEN PRINT " QUERY ERROR !! "
750     !
760   PRINT " INPUT COMMAND = "&Com$
770   PRINT
780     !
790  IF Er$[7,7]="1" OR Er$[8,8]="1" OR Er$[13,13]="1" OR Er$[14,14]="1" THEN
800     GOSUB Trap
810   END IF
820     !
830   RETURN
840     !
850     !
860 Trap: !----------
870     !
880   PRINT "???"
890   STOP
900     !
910     !
920   END
```

(15) DMA transfer of pattern data

This program transfers pattern data to and from an HP9000 series computer, which is used as a controller, by DMA.

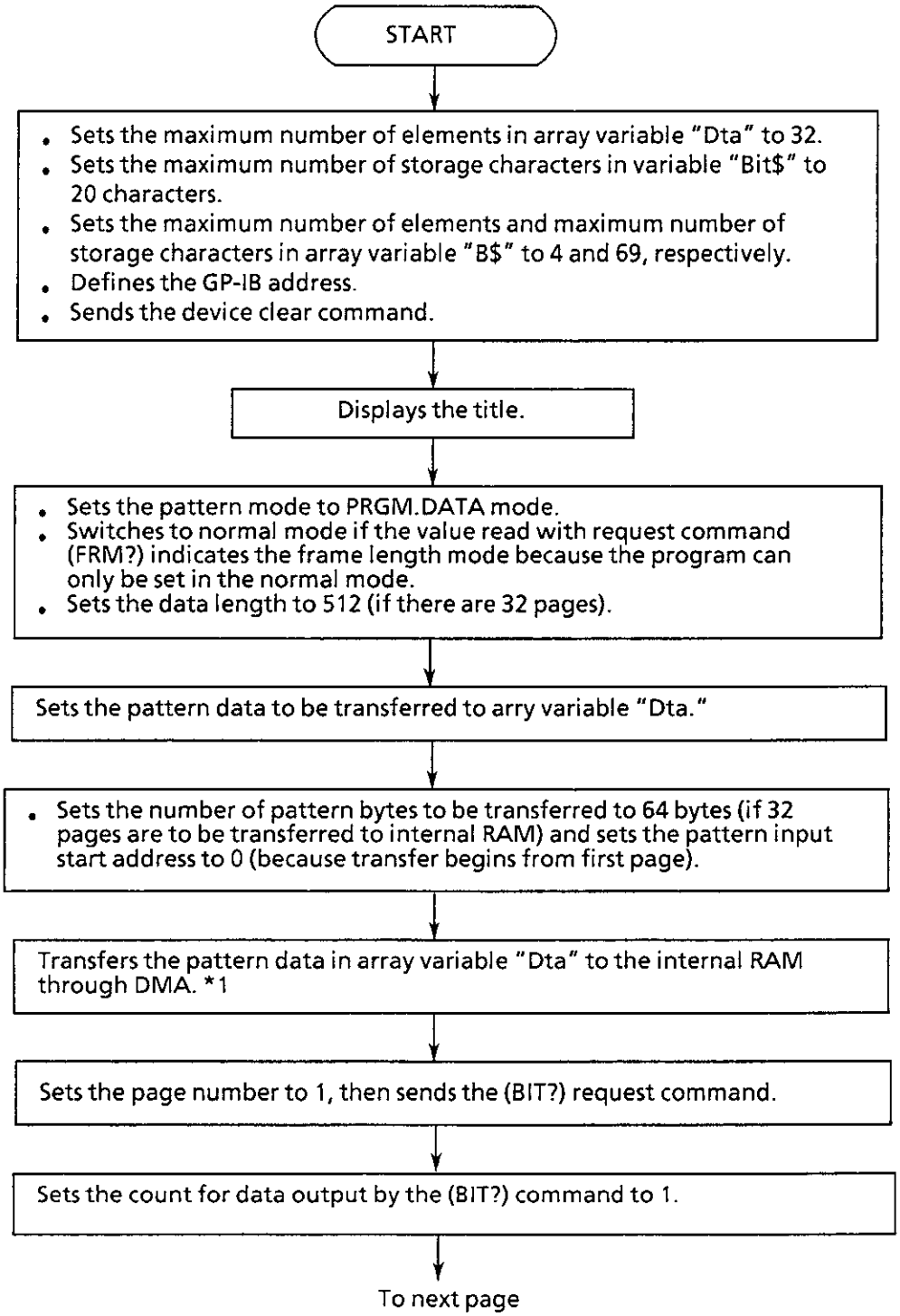
The pattern data can be transferred to the MP1702A/MP1756A/MP1609A/MP1651A via DMA by either the (BIT?) request command as shown in program (a) or by DMA transfer as shown in program (b). The following pages also list the results of program execution.

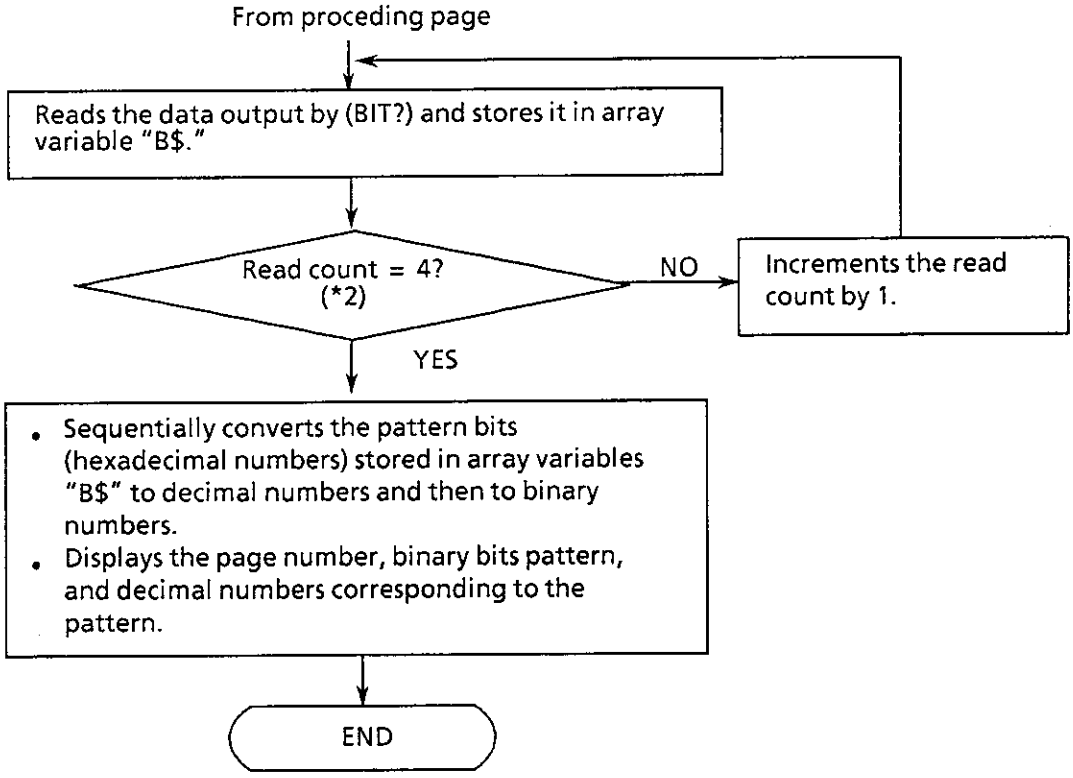
Table 15-5 shows the relationships between array variables "Dta" and decimal, binary, and hexadecimal values for each variable. These relationships vary with the type of controller.

Table 15-5 Relationships between Array Variable and Set Values

Array variable	Set value (Decimal number)	Corresponding binary value and BIT LED No.																Corresponding hexadecimal value	Corresponding page number
		16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1		
Dta(0)	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1H	1
Dta(1)	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2H	2
Dta(2)	4	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	4H	3
Dta(3)	8	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	8H	4
Dta(4)	16	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	10H	5
Dta(5)	32	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	20H	6
Dta(6)	64	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	40H	7
Dta(7)	128	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	80H	8
Dta(8)	256	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	100H	9
Dta(9)	512	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	200H	10
Dta(10)	1024	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	400H	11
Dta(11)	2048	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	800H	12
Dta(12)	4096	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1000H	13
Dta(13)	8192	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000H	14
Dta(14)	16384	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000H	15
Dta(15)	32767	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	7FFFH	16
Dta(16)	−32768	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8000H	17
Dta(17)	−16384	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C000H	18
Dta(18)	−8192	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	E000H	19
Dta(19)	−4096	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	F000H	20
Dta(20)	−2048	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	F800H	21
Dta(21)	−1024	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	FC00H	22
Dta(22)	−512	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	FE00H	23
Dta(23)	−256	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	FF00H	24
Dta(24)	−128	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	FF80H	25
Dta(25)	−64	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	FFC0H	26
Dta(26)	−32	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	FFE0H	27
Dta(27)	−16	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	FFF0H	28
Dta(28)	−8	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	FFF8H	29
Dta(29)	−4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	FFFCH	30
Dta(30)	−2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	FFFEH	31
Dta(31)	−1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFFH	32

(a) DMA transfer of pattern data (Pattern data output by (BIT?) request command)





*1 Pattern data transfer <OUPUT Add USING "W";Dta(*)>

W: Outputs the 2's complement of 16 bits of data an integer.

The upper byte of data is always transferred through the 8-BIT I/O port of the GP-IB interface board before the lower byte.

*: Outputs the entire "Dta" array.

*2 Read count of data output by (BIT?) command

In this program, the data length is set to 512, so there are pattterns on 32 pages (512 ÷ 16 = 32).

Because 8-pages of data is transferred by BIT? at one time, the program must read data four times in a single transfer session.

Program listing

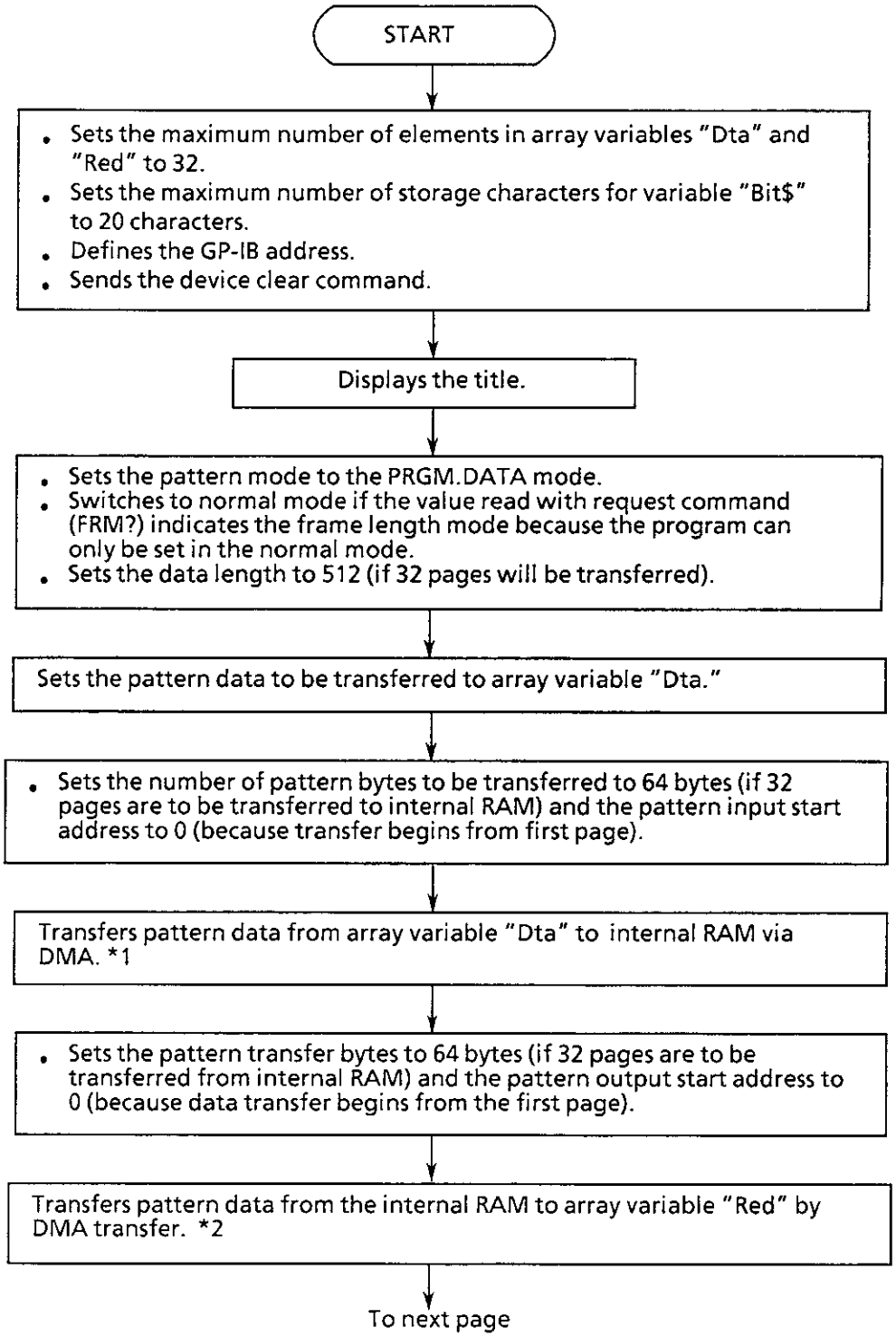
```
10    !************************************************************
20    !                                                           *
30    !                MP1702A / MP1609A / MP1651A                *
40    !      PROGRAMMABLE PATTERN DATA DMA TRANSFER  SAMPLE SOFT  *
50    !                                                 ED_DMA    *
60    !************************************************************
70    !
80    DIM Dta(31)
90    DIM Bit$[20]
100   DIM B$(4)[69]
110   LET Add=701               ! DEVICE ADDRESS
120   CLEAR Add                 ! DEVICE CLEAR
130    !
140   CLEAR SCREEN
150    !
160   PRINT "** MP1702A/MP1609A/MP1651A PATTERN DATA DMA TRANSFER **"
170   PRINT
180    !
190   OUTPUT Add;"PTN 1"        ! PATTERN MODE : PROG.DATA
200    !
210   OUTPUT Add;"FRM?"
220   ENTER Add;Frm$
230   IF Frm$="FRM 1" THEN OUTPUT Add;"FRM 0"
240    !
250   OUTPUT Add;"DLN 512"      ! DATA LENGTH : 512
260    !
270    !--------- SET PATTERN BIT DATA ---------
280    !
290   Dta(0)=1
300   Dta(1)=2
310   Dta(2)=4
320   Dta(3)=8
330   Dta(4)=16
340   Dta(5)=32
350   Dta(6)=64
360   Dta(7)=128
370   Dta(8)=256
380   Dta(9)=512
390   Dta(10)=1024
400   Dta(11)=2048
410   Dta(12)=4096
420   Dta(13)=8192
430   Dta(14)=16384
440   Dta(15)=32767
450   Dta(16)=-32768
460   Dta(17)=-16384
470   Dta(18)=-8192
480   Dta(19)=-4096
490   Dta(20)=-2048
500   Dta(21)=-1024
510   Dta(22)=-512
520   Dta(23)=-256
530   Dta(24)=-128
540   Dta(25)=-64
550   Dta(26)=-32
560   Dta(27)=-16
570   Dta(28)=-8
580   Dta(29)=-4
590   Dta(30)=-2
600   Dta(31)=-1
610    !
```

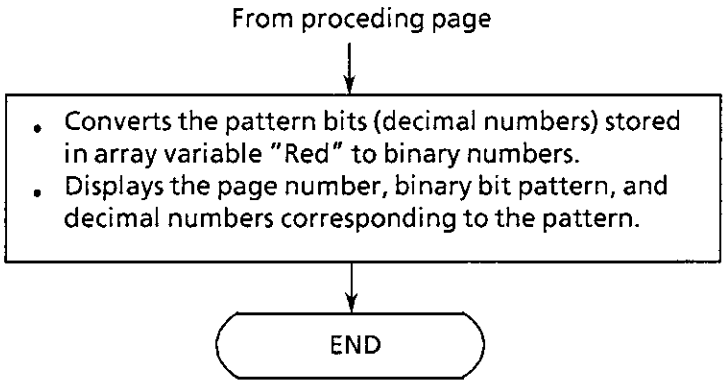
```
620   OUTPUT Add;"WRT 64,0"
630    !
640   OUTPUT Add USING "W";Dta(*)
650    !
660    !
670   OUTPUT Add;"PAG 1;BIT?"
680    !
690   FOR I=1 TO 4
700     ENTER Add;B$(I)
710   NEXT I
720    !
730   Y=1
740    !
750   FOR C=1 TO 4
760     !
770     X=17
780      !
790     FOR I=1 TO 8
800        !
810       Bit$=B$(C)[X,X+3]
820        !
830       Bt=IVAL(Bit$,16)              ! HEX --> DIG
840       Bit$=IVAL$(Bt,2)              ! DIG --> BIN
850        !
860       IMAGE "PATTERN BIT PAGE=",AA,XXX,AAAAAAAAAAAAAAAA,XX,DDDDDD
870       PRINT USING 860;VAL$(Y);Bit$;IVAL(Bit$,2)
880        !
890       X=X+7
900       Y=Y+1
910        !
920     NEXT I
930      !
940   NEXT C
950    !
960    !
970   END
```

Executed results of pattern-data DMA transfer (Pattern data output by (BIT?) request command)

```
PATTERN BIT PAGE=1     0000000000000001        1
PATTERN BIT PAGE=2     0000000000000010        2
PATTERN BIT PAGE=3     0000000000000100        4
PATTERN BIT PAGE=4     0000000000001000        8
PATTERN BIT PAGE=5     0000000000010000       16
PATTERN BIT PAGE=6     0000000000100000       32
PATTERN BIT PAGE=7     0000000001000000       64
PATTERN BIT PAGE=8     0000000010000000      128
PATTERN BIT PAGE=9     0000000100000000      256
PATTERN BIT PAGE=10    0000001000000000      512
PATTERN BIT PAGE=11    0000010000000000     1024
PATTERN BIT PAGE=12    0000100000000000     2048
PATTERN BIT PAGE=13    0001000000000000     4096
PATTERN BIT PAGE=14    0010000000000000     8192
PATTERN BIT PAGE=15    0100000000000000    16384
PATTERN BIT PAGE=16    0111111111111111    32767
PATTERN BIT PAGE=17    1000000000000000   -32768
PATTERN BIT PAGE=18    1100000000000000   -16384
PATTERN BIT PAGE=19    1110000000000000    -8192
PATTERN BIT PAGE=20    1111000000000000    -4096
PATTERN BIT PAGE=21    1111100000000000    -2048
PATTERN BIT PAGE=22    1111110000000000    -1024
PATTERN BIT PAGE=23    1111111000000000     -512
PATTERN BIT PAGE=24    1111111100000000     -256
PATTERN BIT PAGE=25    1111111110000000     -128
PATTERN BIT PAGE=26    1111111111000000      -64
PATTERN BIT PAGE=27    1111111111100000      -32
PATTERN BIT PAGE=28    1111111111110000      -16
PATTERN BIT PAGE=29    1111111111111000       -8
PATTERN BIT PAGE=30    1111111111111100       -4
PATTERN BIT PAGE=31    1111111111111110       -2
PATTERN BIT PAGE=32    1111111111111111       -1
```

(b) DMA transfer of pattern data (Pattern data output by DMA transfer)





*1 Pattern data transfer <OUPUT Add USING "W";Dta(*)>

W: Outputs the 2's complement integer of 16 bits of data.

The upper byte of data is always transferred through the 8-Bit I/O port of the GP-IB interface board before the lower byte.

*: Outputs the entire "Dta" array.

*2 Pattern data transfer using <ENTER Add USING "W";Red(*)>

W: Outputs the 2's complement integer of 16 bits of data.

The upper byte of data is always transferred through the 8-bit I/O port of the GP-IB interface board before the lower byte.

* Data are stored in the entire "Red" array.

Program listing

```
10    !*************************************************************
20    !                                                            *
30    !               MP1702A / MP1609A / MP1651A                  *
40    !      PROGRAMMABLE PATTERN DATA DMA TRANSFER  SAMPLE SOFT   *
50    !                                                  ED_RED    *
60    !*************************************************************
70    !
80    DIM Red(31)
90    DIM Dta(31)
100   DIM Bit$[20]
110   LET Add=701               ! DEVICE ADDRESS
120   CLEAR Add                 ! DEVICE CLEAR
130    !
140   CLEAR SCREEN
150    !
160   PRINT "** MP1702A/MP1609A/MP1651A PATTERN DATA DMA TRANSFER **"
170   PRINT
180    !
190   OUTPUT Add;"PTN 1"        ! PATTERN MODE : PROG.DATA
200    !
210   OUTPUT Add;"FRM?"
220   ENTER Add;Frm$
230   IF Frm$="FRM 1" THEN OUTPUT Add;"FRM 0"
240    !
250   OUTPUT Add;"DLN 512"      ! DATA LENGTH : 512
260    !
270    !--------- SET PATTERN BIT DATA ---------
280    !
290   Dta(0)=1
300   Dta(1)=2
310   Dta(2)=4
320   Dta(3)=8
330   Dta(4)=16
340   Dta(5)=32
350   Dta(6)=64
360   Dta(7)=128
370   Dta(8)=256
380   Dta(9)=512
390   Dta(10)=1024
400   Dta(11)=2048
410   Dta(12)=4096
420   Dta(13)=8192
430   Dta(14)=16384
440   Dta(15)=32767
450   Dta(16)=-32768
460   Dta(17)=-16384
470   Dta(18)=-8192
480   Dta(19)=-4096
490   Dta(20)=-2048
500   Dta(21)=-1024
510   Dta(22)=-512
520   Dta(23)=-256
530   Dta(24)=-128
540   Dta(25)=-64
550   Dta(26)=-32
560   Dta(27)=-16
570   Dta(28)=-8
580   Dta(29)=-4
590   Dta(30)=-2
600   Dta(31)=-1
610    !
```

```
620   OUTPUT Add;"WRT 64,0"
630    !
640   OUTPUT Add USING "W";Dta(*)
650    !
660    !
670   OUTPUT Add;"RED 64,0"
680    !
690   ENTER Add USING "W";Red(*)
700    !
710   FOR I=1 TO 32
720      !
730     Bit$=IVAL$(Red(I-1),2)              ! DIG --> BIN
740      !
750     IMAGE "PATTERN BIT PAGE=",AA,XXX,AAAAAAAAAAAAAAAA,XX,DDDDDD
760     PRINT USING 750;VAL$(I);Bit$;Red(I-1)
770        !
780   NEXT I
790    !
800   END
```

Executed results of data pattern transfer by DMA (Pattern data output by DMA transfer)

```
PATTERN BIT PAGE=1     0000000000000001        1
PATTERN BIT PAGE=2     0000000000000010        2
PATTERN BIT PAGE=3     0000000000000100        4
PATTERN BIT PAGE=4     0000000000001000        8
PATTERN BIT PAGE=5     0000000000010000       16
PATTERN BIT PAGE=6     0000000000100000       32
PATTERN BIT PAGE=7     0000000001000000       64
PATTERN BIT PAGE=8     0000000010000000      128
PATTERN BIT PAGE=9     0000000100000000      256
PATTERN BIT PAGE=10    0000001000000000      512
PATTERN BIT PAGE=11    0000010000000000     1024
PATTERN BIT PAGE=12    0000100000000000     2048
PATTERN BIT PAGE=13    0001000000000000     4096
PATTERN BIT PAGE=14    0010000000000000     8192
PATTERN BIT PAGE=15    0100000000000000    16384
PATTERN BIT PAGE=16    0111111111111111    32767
PATTERN BIT PAGE=17    1000000000000000   -32768
PATTERN BIT PAGE=18    1100000000000000   -16384
PATTERN BIT PAGE=19    1110000000000000    -8192
PATTERN BIT PAGE=20    1111000000000000    -4096
PATTERN BIT PAGE=21    1111100000000000    -2048
PATTERN BIT PAGE=22    1111110000000000    -1024
PATTERN BIT PAGE=23    1111111000000000     -512
PATTERN BIT PAGE=24    1111111100000000     -256
PATTERN BIT PAGE=25    1111111110000000     -128
PATTERN BIT PAGE=26    1111111111000000      -64
PATTERN BIT PAGE=27    1111111111100000      -32
PATTERN BIT PAGE=28    1111111111110000      -16
PATTERN BIT PAGE=29    1111111111111000       -8
PATTERN BIT PAGE=30    1111111111111100       -4
PATTERN BIT PAGE=31    1111111111111110       -2
PATTERN BIT PAGE=32    1111111111111111       -1
```

(16) DMA transfer of pattern data

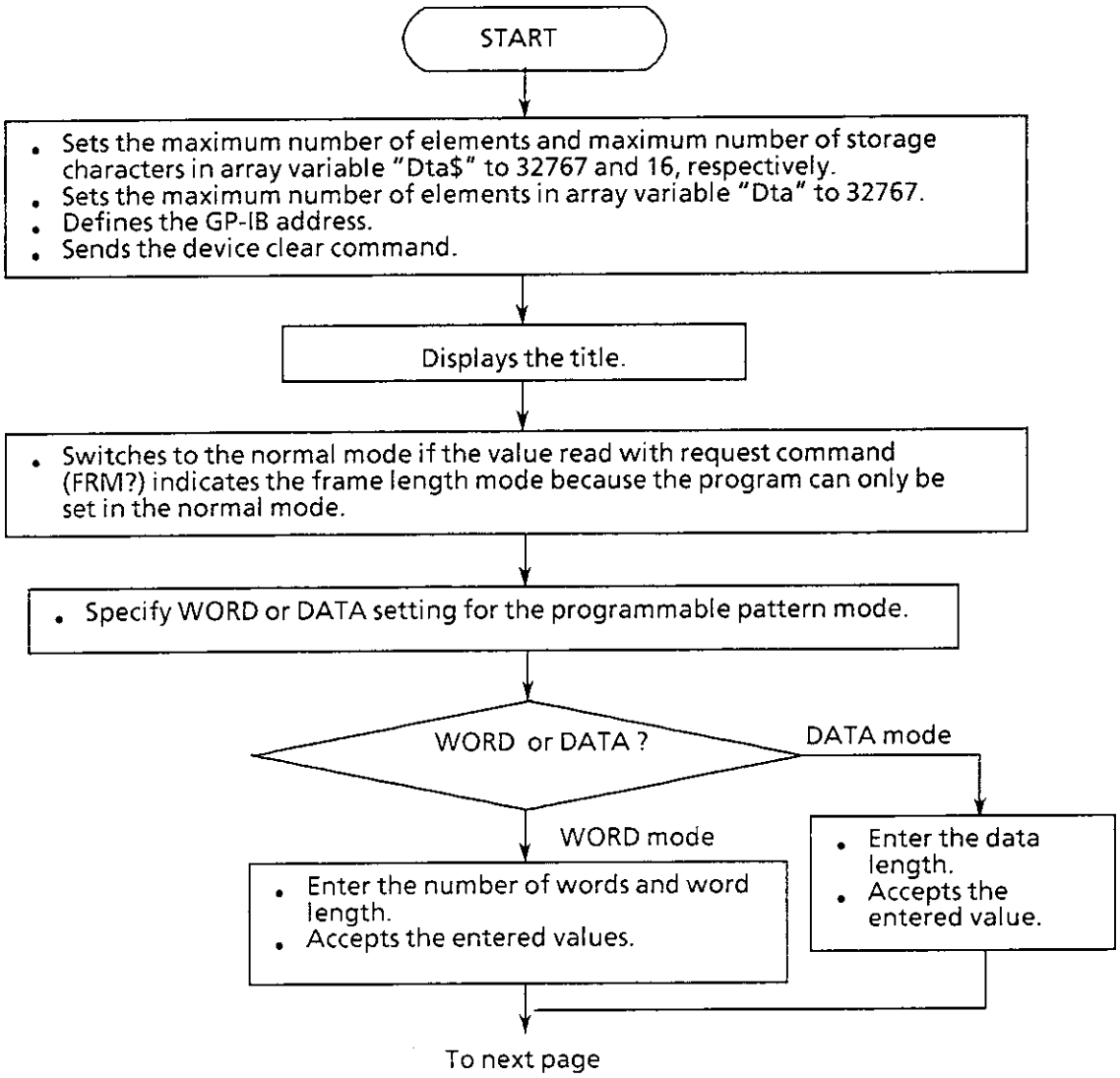
This program transfers pattern data to and from an HP9000 series computer, which is used as a controller, by DMA.

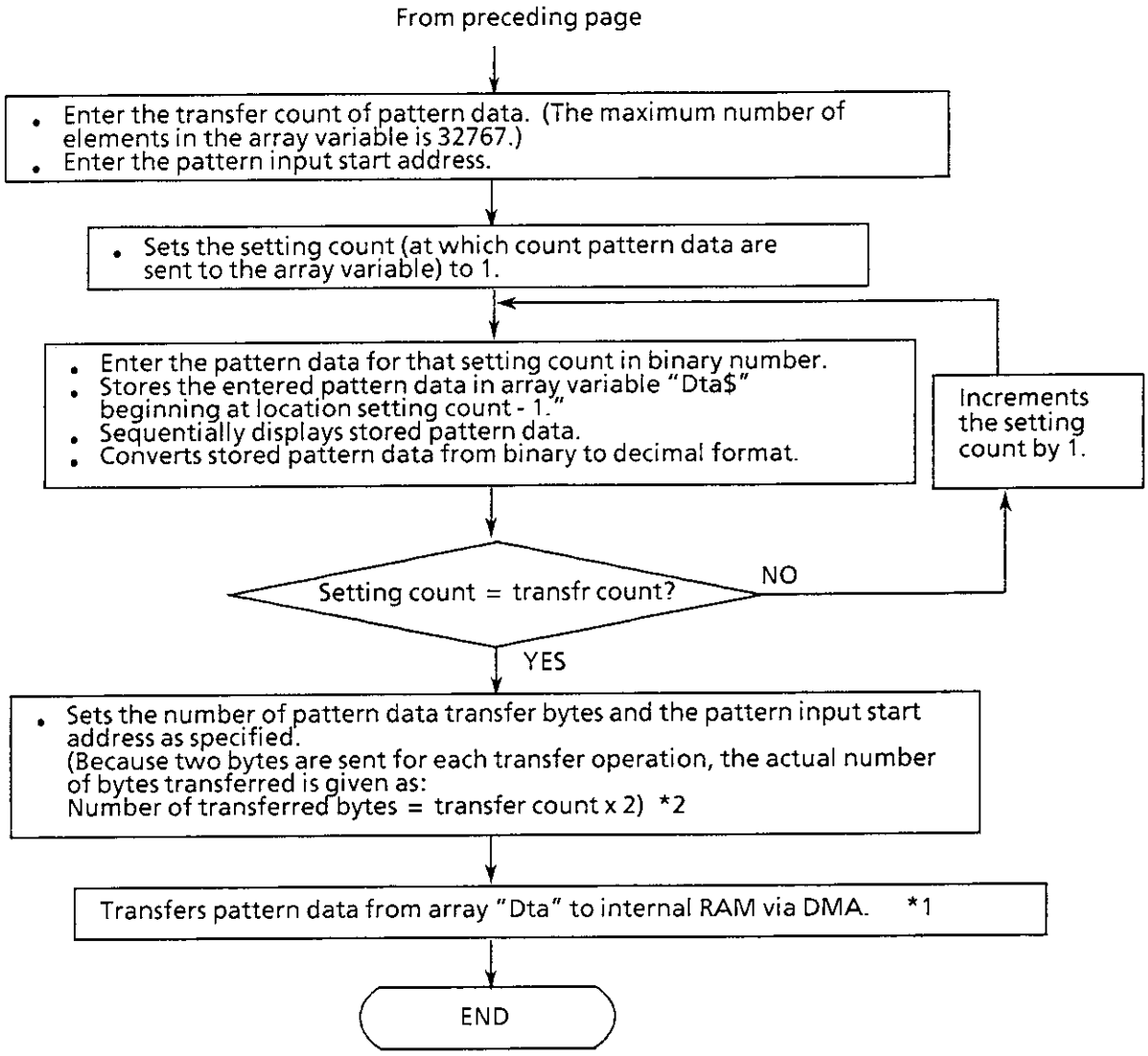
When executed, this program first selects an output pattern and initializes the transfer conditions.

Next, it defines the number of pattern data entries to be transferred (the number of pages to be transferred = transfer count) and the pattern input start address (the page at which transfer is to begin: beginning page number - 1 = pattern input start address).

Then, binary pattern data for the necessary number of pages is entered, and transferred to the MP1702A/MP1756A/MP1609A/MP1651A via DMA.

Since the HP9000 series instruction "Dta(*)" is used, data is output for all defined array elements. Therefore, it takes a long time to complete data transfer.





*1: Pattern data transfer <OUTPUT Add USING "W";Dta (*)>

W: Outputs the 2's complement of 16 bits of data.

The upper byte of data is always transferred through the 8-bit I/O port of the GP-IB interface board before the lower byte.

*: Outputs the entire "Dta" array.

*2 Number of pattern transfer bytes

Although the MP1702A/MP1756A/MP1609A/MP1651A can transfer up to 65,536 bytes of pattern data, the maximum number of array elements available in the HP9000 is 32,767.

Consequently, the maximum number of pattern data bytes that can be transferred is 32,767 × 2 = 65,534 bytes.

Program listing

```
10    !*************************************************************
20    !                                                            *
30    !               MP1702A / MP1609A / MP1651A                  *
40    !   PROGRAMMABLE PATTERN DATA DMA TRANSFER  SAMPLE SOFT      *
50    !                                                 ED_DMA2 *
60    !*************************************************************
70    !
80    !-------------------------------------------------------------!
90    !                         MAIN ROUTINE                        !
100   !-------------------------------------------------------------!
110   !
120   DIM Dta$(32766)[16]
130   DIM Dta(32766)
140   LET Add=701
150   CLEAR Add
160    !
170   PRINT "** MP1702A/MP1609A/MP1651A PATTERN DATA DMA TRANSFER **"
180   PRINT
190    !
200   OUTPUT Add;"FRM?"
210   ENTER Add;Frm$
220   IF Frm$="FRM 1" THEN OUTPUT Add;"FRM 0"
230    !
240   LOOP
250     INPUT "PROGRAMMABLE PATTERN MODE  [ WORD=0 , DATA=1 ]?",Ptn
260   EXIT IF Ptn=0 OR Ptn=1
270   END LOOP
280    !
290   OUTPUT Add;"PTN "&VAL$(Ptn)
300    !
310    !
320   IF Ptn=1 THEN
330     GOSUB Prog_d
340   ELSE
350     GOSUB Prog_w
360   END IF
370    !
380   GOSUB D_set
390    !
400   STOP
410    !
420    !-------------------------------------------------------------!
430    !                         SUB ROUTINE                         !
440    !-------------------------------------------------------------!
450    !
460 Prog_w:  !--------------------------------- PROG.Word  mode
470    !
480   PRINT "                ** PROG.WORD MODE **"
490   PRINT
500    !
510   LOOP
520     INPUT "NUMBER OF WORD DATA [ 1~32768 ] ?",Wnb
530   EXIT IF Wnb>0 AND Wnb<32769
540   END LOOP
550    !
560   OUTPUT Add;"WNB "&VAL$(Wnb)
570    !
580    !
590   LOOP
600     INPUT "WORD LENGTH [ 2~16 ] ?",Wln
610   EXIT IF Wln>1 AND Wln<17
```

```
620    END LOOP
630     !
640    OUTPUT Add;"WLN "&VAL$(Wln)
650     !
660    RETURN
670     !
680     !
690 Prog_d:  !-------------------------------- PROG.Data  mode
700     !
710    PRINT "                 ** PROG.DATA MODE **"
720    PRINT
730     !
740    LOOP
750      INPUT "DATA LENGTH [ 2^524288] ?",Dln
760    EXIT IF Dln>1 AND Dln<524289
770    END LOOP
780     !
790    OUTPUT Add;"DLN "&VAL$(Dln)
800    RETURN
810     !
820     !
830 D_set:   !---------------------------- SET Bit pattern data
840     !
850    INPUT "PATTERN DATA SETTING COUNT ? [ MAX 32767 ]",Cnt
860    INPUT "PATTERN DATA WRITE TOP ADDRESS ? [ 0 ~ 32767 ]",Top$
870     !
880    FOR I=1 TO Cnt
890       !
900      INPUT "BIT PATTERN DATA  BIT16-->BIT1 [0/1]?",Dta$(I-1)
910      PRINT "BIT PATTERN DATA= "&Dta$(I-1)
920       !
930      Dta(I-1)=IVAL(Dta$(I-1),2)          ! BIN --> DIG
940       !
950    NEXT I
960     !
970    OUTPUT Add;"WRT "&VAL$(Cnt*2)&","&Top$
980     !
990    OUTPUT Add USING "W";Dta(*)
1000    !
1010   RETURN
1020    !
1030    !
1040   END
```

15.5 Sample Programs for the COMPAQ as Controller

<Description of common parts of sample programs>

① <common shared IBSTA%, IBERR, IBCNT> is defined at the beginning of the program for GP-IB control.

The statement in < > is a command of QUICK BASIC of the COMPAQ, and is a necessary definition for the usage of GP-IB driver (software).

② *1 500 ms WAIT after interface clear is executed

After an interface clear is executed, an approximately 500 ms WAIT is inserted during which the MP1702A/MP1756A/MP1609A/MP1651A receives and executes the interface clear command from the COMPAQ and becomes stable.

③ *2 Trap

When a GP-IB commmand is received from the COMPAQ or when data is read from the MP1702A/MP1756A/MP1609A/MP1651A, normal command execution is verified. If an error occurs, error information is displayed and the program is aborted.

Similar processing is also performed for IBFIND (device definition), IBSIC (interface clear), and IBRSP (serial polling).

④ wrtcmd: '-----Write comamnd -----

wrt1$ = wrt1$ + chr$(13) + chr$(10)

CR, LF are appended to the end of the data stored in variable wrt1$ and reassigned to the same variable.

CR: chr$(13)

LF: chr$(10)

CALL IBWRT(ED%, wrt1$) sends the data stored in variable wrt1$ to the MP1702A/MP1756A/MP1609A/MP1651A.

⑤ readcmd:'----- Read command -----

rd$ = SPACE$(○○)

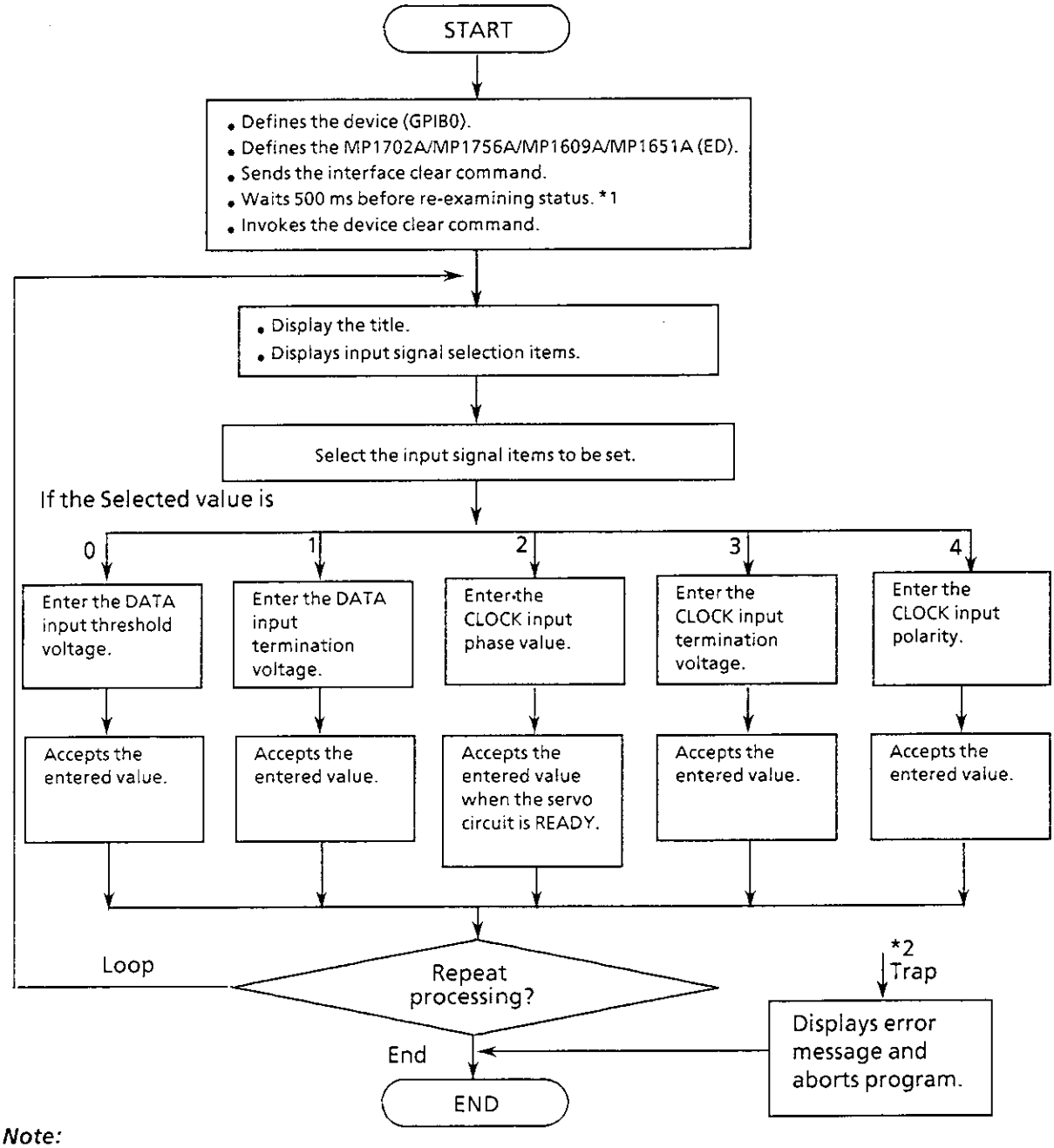
Initializes variable rd$ that stores the data read from the MP1702A/MP1756A/MP1609A/MP1651A to spaces. The ○○ in SPACE$ (○○) represent [number of characters of read data + 2 characters (CR and LF)].

Subroutine CALL IBRD(ED%,rd$) reads data from the MP1702A/MP1756A/MP1609A/MP1651A and stores it in variable rd$.

(1) Setting the input signal

This program controls input signal conventions and characteristics.

Values of various MP1702A/MP1756A/MP1609A/MP1651A signals are input after the appropriate selection has been made (DATA and CLOCK voltages, phases, polarities,etc.). Note that when setting the clock input phase, the program enters a delay state to ensure that the instrument is READY before applying the delay value.



Note:

See the preceding page for *1 and *2.

Program listing

```
' ************************************************************
' *                                                          *
' *      MP1702A/MP1609A/MP1651A  INPUT SIGNAL SAMPLE        *
' *                                                          *
' *                                                 ED_IN    *
' ************************************************************
'
common shared IBSTA%,IBERR%,IBCNT%      ' Setup GPIB-PC functions
GOSUB gpinit                            ' Setup GPIB interface
'
DO
  CLS
  '
  PRINT "** MP1702A/MP1609A/MP1651A INPUT SIGNAL SAMPLE SOFT **"
  PRINT
  PRINT " INPUT SIGNAL * DATA THRESHOLD = [0] "
  PRINT "              * DATA TERMINATION = [1] "
  PRINT "              * CLOCK PHASE ADJUST = [2] "
  PRINT "              * CLOCK TERMINATION = [3] "
  PRINT "              * CLOCK POLARITY = [4] "
  PRINT
  INPUT "SELECT INPUT SIGNAL [ 0 or 1 or 2 or 3 or 4 ]";inp$
  PRINT
  '
  SELECT CASE inp$
    '
  CASE "0"
      INPUT "DATA THRESHOLD [-3.000 ~ +1.750 V ] STEP 0.002 V ";dth$
      wrt$ = "DTH " + dth$ : GOSUB wrtcmd
      '
  CASE "1"
      INPUT "DATA TERMINATION [ GND=0 , -2V=1 ] ";dtm$
      wrt$ = "DTM " + dtm$ : GOSUB wrtcmd
      '
  CASE "2"
      PRINT "       <<<  CLOCK PHASE ADJUST >>>"
      PRINT "MP1702A/MP1609A --> -500 ~  500  ps  1ps STEP"
      PRINT "MP1651A -->        -1000 ~ 1000  ps  2ps STEP"
      PRINT
      INPUT "CLOCK PHASE ADJUST";cpa$
      '
      DO
        wrt$="DLY?" : GOSUB wrtcmd : GOSUB readcmd
      LOOP UNTIL MID$(rd$,1 ,5)="DLY 0"
      '
      wrt$ = "CPA " + cpa$ : GOSUB wrtcmd
      '
  CASE "3"
      INPUT "CLOCK TERMINATION [ GND=0 , -2V=1 ] ";ctm$
      wrt$ = "CTM " + ctm$ : GOSUB wrtcmd
      '
  CASE "4"
      INPUT "CLOCK POLARITY [ CLK=0 , NCLK=1 ] ";cpl$
      wrt$ = "CPL " + cpl$ : GOSUB wrtcmd
      '
  END SELECT
  '
 `INPUT "NEXT DATA SET  [ YES=0 , NO=1 ]";loop$
  '
LOOP UNTIL loop$ = "1"
'
STOP
```

```
'
gpinit: ' ---------- Setup GPIB interface  ----------
'
    CALL IBFIND("GPIB0", GPIB0%)              ' Open device (GPIB0)
    IF GPIB0% < 0 THEN GOTO trap              ' system error
    '
    CALL IBFIND("ED", ED%)                    ' Open device (ED)
    IF ED% < 0 THEN GOTO trap                 ' system error
    '
    CALL IBSIC(GPIB0%)                        ' Interface clear
    IF IBSTA% < 0 THEN GOTO trap              ' system error
    '
    tim = 0.5
    GOSUB waidly
    '
    CALL IBCLR(ED%)                           ' Device clear
    '
    RETURN
'
wrtcmd: ' ----------  Write command  ----------
'
    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(ED%, wrt$)                     ' Write command
    IF IBSTA% < 0 THEN GOTO trap              ' Trap
    '
    RETURN
'
readcmd: ' ----------  Read command  ----------
'
    rd$=SPACE$(7)
    CALL IBRD(ED%, rd$)                       ' Read command
    IF IBSTA% < 0 THEN GOTO trap              ' Trap
    '
    RETURN
'
waidly: ' ----------  Wait delay  ----------
'
    stm = TIMER
    etm = TIMER
    WHILE etm - stm < tim
        etm = TIMER
        IF  etm < stm THEN etm = stm + 86400
    WEND
    '
    RETURN
'
trap:   ' ----------  System trap  ----------
'
    PRINT "IBERR%:" + STR$(IBERR%)
    STOP
    '
    '
    END
```

(2) Enabling automatic search for input thresholds (auto search)

This program executes "auto search" after the MP1702A/MP1756A/MP1609A/MP1651A has been connected to the MP1701B/MP1755A/MP1608A/MP1650A.

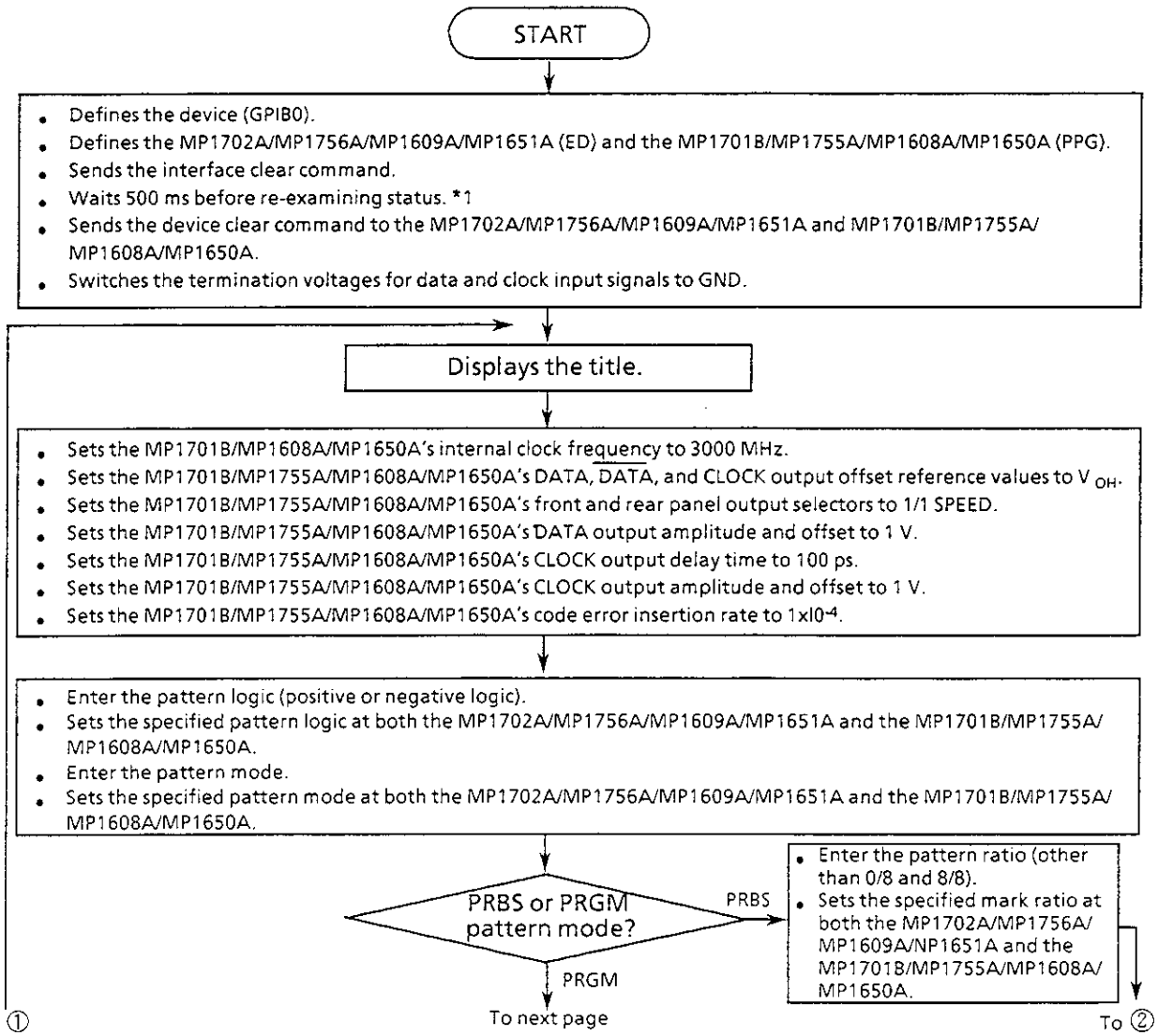
First, the program sets the conditions necessary to execute auto search on the MP1702A/MP1756A/MP1609A/MP1651A and MP1701B/MP1755A/MP1608A/MP1650A.

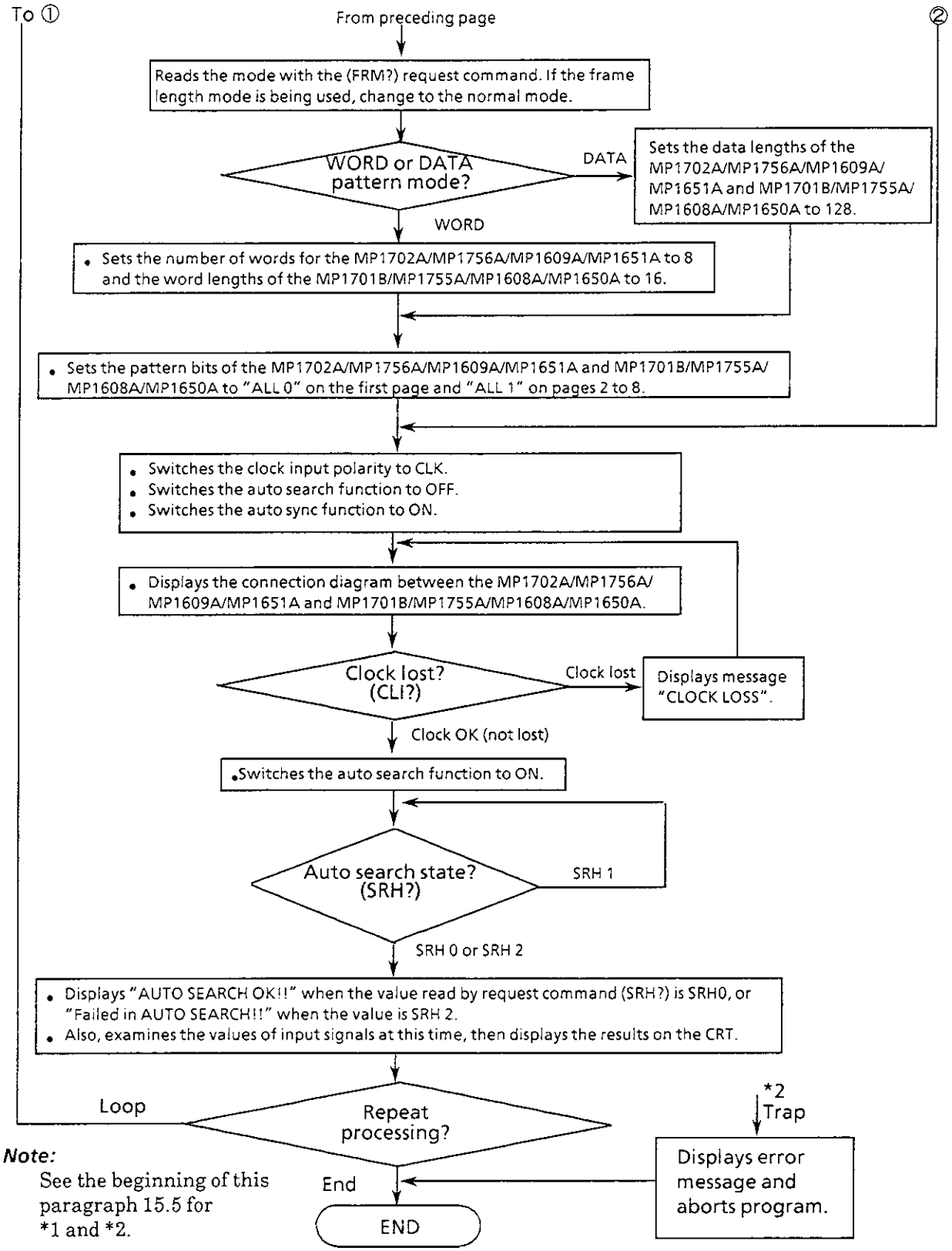
Next, it confirms that the clock is not lost (since auto search will not work if the clock is lost), and then turns ON the AUTO SEARCH function.

Afterwards, it reads the status of the AUTO SEARCH with the SRH? request command and displays this status on the CRT screen.

Finally, it reads the values of input signal and displays results on the CRT screen.

Note: When using the MP1755A for the transmitter, internal clock frequency setting is ineffective. Set the frequency of the external clock source to 3000 MHz in advance.





Note:

See the beginning of this paragraph 15.5 for *1 and *2.

Program listing

```
' ************************************************************
' *                                                          *
' *     MP1702A/MP1609A/MP1651A  AUTO SEARCH SAMPLE SOFT     *
' *                                                          *
' *                                                ED_SRCH   *
' ************************************************************
'
'------------------------------------------------------------'
'                          MAIN  ROUTINE                     '
'------------------------------------------------------------'
'
common shared IBSTA%,IBERR%,IBCNT%   ' Setup GPIB-PC functions
GOSUB gpinit                         ' Setup GPIB interface
wrt$="DTM 0;CTM 0" : GOSUB wrtcmd2
'
SCREEN 9
'
DO
    CLS
    PRINT "** MP1702A/MP1609A/MP1651A **"
    PRINT "   AUTO SEARCH SAMPLE SOFT"
    PRINT
    '
    GOSUB DSet                       ' Data setting
    GOSUB Clock                      ' Check Clock loss ?
    GOSUB Srch                       ' Auto search : ON
    GOSUB Result                     ' Display Result
    '
    INPUT "NEXT DATA SET [ YES=0, NO=1 ]";loop$
    '
LOOP UNTIL loop$="1"
'
STOP
'
'------------------------------------------------------------'
'                          SUB ROUTINE                       '
'------------------------------------------------------------'
  '
DSet:  '---------- Data Setting ----------
    '
    wrt$= "CLK 1;RES 1;FRQ 3000" : GOSUB wrtcmd1
    wrt$= "OFS 0;SPD 0"          : GOSUB wrtcmd1
    wrt$= "DAP 1;DOS 1"          : GOSUB wrtcmd1
    wrt$= "CDL 100;CAP 1;COS 1"  : GOSUB wrtcmd1
    wrt$= "EAD 1"                : GOSUB wrtcmd1
    '
    INPUT "LOGIC [ POSITIVE=0, NEGATIVE=1 ]";lgc$
    wrt$="LGC "+lgc$          : GOSUB wrtcmd1
    wrt$="LGC "+lgc$          : GOSUB wrtcmd2
    '
    PRINT "PATTERN MODE [ WORD=0 ]"
    PRINT "              [ DATA=1 ]"
    PRINT "              [ PN7=2 ]"
    PRINT "              [ PN9=3 ]"
    PRINT "              [ PN11=5 ]"
    PRINT "              [ PN15=6 ]"
    PRINT "              [ PN20=7 ]"
    PRINT "              [ PN23=8 ]"
    PRINT "              [ PN31=9 ]"
    PRINT
    '
```

```
    DO
      INPUT "PATTERN MODE [ 0~9 (except 4) ]";ptn
    LOOP UNTIL ptn<>4 AND ptn>=0 AND ptn<=9
    '
    wrt$="PTN "+STR$(ptn)     : GOSUB wrtcmd1
    wrt$="PTN "+STR$(ptn)     : GOSUB wrtcmd2
    '
    IF ptn=0 or ptn=1 THEN
        GOSUB Prog
    ELSE
        GOSUB Prbs
    END IF
    '
    wrt$="CPL 0"              : GOSUB wrtcmd2
    wrt$="SRH 0"              : GOSUB wrtcmd2
    wrt$="SYN 1"              : GOSUB wrtcmd2
    '
    RETURN
    '
Clock: '---------- Set data , Check connection ----------
    '
    DO
        GOSUB Connect                ' Display device connection
        '
        wrt$="CLI?" : GOSUB wrtcmd2 : GOSUB readcmd
        '
        IF MID$(rd$,1,5)="CLI 1" THEN PRINT "** CLOCK LOSS **"
        '
    LOOP UNTIL MID$(rd$,1,5)="CLI 0"
    '
    RETURN
    '
Srch: '---------- Auto search ON ----------
    '
    wrt$="SRH 1" : GOSUB wrtcmd2       ' Set Auto search : ON
    '
    DO
        wrt$="SRH?" : GOSUB wrtcmd2 : GOSUB readcmd
    LOOP UNTIL MId$(rd$,1,5)="SRH 0" or MId$(rd$,1,5)="SRH 2"
    '
    IF MID$(rd$,1,5)="SRH 0"  THEN
        PRINT "** AUTO SEARCH OK **"
    ELSE
        PRINT "** Failed in AUTO SEARCH **"
    END IF
    '
    RETURN
    '
Result: '---------- display Result ----------
    '
    wrt$="DTH?" : GOSUB wrtcmd2 : GOSUB readcmd
    PRINT "DATA THRESHOLD = "+MID$(rd$,5,6)+" V"
    '
    wrt$="DTM?" : GOSUB wrtcmd2 : GOSUB readcmd
    IF MID$(rd$,1,5)="DTM 0" THEN
        rd$="GND"
    ELSE
        rd$="-2V"
    END IF
    PRINT "DATA TERMINATION = "+rd$
    '
    wrt$="CPA?" : GOSUB wrtcmd2 : GOSUB readcmd
    PRINT "CLOCK PHASE ADJUST = "+MID$(rd$,6,4)+" ps"
    '
```

```
    wrt$="CTM?" : GOSUB wrtcmd2 : GOSUB readcmd
    IF MID$(rd$,1,5)="CTM 0" THEN
        rd$="GND"
    ELSE
        rd$="-2V"
        GOSUB readcmd
    END IF
    PRINT "CLOCK TERMINATION = "+rd$
    '
    wrt$="CPL?" : GOSUB wrtcmd2 : GOSUB readcmd
    IF MID$(rd$,1,5)="CPL 0" THEN
        rd$="CLK"
    ELSE
        rd$="N CLK"
    END IF
    PRINT "CLOCK POLARITY = "+rd$
    '
    RETURN
    '
Prog: '---------- Set PROG (WORD),(DATA) ----------
    '
    wrt$="FRM?" : GOSUB wrtcmd2 : GOSUB readcmd
    IF MID$(rd$,1,5)="FRM 1" THEN wrt$="FRM 0" : GOSUB wrtcmd2
    '
    IF ptn=1 THEN
        PRINT"** PATTERN MODE PROG (DATA) **"
        PRINT
        wrt$="DLN 128" : GOSUB wrtcmd1
        wrt$="DLN 128" : GOSUB wrtcmd2
    ELSE
        PRINT"** PATTERN MODE PROG (WORD) **"
        PRINT
        wrt$="WNB 8;WLN 16" : GOSUB wrtcmd1
        wrt$="WNB 8;WLN 16" : GOSUB wrtcmd2
    END IF
    '
    wrt$="PAG 1;PST 0" : GOSUB wrtcmd1
    wrt$="PAG 1;PST 0" : GOSUB wrtcmd2
    '
    FOR I=2 TO 8
        wrt$="PAG "+STR$(I)+";PST 1" : GOSUB wrtcmd1
    NEXT I
    '
    FOR I=2 TO 8
        wrt$="PAG "+STR$(I)+";PST 1" : GOSUB wrtcmd2
    NEXT I
    '
    RETURN
    '
Prbs: '---------- Set Mark ratio ----------
    '
    DO
        INPUT "MARK RATIO [ 1/8:7/8=1, 1/4:3/4=2, 1/2:N1/2=3 ]";mrk
    LOOP UNTIL mrk=1 or mrk=2 or mrk=3
    wrt$="MRK "+STR$(mrk) : GOSUB wrtcmd1
    wrt$="MRK "+STR$(mrk) : GOSUB wrtcmd2
    '
    RETURN
    '
gpinit: ' ---------- Set up GP-IB functions  ----------
    '
    CALL IBFIND("GPIB0", GPIB0%)          ' Open device (GPIB0)
    IF GPIB0% < 0 THEN GOTO trap          ' system error
    '
    CALL IBFIND("PPG", PPG%)              ' Open device(PPG)
    IF PPG% < 0 THEN GOTO trap            ' system error
```

```
     CALL IBFIND("ED", ED%)                  ' Open device(ED)
     IF ED% < 0 THEN GOTO trap               ' system error
     '
     CALL IBSIC(GPIB0%)                      ' Interface clear
     IF IBSTA% < 0 THEN GOTO trap            ' system error
     '
     tim = 0.5
     GOSUB waidly
     '
     CALL IBCLR(PPG%)                        ' Device clear(PPG)
     '
     CALL IBCLR(ED%)                         ' Device clear(ED)
     '
     RETURN
          '
wrtcmd1: ' ----------  Write command (PPG) ----------
          '
     wrt$=wrt$+chr$(13)+chr$(10)
     CALL IBWRT(PPG%, wrt$)                  ' Write command (PPG)
     IF IBSTA% < 0 THEN GOTO trap            ' Trap
     '
     RETURN
          '
wrtcmd2: ' ----------  Write command (ED) ----------
          '
     wrt$=wrt$+chr$(13)+chr$(10)
     CALL IBWRT(ED%, wrt$)                   ' Write command (ED)
     IF IBSTA% < 0 THEN GOTO trap            ' Trap
     '
     RETURN
          '
readcmd: ' ---------- Read command (ED) ----------
          '
     rd$=SPACE$(12)
     CALL IBRD(ED%, rd$)                     ' Read command (ED)
     IF IBSTA% < 0 THEN GOTO trap            ' Trap
     '
     RETURN
          '
waidly:  ' ----------  Wait delay  ----------
          '
     stm = TIMER
     etm = TIMER
     WHILE etm - stm < tim
          etm = TIMER
          IF etm < stm THEN etm = etm + 86400
     WENd
     '
     RETURN
          '
trap:    ' ----------  System trap  ----------
          '
     PRINT "IBERR%:" + STR$(IBERR%)
     STOP
          '
Connect: '---------- display Connection ----------
          '
     WINDOW (-600,-500) -(600,500)
     '
     LINE (-50,-50) - (600,500),14,B
     LINE (10,100) - STEP(235,200),,B
     LINE (300,100) - STEP(235,200),,B
     '
```

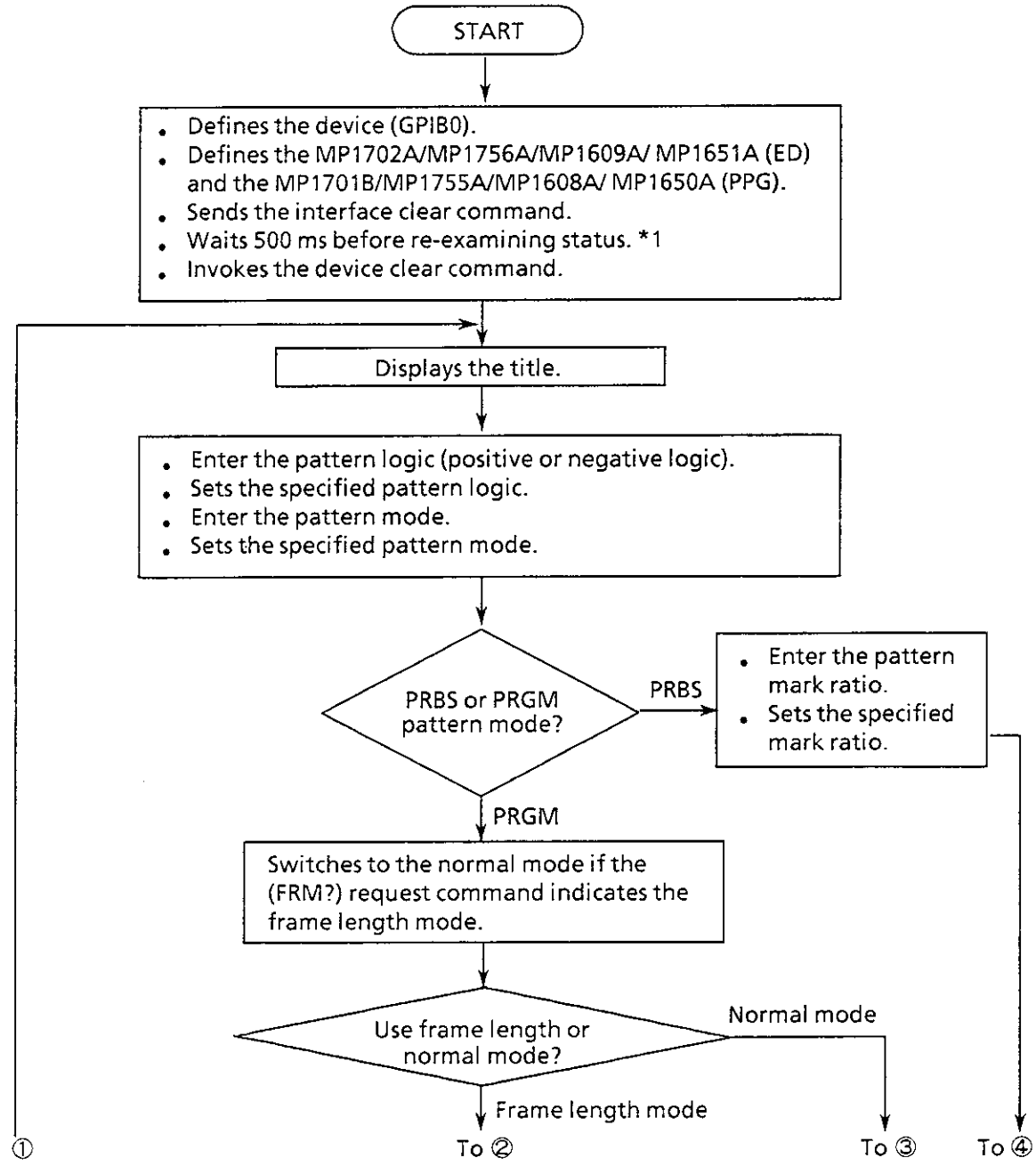
```
CIRCLE (130,130),8
LINE (130,125) - STEP(0,-55)
LINE (130,70) - STEP(220,0)
LINE (350,70) - STEP(0,55)
CIRCLE (350,130),8
'
CIRCLE (190,130),8
LINE (190,125) - STEP(0,-100)
LINE (190,25) - STEP(220,0)
LINE (410,25) - STEP(0,100)
CIRCLE (410,130),8
'
LOCATE 2,50
PRINT "<<  CONNECTION  >>"
'
LOCATE 4,42: PRINT "MP1701B/MP1608A     MP1702A/MP1609A"
LOCATE 5,42: PRINT "     /MP1650A            /MP1651A"
'
LOCATE 9,45: PRINT "DATA  CLOCK1"
'
LOCATE 9,62: PRINT "DATA  CLOCK"
'
LOCATE 23,1
INPUT "Aer you ready ?  Press return key to start.",A
LOCATE 23,1
PRINT "                                                  "
'
LOCATE 18,1
'
RETURN
'
END
```

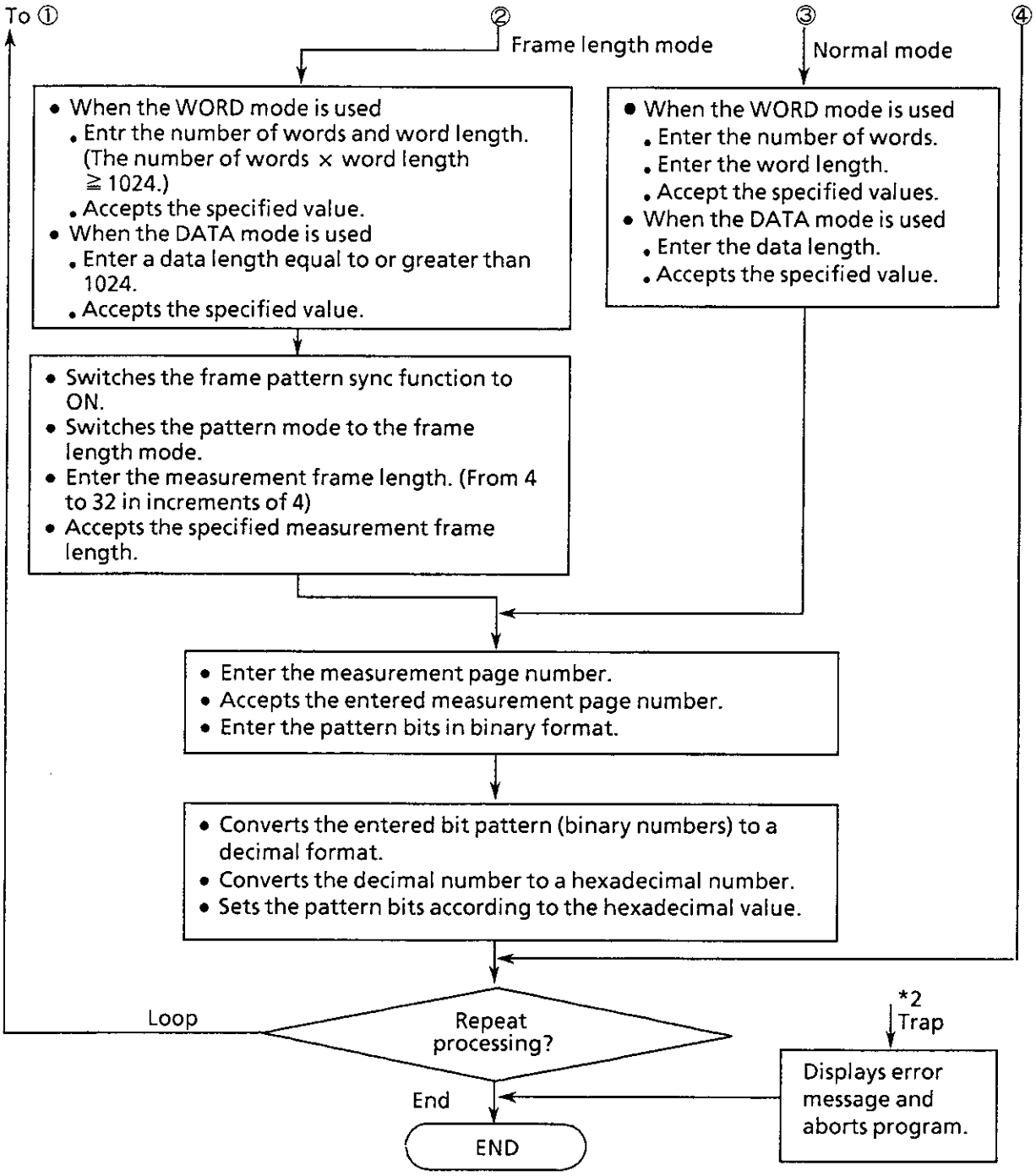
(3) Setting measurement patterns

This program is used to control the characteritics and features of measurement patterns.

First, the program selects the measurement pattern logic and pattern mode. Then, it sets necessary conditions for each measurement pattern mode. However, if the PRGM mode is set, either the WORD (normal mode) or FRAME (frame length mode) data length may be selected along with the conditions for that mode.

Note that the hexadecimal pattern data from the previously entered page number is set in the PRGM mode.





Note: See the beginning of this paragraph 15.5 for *1 and *2.

Program listing

```
' ************************************************************
' *                                                          *
' *      MP1702A/MP1609A/MP1651A  PATTERN  SAMPLE SOFT       *
' *                                                          *
' *                                                 ED_PAT   *
' ************************************************************
'
'-----------------------------------------------------------'
'                          MAIN  ROUTINE                    '
'-----------------------------------------------------------'
'
common shared IBSTA%,IBERR%,IBCNT%   ' Setup GPIB-PC functions
GOSUB gpinit                         ' Setup GPIB interface
  '
DO
    CLS
    '
    PRINT "** MP1702A/MP1609A/MP1651A PATTERN SAMPLE SOFT **"
    PRINT
    '
    GOSUB pattern
    '
    PRINT
    INPUT "NEXT DATA SET [ YES=0 , NO=1 ]";loop$
    '
LOOP UNTIL loop$ = "1"
'
STOP
'
'
'-----------------------------------------------------------'
'                          SUB ROUTINE                      '
'-----------------------------------------------------------'
    '
pattern:'----------  Set Logic,Pattern mode ----------
    '
    INPUT "LOGIC MODE [ POSITIV=0 , NEGATIV=1 ]";lgc$
    PRINT
    WRT$= "LGC "+lgc$ : GOSUB wrtcmd
    '
    PRINT "PATTERN MODE [ WORD=0, DATA=1, PN7=2, PN9=3, PN11=5"
    PRINT "               PN15=6, PN20=7, PN23=8, PN31=9 ]"
    PRINT
    '
    DO
      INPUT "PATTERN MODE [ 0 or 1 or 2 or 3 or 5 or 6 or 7 or 8 or 9 ] ";ptn
    LOOP UNTIL ptn <> 4 AND ptn>=0 AND ptn<=9
    '
    wrt$ = "PTN "+STR$(ptn) : GOSUB wrtcmd
    '
    IF ptn = 0 OR ptn =1 THEN
        GOSUB ProgMode
    ELSE
      '
      DO
        INPUT "MARK RATIO [ 0/8:8/8=0, 1/8:7/8=1, 1/4:3/4=2, 1/2:N1/2=3 ]"; mrk
      LOOP UNTIL mrk>=0 AND mrk<=3
      wrt$="MRK "+STR$(mrk) : GOSUB wrtcmd
    END IF
    '
    RETURN
```

```
ProgMode: '---------- Select FRAME/WORD & Set Page,Bit ----------
      '
      wrt$="FRM?" : GOSUB wrtcmd : GOSUB readcmd
      IF MID$(rd$,1,5)="FRM 1" THEN wrt$="FRM 0" : GOSUB wrtcmd
      '
      DO
        INPUT "FRAME/WORD LENGTH MODE ? [ WORD=0 , FRAME=1 ]";mode
      LOOP UNTIL mode=0 or mode=1
      '
      IF mode=1 THEN
        GOSUB Frmlen
      ELSE
        GOSUB Wrdlen
      END IF
      '
      INPUT "PAGE ";pag$
      wrt$="PAG "+pag$ : GOSUB wrtcmd
      '
      INPUT "BIT PATTERN  BIT 16 --> BIT 1 [0/1] ";bit$
      GOSUB BtoH
      wrt$ = "BIT #H"+B$ : GOSUB wrtcmd
      '
      RETURN
      '
Frmlen: '---------- Frame length mode ----------
      '
      PRINT "*** Setting FRAME LENGTH mode ***"
      PRINT
      '
      IF ptn=1 THEN
          '
          DO
            INPUT "DATA LENGTH [ 1024 ~ 524288 ] ";dln
          LOOP UNTIL dln>=1024 AND dln<=524288
          wrt$="DLN "+ STR$(dln)
          '
      ELSE
          PRINT "*1 <<  NUMBER OF WORD x WORD LENGTH >=1024  >>"
          PRINT
          '
          DO
              DO
                INPUT "NUMBER OF WORD [ 64 ~ 32768 ]  *1  ";wnb
              LOOP UNTIL wnb>=64 AND wnb<=32768
              wrt$ = "WNB "+STR$(wnb) : GOSUB wrtcmd
              '
              DO
                INPUT "WORD LENGTH [ 2 ~ 16 ] ";wln
              LOOP UNTIL wln>=2 AND wln<=16
              wrt$ = "WLN "+STR$(wln) : GOSUB wrtcmd
          '
          LOOP UNTIL wnb*wln>=1024
      '
      END IF
      '
      wrt$="FSY 1" : GOSUB wrtcmd          ' Frame sync : ON
      wrt$="FRM 1" : GOSUB wrtcmd          ' Frame/Word length : FRAME
      '
      DO
        INPUT "FRAME LENGTH  [ 4 ~ 32 ] STEP 4 ";fln
        f=fln-4*INT(fln/4)
      LOOP UNTIL fln>=4 AND fln<=32 AND f=0
      '
```

```
    wrt$="FLN "+STR$(fln) : GOSUB wrtcmd
    '
    RETURN
    '
Wrdlen: '---------- Word (Normal) length mode ----------
    '
    PRINT "*** Setting NORMAL LENGTH mode ***"
    PRINT
    '
    IF ptn=1 THEN
        '
        DO
          INPUT "DATA LENGTH [ 2 ~ 524288 ] ";dln
        LOOP UNTIL dln>=2 AND dln<=524288
        wrt$="DLN "+ STR$(dln)
        '
    ELSE
        DO
          INPUT "NUMBER OF WORD [ 1 ~ 32768 ] *1 ";wnb
        LOOP UNTIL wnb>=1 AND wnb<=32768
        wrt$ = "WNB "+STR$(wnb) : GOSUB wrtcmd
        '
        DO
          INPUT "WORD LENGTH [ 2 ~ 16 ] ";wln
        LOOP UNTIL wln>=2 AND wln<=16
        wrt$ = "WLN "+STR$(wln) : GOSUB wrtcmd
        '
    END IF
    '
    RETURN
    '
BtoH:   ' ----------  Bin to Hex     ----------
    '
    A=0
    FOR I=15 TO 0 STEP -1
        IF MID$(bit$,16-I,1) = "1" THEN
            A = A + 2^I
        END IF
    NEXT I
    '
    B$ = HEX$(A)
    '
    RETURN
    '
gpinit: ' ---------- Set up GP-IB functions  ----------
    '
    CALL IBFIND("GPIB0", GPIB0%)          ' Open device (GPIB0)
    IF GPIB0% < 0 THEN GOTO trap          ' system error
    '
    CALL IBFIND("ED", ED%)                ' Open device(ED)
    IF ED% < 0 THEN GOTO trap             ' system error
    '
    CALL IBSIC(GPIB0%)                    ' Interface clear
    IF IBSTA% < 0 THEN GOTO trap          ' system error
    '
    tim = 0.5
    GOSUB waidly
    '
    CALL IBCLR(ED%)                       ' Device clear
    '
    RETURN
    '
```

```
wrtcmd:  ' ----------  Write command  ----------
         '
     wrt$=wrt$+chr$(13)+chr$(10)
     CALL IBWRT(ED%, wrt$)                ' Write command
     IF IBSTA% < 0 THEN GOTO trap         ' Trap
     '
     RETURN
         '
readcmd: ' ----------  Read command  ----------
         '
     rd$=SPACE$(7)
     CALL IBRD(ED%, rd$)                  ' Read command
     IF IBSTA% < 0 THEN GOTO trap         ' Trap
     '
     RETURN
         '
waidly:  ' ----------  Wait delay  ----------
         '
     stm = TIMER
     etm = TIMER
     WHILE etm - stm < tim
         etm = TIMER
         IF etm < stm THEN etm = etm + 86400
     WEND
     '
     RETURN
         '
trap:    ' ----------  System trap  ----------
         '
     PRINT "IBERR%:" + STR$(IBERR%)
     STOP
     '
     END
```

(4) Measurement (Measurement type: Outputs data each time a measurement is completed)

The following paragraphs (a) to (g) shown in the next pages are created for uses where the measurement result GP-IB output selection is set for output to be made each time a measurement is completed.

Measurement is initiated after connecting DATA to DATA and CLOCK1 to CLOCK between the MP1702A/MP1756A/MP1609A/MP1651A and MP1701B/MP1755A/MP1608A/MP1650A.

First, the program sets the conditions required for auto search and for measurement.

Next, it confirms that the clock has not been lost and, after confirmation, switches the auuto search function ON. Then, provided that auto search is successfully completed, measurement starts. When measurement is completed, measurement results are read and displayed on CRT.

Read terminate conditions depend on the print format.

If auto search fails, the program is aborted.

Table 15-6 shows methods for determining the end of measurement, the print format of measurement results, and whether one-second data are to be printed or not. Sample programs are provided to illustrate each case.

Table 15-6

Paragraph	Data print format	Data output format (Standard or extended)	Method of determining end of measurement	Print one-second data
(a)	Standard	Standard	Serial polling	As selected
(b)	Standard	Standard	by STB? request command	No
(c)	Standard	Standard	by MSR? request command	No
(d)	Standard	Extended: data including threshold measurement and error performance measurement data	Serial polling	No
(e)	Standard	Extended: data including threshold measurement data	Serial polling	No
(f)	Standard	Extended: data including error performance measurement data	Serial polling	No
(g)	Short (abridged)	Standard	Serial polling	No

Note 1: In programs in paragraphs (b) and (c), the end of measurement is determined by the (STB?) or (MSR?) request command. Therefore, since request commands are continuously sent to the MP1702A/MP1756A/MP1609A/MP1651A throughout the measurement, it remains as talker for the duration of the measurement.

Therefore, measurement results are continuously overwirtten by the (STB?) or (MSR?) request command. To prevent data loss, the results of the (STB?) or (MSR?) command are read and followed by an interface clear command which cancels the talker specification so that measurement data can be output and read.

To simplify programming, the end of measurement should be determined by serial polling.

Note 2: READ command to read measurement results

When detecting end of measurement data on COMPAQ computers with device function mode CALL IBRD (ED%, rd$), it may not be possible to read data because the computer issues untalk and unlisten commands when it detects the measurement terminator. To prevent this, the following adapter function mode is used to read measurement results in this program.

Note 3: When using the MP1755A for the transmitter, internal clock frequency setting is ineffective. Set the frequency of the external clock source to 3000 MHz in advance.

< READ command in adapter function mode >

```
100  CMD$="?"+chr$(42)+"A"
110  CALL IBCMD(GPIB0%,CMD$)
120  rd$=SPACE$(○○)
130  CALL IBRD(GPIB0%,rd$)
140  IF IBSTA% <0 THEN GOTO trap
```

100 ? : Unlisten (This command specifies the MP1702A/MP1756A/MP1609A/MP1651A for unlisten mode.)

chr$(42) : Listener address (Corresponds to the primary address of IBCONF adapter (GPIB0) that is set to 10 in paragraph 15.3. Change the listener address to match the primary address.)

A : Talker address (Corresponds to the primary address of IBCONF device (ED) and the address of MP1702A/MP1756A/MP1609A/MP1651A, both set to 1 in paragraph 15.3. Change the talker address to match the primary address and the MP1702A/MP1756A/MP1609A/MP1651A address.)

110 The commands on line 100 are sent to the (GPIB0) adapter and specify the talker and listener.

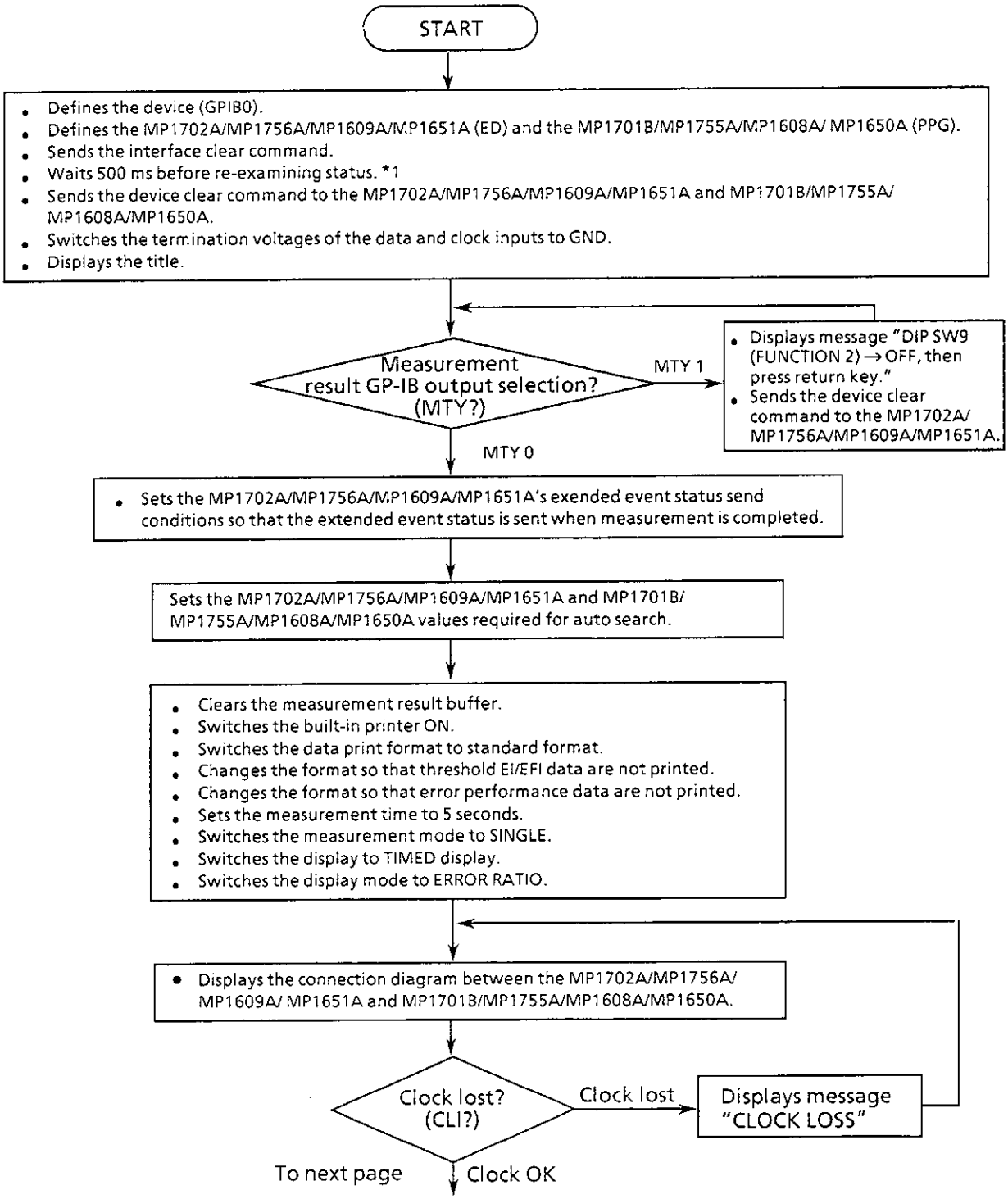
120 Variable "rd$" is initialized with spaces.

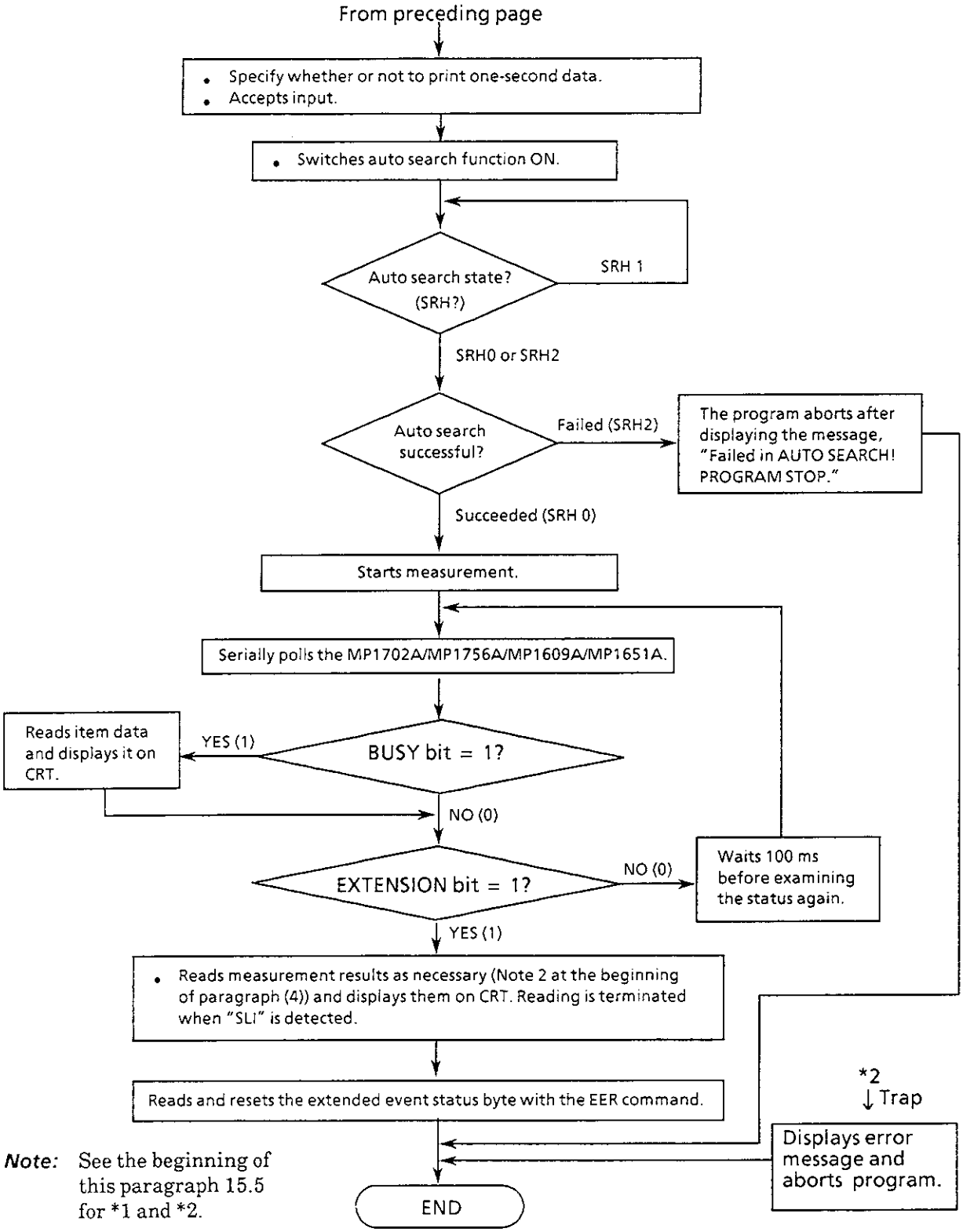
130 The contents read from the talker (MP1702A/MP1756A/MP1609A/MP1651A) by the adapter are stored in rd$.

140 Checks whether the GP-IB commands were executed correctly.

(a) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Standard data are output in standard format at the end of measurement, as determined by serial polling.





Note: See the beginning of this paragraph 15.5 for *1 and *2.

Program listing

```
' ************************************************************
' *                                                          *
' *     MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT     *
' *          -- Standard format , Standard output--          *
' *        -- Measurement end? >> Serial polling --          *
' *                                                ED_M1     *
' ************************************************************
'
'-------------------------------------------------------------'
'                          MAIN  ROUTINE                      '
'-------------------------------------------------------------'
'
common shared IBSTA%,IBERR%,IBCNT%   ' Setup GPIB-PC functions
GOSUB gpinit                         ' Setup GPIB interface
wrt$="DTM 0;CTM 0" : GOSUB wrtcmd2
'
SCREEN 9
PRINT "** MP1702A/MP1609A/MP1651A **"
PRINT "** MEASUREMENT SAMPLE SOFT **"
PRINT
'
GOSUB  MType                         ' Check Measurement type
'
wrt$="EES 64" : GOSUB wrtcmd2        ' EES : MEASURE bit
'
GOSUB DSet                           ' Data setting
GOSUB Mode                           ' Measurement mode
GOSUB Clock                          ' Check Clock loss
GOSUB OneSec                         ' 1 sec dataprint ?
GOSUB Srch                           ' Auto search : ON
'
wrt$="STA"  : GOSUB wrtcmd2          ' Measurement Start
'
GOSUB SPoll                    ' Serial polling--> Extension bit=1
GOSUB Result                   ' Display Result
'
STOP
'
'-------------------------------------------------------------'
'                          SUB ROUTINE                        '
'-------------------------------------------------------------'
'
MType : '---------- Check Measurement type ----------
'
    DO
        wrt$="MTY?" : GOSUB wrtcmd2 : GOSUB readcmd
'
        IF MID$(rd$,1,5)="MTY 1" THEN
          LOCATE 23,1
          INPUT "DIP SW 9 (Function 2)-->OFF, then press Enter key",A
          CALL IBCLR(ED%)
'
        END IF
    LOOP UNTIL MID$(rd$,1,5)="MTY 0"
'
    RETURN
'
DSet: '---------- Data Setting ----------
'
    wrt$= "LGC 0;PTN 2;MRK 1"     : GOSUB wrtcmd1
    wrt$= "LGC 0;PTN 2;MRK 1"     : GOSUB wrtcmd2
```

```
    wrt$= "CLK 1;RES 1;FRQ 3000" : GOSUB wrtcmd1
    wrt$= "OFS 0;SPD 0"          : GOSUB wrtcmd1
    wrt$= "DAP 1;DOS 1"          : GOSUB wrtcmd1
    wrt$= "CDL 100;CAP 1;COS 1"  : GOSUB wrtcmd1
    wrt$= "EAD 1"                : GOSUB wrtcmd1
    '
    wrt$="CPL 0"                 : GOSUB wrtcmd2
    wrt$="SRH 0"                 : GOSUB wrtcmd2
    wrt$="SYN 1"                 : GOSUB wrtcmd2
    '
    RETURN
    '
Mode: '---------- Measurement mode ----------
    '
    wrt$="BCL"    : GOSUB wrtcmd2  ' Measurement buffer clear
    wrt$="PRN 1"  : GOSUB wrtcmd2  ' Printer : ON
    wrt$="FMT 0"  : GOSUB wrtcmd2  ' Data output : Standard format
    wrt$="THR 0"  : GOSUB wrtcmd2  ' Threshold output : OFF
    wrt$="EPF 0"  : GOSUB wrtcmd2  ' Error performance output : OFF
    '
    wrt$="PRD 0,0,0,5" : GOSUB wrtcmd2   ' Measurement period 5sec
    wrt$="MOD 1"  : GOSUB wrtcmd2  ' Measurement mode : SINGLE
    wrt$="TIM 3"  : GOSUB wrtcmd2  ' Display TIMED
    wrt$="DMS 0"  : GOSUB wrtcmd2  ' Display ERROR RATIO
    '
    RETURN
    '
Clock: '---------- Set data , Check connection ----------
    '
    DO
        GOSUB Connect              ' Display device connection
        '
        wrt$="CLI?" : GOSUB wrtcmd2 : GOSUB readcmd
        '
        IF MID$(rd$,1,5)="CLI 1" THEN PRINT "** CLOCK LOSS **"
        '
    LOOP UNTIL MID$(rd$,1,5)="CLI 0"
    '
    RETURN
    '
OneSec: '---------- One second data output? ----------
    '
    INPUT "1 sec data output [ OFF=0, ON=1 ]";osc$
    wrt$="OSC "+osc$ : GOSUB wrtcmd2
    '
    RETURN
    '
Srch: '---------- Auto search ON ----------
    '
    wrt$="SRH 1" : GOSUB wrtcmd2             ' Auto search : ON
    '
    DO
        wrt$="SRH?" : GOSUB wrtcmd2 : GOSUB readcmd
    LOOP UNTIL MID$(rd$,1,5)="SRH 0" or MID$(rd$,1,5)="SRH 2"
    '
    IF MID$(rd$,1,5)="SRH 2"  THEN
        PRINT "<<< Failed in AUTO SEARCH ! >>>"
        PRINT "    ** PROGRAM STOP  **"
        STOP
    END IF
    '
    RETURN
```

```
SPoll:  '---------- Serial polling--> Measurement END? ----------
        '
    DO
        CALL IBRSP(ED%,SPR%)                        ' Serial polling
        IF IBSTA% < 0 THEN  GOTO trap
        '
        s=SPR% AND &H10
        q=SPR% AND &H2
        '
        IF s=&H10 THEN GOSUB readcmd : PRINT rd$    ' BUSY bit=1
        '
        tim = 0.1 : GOSUB waidly
    LOOP UNTIL q=&H2                                ' EXTENSION bit=1
    '
    RETURN
        '
Result: '---------- Display Result ----------
        '
    DO
        GOSUB readcmd : PRINT rd$
        '
    LOOP UNTIL MID$(rd$,1,3)="SLI"
    '
    wrt$="EER?" : GOSUB wrtcmd2          ' RESET Extension byte
    GOSUB readcmd
        '
    RETURN
        '
gpinit: ' ---------- Set up GP-IB functions  ----------
        '
    CALL IBFIND("GPIB0", GPIB0%)          ' Open device (GPIB0)
    IF GPIB0% < 0 THEN GOTO trap          ' system error
    '
    CALL IBFIND("PPG", PPG%)              ' Open device(PPG)
    IF PPG% < 0 THEN GOTO trap            ' system error
    '
    CALL IBFIND("ED", ED%)                ' Open device(ED)
    IF ED% < 0 THEN GOTO trap             ' system error
    '
    CALL IBSIC(GPIB0%)                    ' Interface clear
    IF IBSTA% < 0 THEN GOTO trap          ' system error
    '
    tim = 0.5
    GOSUB waidly
    '
    CALL IBCLR(PPG%)                      ' Device clear(PPG)
    '
    CALL IBCLR(ED%)                       ' Device clear(ED)
    '
    RETURN
        '
wrtcmd1: ' ----------  Write command (PPG) ----------
        '
    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(PPG%, wrt$)                ' Write command (PPG)
    IF IBSTA% < 0 THEN GOTO trap          ' Trap
    '
    RETURN
        '
```

```
wrtcmd2:  ' ----------  Write command (ED) ----------
          '
     wrt$=wrt$+chr$(13)+chr$(10)
     CALL IBWRT(ED%, wrt$)                  ' Write command (ED)
     IF IBSTA% < 0 THEN GOTO trap           ' Trap
          '
     RETURN
          '
readcmd:  ' ----------  Read command (ED) ----------
          '
     CMD$="?"+chr$(42)+"A"
     CALL IBCMD(GPIB0%,CMD$)
          '
     rd$=SPACE$(28)
     CALL IBRD(GPIB0%, rd$)                 ' Read command (GPIB0)
     IF IBSTA% < 0 THEN GOTO trap           ' Trap
          '
     RETURN
          '
waidly:   ' ----------  Wait delay  ----------
          '
     stm = TIMER
     etm = TIMER
     WHILE etm - stm < tim
         etm = TIMER
         IF etm < stm THEN etm = etm + 86400
     WEND
          '
     RETURN
          '
trap:     ' ----------  System trap  ----------
          '
     PRINT "IBERR%:" + STR$(IBERR%)
      STOP
          '
Connect:  '---------- display Connection ----------
          '
     WINDOW (-600,-500) -(600,500)
          '
     LINE (-50,-50) - (600,500),14,B
     LINE (10,100) - STEP(235,200),,B
     LINE (300,100) - STEP(235,200),,B
          '
     CIRCLE (130,130),8
     LINE (130,125) - STEP(0,-55)
     LINE (130,70) - STEP(220,0)
     LINE (350,70) - STEP(0,55)
     CIRCLE (350,130),8
          '
     CIRCLE (190,130),8
     LINE (190,125) - STEP(0,-100)
     LINE (190,25) - STEP(220,0)
     LINE (410,25) - STEP(0,100)
     CIRCLE (410,130),8
          '
     LOCATE 2,50: PRINT "<<  CONNECTION  >>"
          '
     LOCATE 4,42: PRINT "MP1701B/MP1608A    MP1702A/MP1609A"
     LOCATE 5,42: PRINT "    /MP1650A           /MP1651A"
          '
     LOCATE 9,45: PRINT "DATA  CLOCK1"
          '
     LOCATE 9,62: PRINT "DATA  CLOCK"
          '
```

```
LOCATE 23,1
INPUT "    Aer you ready ?  Press return key to start     ",A
'
LOCATE 23,1
PRINT "                                                    "
'
LOCATE 4,1
'
RETURN
'
END
```

EXECUTED RESULT

① When one-second data is not specified for printing

Output to GP-IB

```
STA 90,02,28,15,30,45
STA 90,02,28,15,30,45
END 90,02,28,15,30,50
ELP      0,00,00,05
ER  1.0000E-04
EC      2500006
EI            5
EFI      0.0000
PFI           0
CLI           0
SLI           0
```

Printout from built-in printer

```
90-02-28/15:30:45 START
**************************
 START 90-02-28/15:30:45
 END   90-02-28/15:30:50
 ELP           0/00:00:05
 ==ERROR MEASUREMENT==
 ERROR RATIO 1.0000E-04
 ERROR COUNT    2500006
 EI                   5
 EFI             0.0000%
 PFI                  0
 CLI                  0
 SLI                  0
**************************
```

② When one-second data is specified for printing

Output to GP-IB

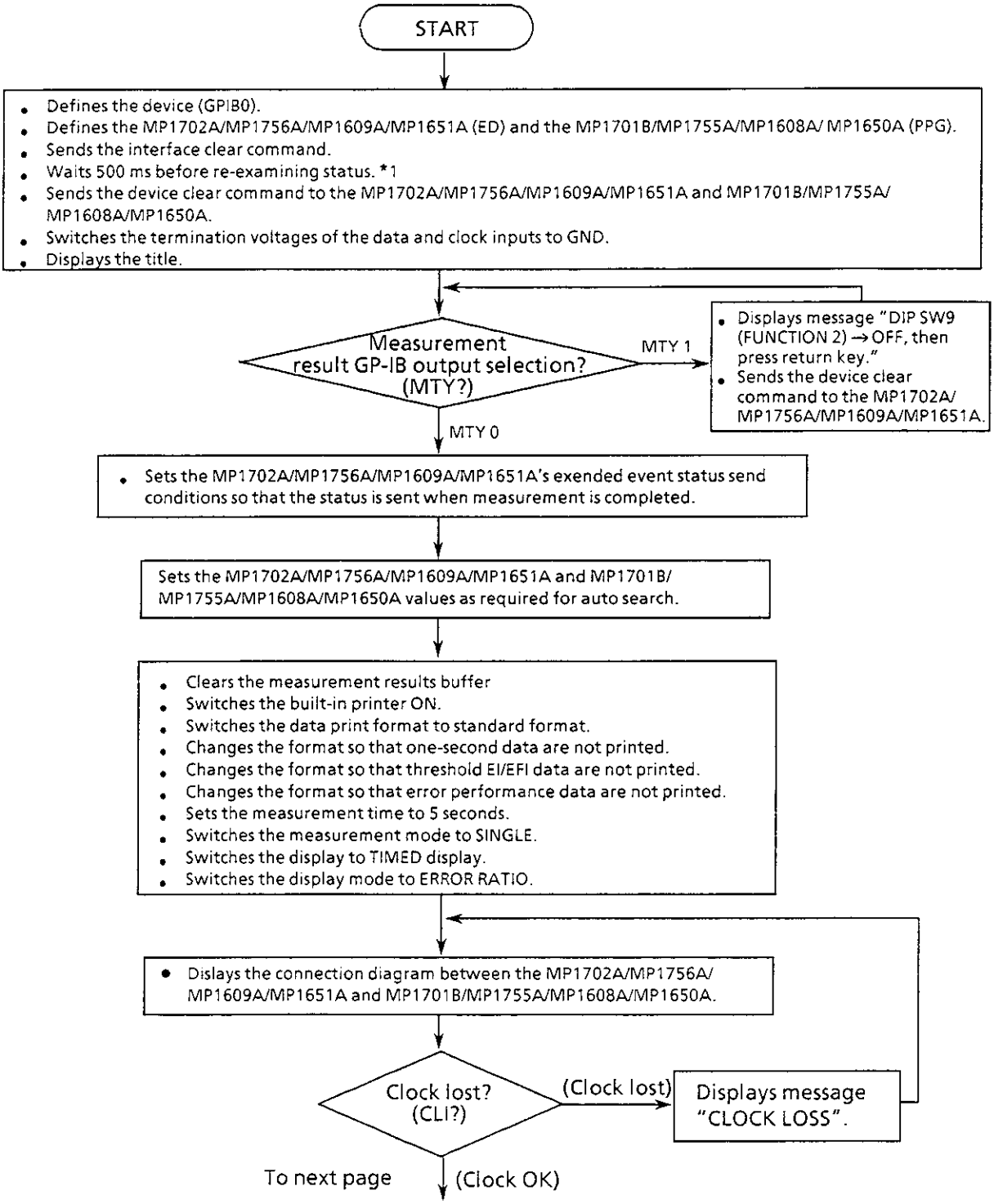
```
STA 90,02,28,15,31,27
OSD 1.0000E-04,     500002
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500002
STA 90,02,28,15,31,27
END 90,02,28,15,31,32
ELP      0,00,00,05
ER  1.0000E-04
EC      2500007
EI            5
EFI      0.0000
PFI           0
CLI           0
SLI           0
```

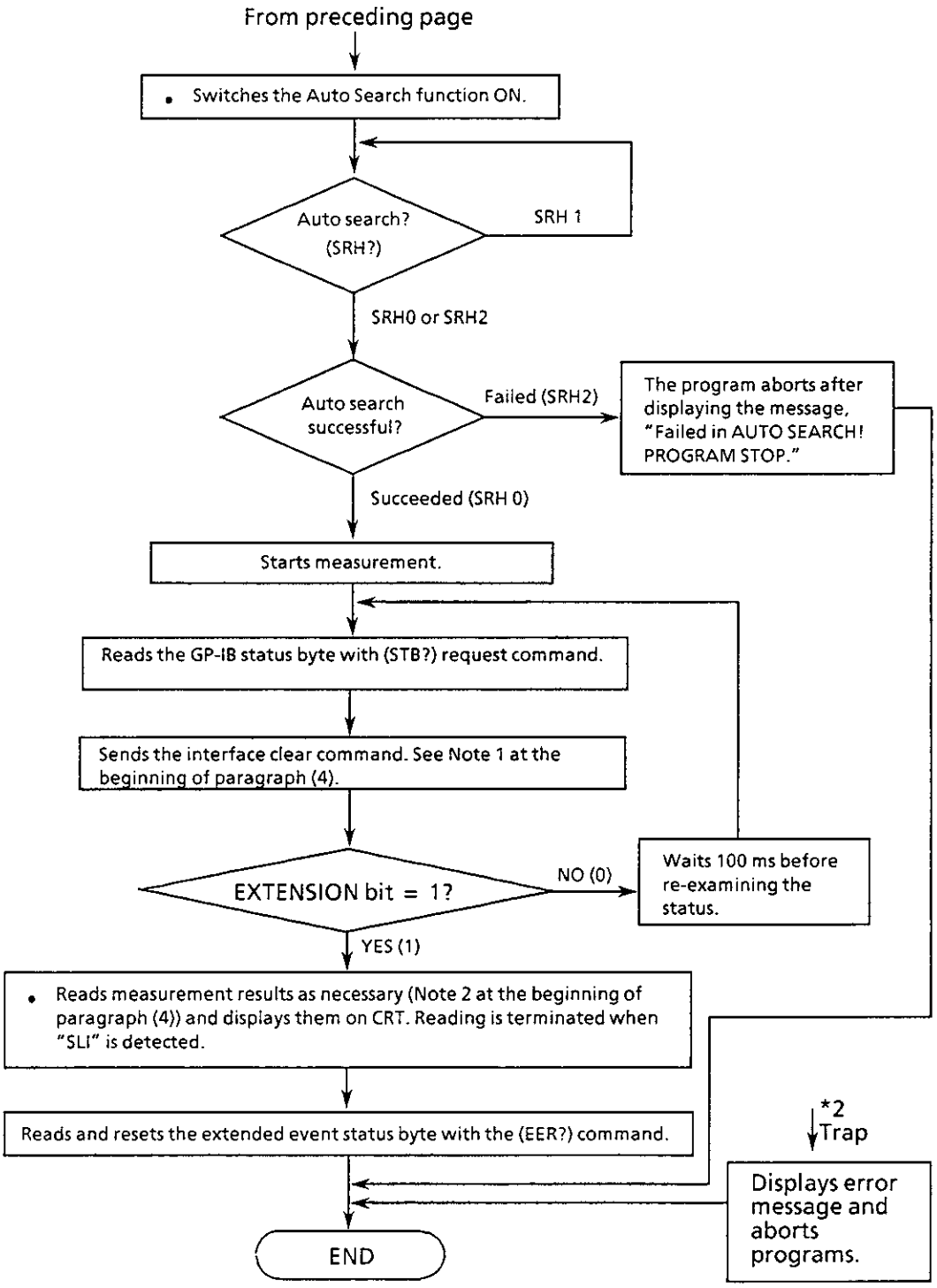
Printout from built-in printer

```
90-02-28/15:31:27 START
31:28 1.00E-04    500002
31:29 1.00E-04    500001
31:30 1.00E-04    500001
31:31 1.00E-04    500001
31:32 1.00E-04    500002
**************************
 START 90-02-28/15:31:27
 END   90-02-28/15:31:32
 ELP           0/00:00:05
 ==ERROR MEASUREMENT==
 ERROR RATIO 1.0000E-04
 ERROR COUNT    2500007
 EI                   5
 EFI             0.0000%
 PFI                  0
 CLI                  0
 SLI                  0
**************************
```

(b) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Standard data is output in the standard format at the end of measurement, as determined by the (STB?) request command.





Note: See the beginning of this paragraph 15.5 for *1 and *2.

Program listing

```
' ************************************************************
' *                                                          *
' *     MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT     *
' *          -- Standard format , Standard output --         *
' *     -- Measurement end? >> Request command STB? --       *
' *                                                ED_M2     *
' ************************************************************
'
'-----------------------------------------------------------'
'                          MAIN  ROUTINE                    '
'-----------------------------------------------------------'
'
common shared IBSTA%,IBERR%,IBCNT%   ' Setup GPIB-PC functions
GOSUB gpinit                         ' Setup GPIB interface
wrt$="DTM 0;CTM 0" : GOSUB wrtcmd2
'
SCREEN 9
PRINT "** MP1702A/MP1609A/MP1651A **"
PRINT "** MEASUREMENT SAMPLE SOFT **"
PRINT
'
GOSUB  MType                         ' Check Measurement type
'
wrt$="EES 64" : GOSUB wrtcmd2        ' EES : MEASURE bit
'
GOSUB DSet                           ' Data setting
GOSUB Mode                           ' Measurement mode
GOSUB Clock                          ' Check Clock loss ?
GOSUB Srch                           ' Auto search : ON
'
wrt$="STA"  : GOSUB wrtcmd2          ' Measurement Start
'
GOSUB StByte                  ' Request STB?--> Extension bit=1
GOSUB Result                  ' Display Result
'
STOP
'
'-----------------------------------------------------------'
'                          SUB ROUTINE                      '
'-----------------------------------------------------------'
'
MType : '---------- Check Measurement type ----------
'
  DO
    wrt$="MTY?" : GOSUB wrtcmd2 : GOSUB readcmd
    '
    IF MID$(rd$,1,5)="MTY 1" THEN
      LOCATE 23,1
      INPUT "DIP SW 9 (Function 2)-->OFF, then press Enter key",A
      CALL IBCLR(ED%)
      '
    END IF
  LOOP UNTIL MID$(rd$,1,5)="MTY 0"
  '
  RETURN
    '
```

```
DSet:  '---------- Data Setting ----------

     wrt$= "LGC 0;PTN 2;MRK 1"     : GOSUB wrtcmd1
     wrt$= "LGC 0;PTN 2;MRK 1"     : GOSUB wrtcmd2
     '
     wrt$= "CLK 1;RES 1;FRQ 3000" : GOSUB wrtcmd1
     wrt$= "OFS 0;SPD 0"           : GOSUB wrtcmd1
     wrt$= "DAP 1;DOS 1"           : GOSUB wrtcmd1
     wrt$= "CDL 100;CAP 1;COS 1"   : GOSUB wrtcmd1
     wrt$= "EAD 1"                 : GOSUB wrtcmd1
     '
     wrt$="CPL 0"                  : GOSUB wrtcmd2
     wrt$="SRH 0"                  : GOSUB wrtcmd2
     wrt$="SYN 1"                  : GOSUB wrtcmd2
     '
     RETURN
     '
Mode:  '---------- Measurement mode ----------
     '
     wrt$="BCL"    : GOSUB wrtcmd2   ' Measurement buffer clear
     wrt$="PRN 1" : GOSUB wrtcmd2   ' Printer : ON
     wrt$="FMT 0" : GOSUB wrtcmd2   ' Data output : Standard format
     wrt$="OSC 0" : GOSUB wrtcmd2   ' 1 sec data output : OFF
     wrt$="THR 0" : GOSUB wrtcmd2   ' Threshold output : OFF
     wrt$="EPF 0" : GOSUB wrtcmd2   ' Error performance output : OFF
     '
     wrt$="PRD 0,0,0,5" : GOSUB wrtcmd2    ' Measurement period 5sec
     wrt$="MOD 1" : GOSUB wrtcmd2   ' Measurement mode : SINGLE
     wrt$="TIM 3" : GOSUB wrtcmd2   ' Display TIMED
     wrt$="DMS 0" : GOSUB wrtcmd2   ' Display ERROR RATIO
     '
     RETURN
     '
Clock:  '---------- Set data , Check connection ----------
     '
     DO
          GOSUB Connect              ' Display device connection
          '
          wrt$="CLI?" : GOSUB wrtcmd2 : GOSUB readcmd
          '
          IF MID$(rd$,1,5)="CLI 1" THEN PRINT "** CLOCK LOSS **"
          '
     LOOP UNTIL MID$(rd$,1,5)="CLI 0"
     '
     RETURN
     '
Srch:  '---------- Auto search ON ----------
     '
     wrt$="SRH 1" : GOSUB wrtcmd2       ' Set Auto search : ON
     '
     DO
          wrt$="SRH?" : GOSUB wrtcmd2 : GOSUB readcmd
     LOOP UNTIL MID$(rd$,1,5)="SRH 0" or MID$(rd$,1,5)="SRH 2"
     '
     IF MID$(rd$,1,5)="SRH 2"  THEN
          PRINT "<<< Failed in AUTO SEARCH ! >>>"
          PRINT "     ** PROGRAM STOP  **"
          STOP
     END IF
     '
     RETURN
     '
```

```
StByte: '---------- Request STB? --> Measurement END? ----------
        '
    DO
        wrt$="STB?" : GOSUB wrtcmd2 : GOSUB readcmd
        '
        CALL IBSIC(GPIB0%)                      ' Interface clear
        IF IBSTA% < 0 THEN GOTO trap            ' system error
        '
        tim = 0.1 : GOSUB waidly
        '
    LOOP UNTIL MID$(rd$,13,1)="1"     ' Extension bit = 1
    '
    RETURN
        '
Result: '---------- Display Result ----------
        '
    DO
        GOSUB readcmd : PRINT rd$
        '
    LOOP UNTIL MID$(rd$,1,3)="SLI"
    '
    wrt$="EER?" : GOSUB wrtcmd2          ' RESET Extension byte
    GOSUB readcmd
        '
    RETURN
        '
gpinit: ' ---------- Set up GP-IB functions  ----------
        '
    CALL IBFIND("GPIB0", GPIB0%)            ' Open device (GPIB0)
    IF GPIB0% < 0 THEN GOTO trap            ' system error
    '
    CALL IBFIND("PPG", PPG%)                ' Open device(PPG)
    IF PPG% < 0 THEN GOTO trap              ' system error
    '
    CALL IBFIND("ED", ED%)                  ' Open device(ED)
    IF ED% < 0 THEN GOTO trap               ' system error
    '
    CALL IBSIC(GPIB0%)                      ' Interface clear
    IF IBSTA% < 0 THEN GOTO trap            ' system error
    '
    tim = 0.5
    GOSUB waidly
    '
    CALL IBCLR(PPG%)                        ' Device clear(PPG)
    '
    CALL IBCLR(ED%)                         ' Device clear(ED)
    '
    RETURN
        '
wrtcmd1: ' ----------  Write command (PPG) ----------
        '
    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(PPG%, wrt$)                  ' Write command (PPG)
    IF IBSTA% < 0 THEN GOTO trap            ' Trap
    '
    RETURN
        '
wrtcmd2: ' ----------  Write command (ED) ----------
        '
    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(ED%, wrt$)                   ' Write command (ED)
    IF IBSTA% < 0 THEN GOTO trap            ' Trap
    '
    RETURN
        '
```

```
readcmd:  ' ---------- Read command (ED) ----------
          '
     CMD$="?"+chr$(42)+"A"
     CALL IBCMD(GPIB0%,CMD$)
     '
     rd$=SPACE$(23)
     CALL IBRD(GPIB0%, rd$)              ' Read command (GPIB0)
     IF IBSTA% < 0 THEN GOTO trap        ' Trap
     '
     RETURN
          '
waidly:  ' ----------  Wait delay  ----------
          '
     stm = TIMER
     etm = TIMER
     WHILE etm - stm < tim
         etm = TIMER
         IF etm < stm THEN etm = etm + 86400
     WEND
     '
     RETURN
          '
trap:    ' ----------  System trap  ----------
          '
     PRINT "IBERR%:" + STR$(IBERR%)
      STOP
          '
Connect: '---------- display Connection ----------
          '
     WINDOW (-600,-500) -(600,500)
     '
     LINE (-50,-50) - (600,500),14,B
     LINE (10,100) - STEP(235,200),,B
     LINE (300,100) - STEP(235,200),,B
     '
     CIRCLE (130,130),8
     LINE (130,125) - STEP(0,-55)
     LINE (130,70) - STEP(220,0)
     LINE (350,70) - STEP(0,55)
     CIRCLE (350,130),8
     '
     CIRCLE (190,130),8
     LINE (190,125) - STEP(0,-100)
     LINE (190,25) - STEP(220,0)
     LINE (410,25) - STEP(0,100)
     CIRCLE (410,130),8
     '
     LOCATE 2,50: PRINT "<<  CONNECTION  >>"
     '
     LOCATE 4,42: PRINT "MP1701B/MP1608A    MP1702A/MP1609A"
     LOCATE 5,42: PRINT "    /MP1650A           /MP1651A"
     '
     LOCATE 9,45: PRINT "DATA  CLOCK1"
     LOCATE 9,62: PRINT "DATA  CLOCK"
     '
     LOCATE 23,1
     INPUT "   Aer you ready ?  Press return key to start    ",A
     '
     LOCATE 23,1
     PRINT "                                                  "
     '
     LOCATE 4,1
     '
     RETURN
     '
     END
```

EXECUTED RESULT

Output to GP-IB

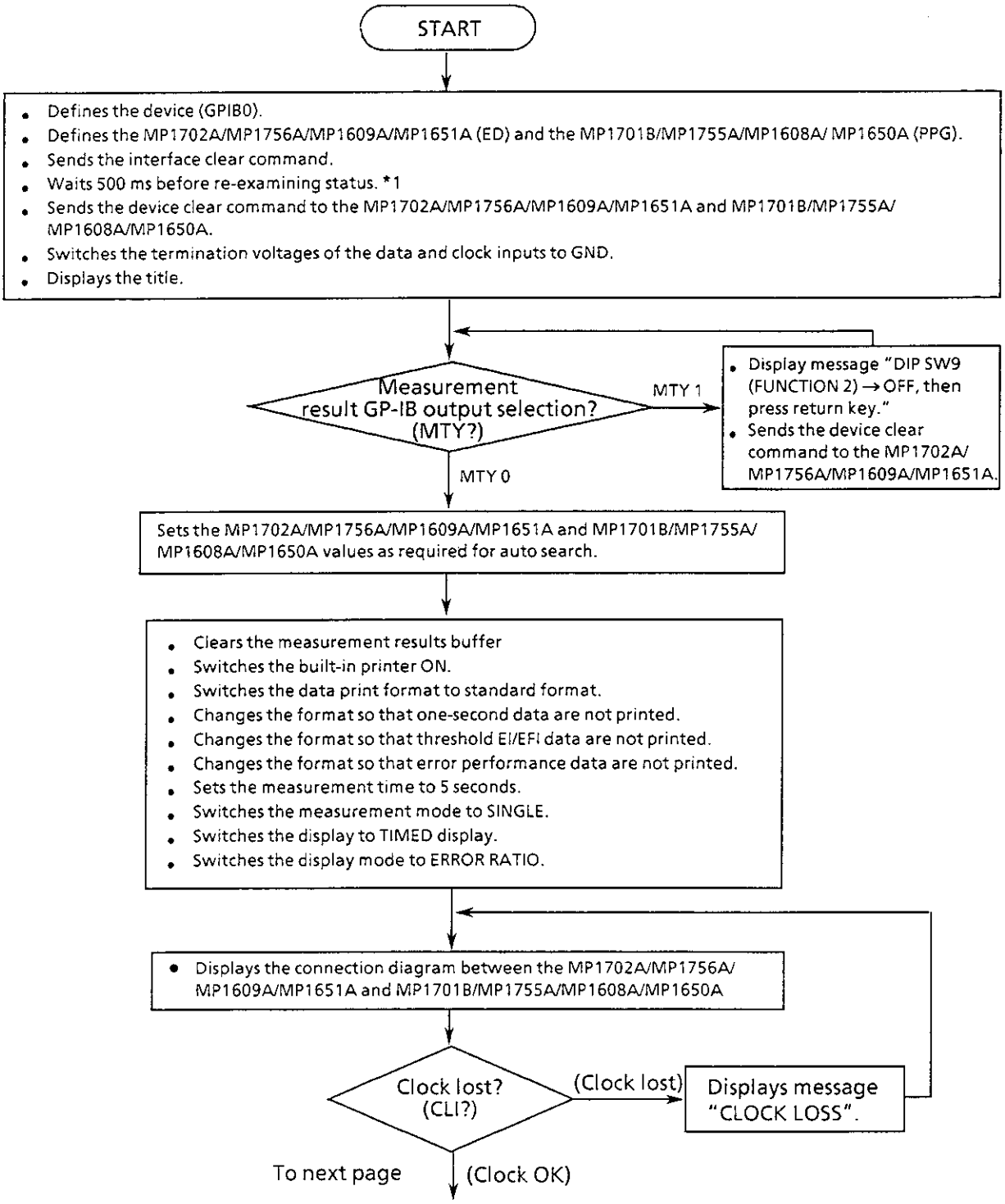
```
STA 90,02,28,15,34,28

STA 90,02,28,15,34,28

END 90,02,28,15,34,33

ELP      0,00,00,05

ER  1.0000E-04

EC      2500006

EI            5

EFI      0.0000

PFI           0

CLI           0

SLI           0
```

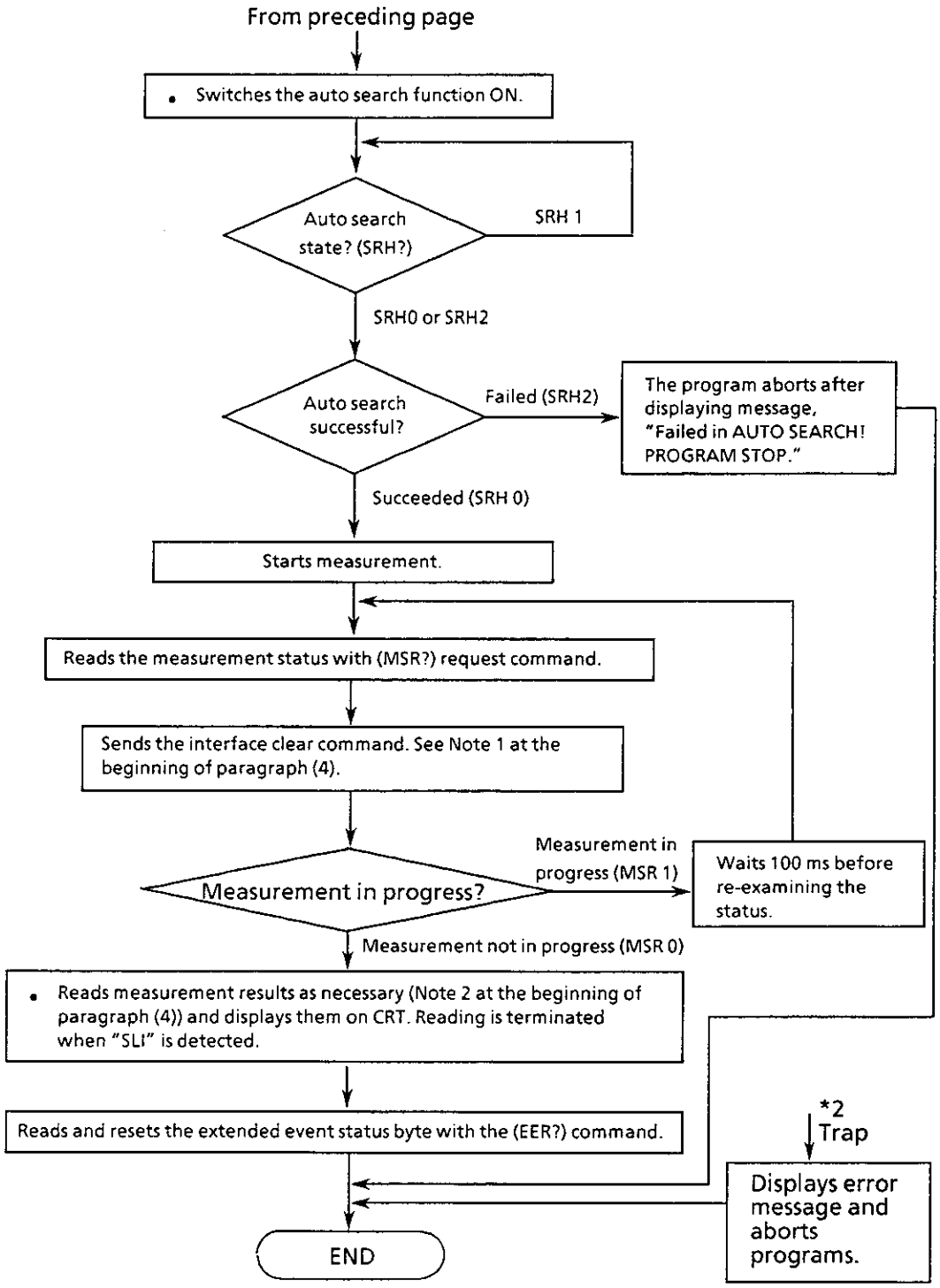
Printout from built-in printer

```
90-02-28/15:34:28 START

**************************
 START 90-02-28/15:34:28
 END   90-02-28/15:34:33
 ELP           0/00:00:05
 ==ERROR MEASUREMENT==
 ERROR RATIO 1.0000E-04
 ERROR COUNT    2500006
 EI                   5
 EFI             0.0000%
 PFI                  0
 CLI                  0
 SLI                  0
**************************
```

(c) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Standard data are output in standard format at the end of measurement, as determined by the (MSR?) request command.





Note: See the beginning of this paragraph 15.5 for *1 and *2.

Program listing

```
' ************************************************************
' *                                                          *
' *     MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT     *
' *          -- Standard format , Standard output --         *
' *      -- Measurement end? >> Request command MSR? --      *
' *                                                  ED_M3   *
' ************************************************************
'
'-------------------------------------------------------------'
'                          MAIN  ROUTINE                      '
'-------------------------------------------------------------'
'
common shared IBSTA%,IBERR%,IBCNT%   ' Setup GPIB-PC functions
GOSUB gpinit                         ' Setup GPIB interface
wrt$="DTM 0;CTM 0" : GOSUB wrtcmd2
'
SCREEN 9
'
PRINT "** MP1702A/MP1609A/MP1651A **"
PRINT "** MEASUREMENT SAMPLE SOFT **"
PRINT
'
GOSUB  MType                         ' Check Measurement type
'
GOSUB DSet                           ' Data setting
GOSUB Mode                           ' Measurement mode
GOSUB Clock                          ' Check Clock loss ?
GOSUB Srch                           ' Auto search : ON
'
wrt$="STA"  : GOSUB wrtcmd2          ' Measurement Start
'
GOSUB Msr                      ' Request MSR?--> Measurement stop?
GOSUB Result                   ' Display Result
'
STOP
'
'-------------------------------------------------------------'
'                          SUB ROUTINE                        '
'-------------------------------------------------------------'
    '
MType : '---------- Check Measurement type ----------
    '
    DO
      wrt$="MTY?" : GOSUB wrtcmd2 : GOSUB readcmd
      '
      IF MID$(rd$,1,5)="MTY 1" THEN
        LOCATE 23,1
        INPUT "DIP SW 9 (Function 2)-->OFF, then press Enter key",A
        CALL IBCLR(ED%)
        '
      END IF
    LOOP UNTIL MID$(rd$,1,5)="MTY 0"
    '
    RETURN
    '
DSet: '---------- Data Setting ----------
    '
    wrt$= "LGC 0;PTN 2;MRK 1"     : GOSUB wrtcmd1
    wrt$= "LGC 0;PTN 2;MRK 1"     : GOSUB wrtcmd2
```

```
'
    wrt$= "CLK 1;RES 1;FRQ 3000" : GOSUB wrtcmd1
    wrt$= "OFS 0;SPD 0"          : GOSUB wrtcmd1
    wrt$= "DAP 1;DOS 1"          : GOSUB wrtcmd1
    wrt$= "CDL 100;CAP 1;COS 1"  : GOSUB wrtcmd1
    wrt$= "EAD 1"                : GOSUB wrtcmd1
    '
    wrt$="CPL 0"                 : GOSUB wrtcmd2
    wrt$="SRH 0"                 : GOSUB wrtcmd2
    wrt$="SYN 1"                 : GOSUB wrtcmd2
    '
    RETURN
    '
Mode: '---------- Measurement mode ----------
    '
    wrt$="BCL"   : GOSUB wrtcmd2  ' Measurement buffer clear
    wrt$="PRN 1" : GOSUB wrtcmd2  ' Printer : ON
    wrt$="FMT 0" : GOSUB wrtcmd2  ' Data output : Standard format
    wrt$="OSC 0" : GOSUB wrtcmd2  ' 1 sec data output : OFF
    wrt$="THR 0" : GOSUB wrtcmd2  ' Threshold output : OFF
    wrt$="EPF 0" : GOSUB wrtcmd2  ' Error performance output : OFF
    '
    wrt$="PRD 0,0,0,5" : GOSUB wrtcmd2   ' Measurement period 5sec
    wrt$="MOD 1" : GOSUB wrtcmd2  ' Measurement mode : SINGLE
    wrt$="TIM 3" : GOSUB wrtcmd2  ' Display TIMED
    wrt$="DMS 0" : GOSUB wrtcmd2  ' Display ERROR RATIO
    '
    RETURN
    '
Clock: '---------- Set data , Check connection ----------
    '
    DO
        GOSUB Connect              ' Display device connection
        '
        wrt$="CLI?" : GOSUB wrtcmd2 : GOSUB readcmd
        '
        IF MID$(rd$,1,5)="CLI 1" THEN PRINT "** CLOCK LOSS **"
        '
    LOOP UNTIL MID$(rd$,1,5)="CLI 0"
    '
    RETURN
    '
Srch: '---------- Auto search ON ----------
    '
    wrt$="SRH 1" : GOSUB wrtcmd2      ' Set Auto search : ON
    '
    DO
        wrt$="SRH?" : GOSUB wrtcmd2 : GOSUB readcmd
    LOOP UNTIL MID$(rd$,1,5)="SRH 0" or MID$(rd$,1,5)="SRH 2"
    '
    IF MID$(rd$,1,5)="SRH 2"  THEN
        PRINT "<<< Failed in AUTO SEARCH ! >>>"
        PRINT "    ** PROGRAM STOP  **"
        STOP
    END IF
    '
    RETURN
    '
Msr: '---------- Request command MSR?-->Measurement stop? ----------
    '
    DO
        wrt$="MSR?" : GOSUB wrtcmd2 : GOSUB readcmd
        '
        CALL IBSIC(GPIB0%)                    ' Interface clear
```

```
        IF IBSTA% < 0 THEN GOTO trap          ' system error
        tim = 0.1    : GOSUB waidly
        '
    LOOP UNTIL MID$(rd$,1,5)="MSR 0"          ' Measurement stop
    '
    RETURN
    '
Result: '---------- Display Result ----------
    '
    DO
      GOSUB readcmd : PRINT rd$
    '
    LOOP UNTIL MID$(rd$,1,3)="SLI"
    '
    wrt$="EER?" : GOSUB wrtcmd2           ' RESET Extension byte
    GOSUB readcmd
    '
    RETURN
    '
gpinit: ' ---------- Set up GP-IB functions  ----------
    '
    CALL IBFIND("GPIB0", GPIB0%)          ' Open device (GPIB0)
    IF GPIB0% < 0 THEN GOTO trap          ' system error
    '
    CALL IBFIND("PPG", PPG%)              ' Open device(PPG)
    IF PPG% < 0 THEN GOTO trap            ' system error
    '
    CALL IBFIND("ED", ED%)                ' Open device(ED)
    IF ED% < 0 THEN GOTO trap             ' system error
    '
    CALL IBSIC(GPIB0%)                    ' Interface clear
    IF IBSTA% < 0 THEN GOTO trap          ' system error
    '
    tim = 0.5
    GOSUB waidly
    '
    CALL IBCLR(PPG%)                      ' Device clear(PPG)
    '
    CALL IBCLR(ED%)                       ' Device clear(ED)
    '
    RETURN
    '
wrtcmd1: ' ----------  Write command (PPG) ----------
    '
    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(PPG%, wrt$)                ' Write command (PPG)
    IF IBSTA% < 0 THEN GOTO trap          ' Trap
    '
    RETURN
    '
wrtcmd2: ' ----------  Write command (ED) ----------
    '
    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(ED%, wrt$)                 ' Write command (ED)
    IF IBSTA% < 0 THEN GOTO trap          ' Trap
    '
    RETURN
    '
readcmd: ' ---------- Read command (ED) ----------
    '
    CMD$="?"+chr$(42)+"A"
    CALL IBCMD(GPIB0%,CMD$)
    '
    rd$=SPACE$(23)
    CALL IBRD(GPIB0%, rd$)                ' Read command (GPIB0)
```

```
    IF IBSTA% < 0 THEN GOTO trap          ' Trap
    '
    RETURN
        '
waidly: ' ----------  Wait delay  ----------
        '
    stm = TIMER
    etm = TIMER
    WHILE etm - stm < tim
        etm = TIMER
        IF etm < stm THEN etm = etm + 86400
    WEND
    '
    RETURN
        '
trap:   ' ----------  System trap  ----------
        '
    PRINT "IBERR%:" + STR$(IBERR%)
     STOP
        '
Connect: '---------- display Connection ----------
        '
    WINDOW (-600,-500) -(600,500)
    '
    LINE (-50,-50) - (600,500),14,B
    LINE (10,100) - STEP(235,200),,B
    LINE (300,100) - STEP(235,200),,B
    '
    CIRCLE (130,130),8
    LINE (130,125) - STEP(0,-55)
    LINE (130,70) - STEP(220,0)
    LINE (350,70) - STEP(0,55)
    CIRCLE (350,130),8
    '
    CIRCLE (190,130),8
    LINE (190,125) - STEP(0,-100)
    LINE (190,25) - STEP(220,0)
    LINE (410,25) - STEP(0,100)
    CIRCLE (410,130),8
    '
    LOCATE 2,50: PRINT "<<  CONNECTION  >>"
    '
    LOCATE 4,42: PRINT "MP1701B/MP1608A     MP1702A/MP1609A"
    LOCATE 5,42: PRINT "     /MP1650A            /MP1651A"
    '
    LOCATE 9,45: PRINT "DATA  CLOCK1"
    LOCATE 9,62: PRINT "DATA  CLOCK"
    '
    LOCATE 23,1
    INPUT "   Aer you ready ?  Press return key to start      ",A
    '
    LOCATE 23,1
    PRINT "                                                    "
    '
    LOCATE 4,1
    '
    RETURN
    '
    END
```

EXECUTED RESULT

Output to GP-IB

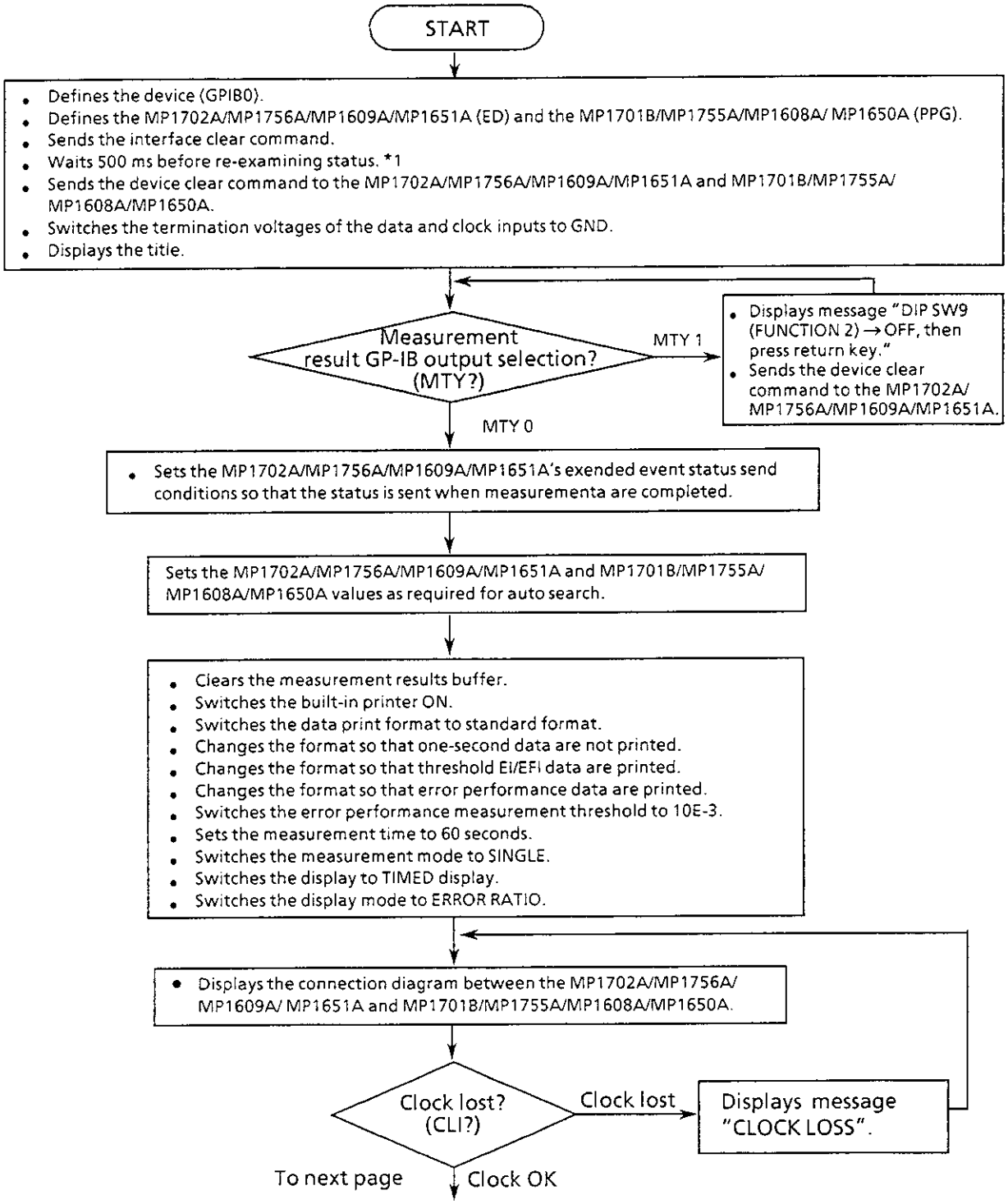
```
STA 90,02,28,15,37,07

STA 90,02,28,15,37,07

END 90,02,28,15,37,12

ELP      0,00,00,05

ER  1.0000E-04

EC      2500006

EI            5

EFI      0.0000

PFI           0

CLI           0

SLI           0
```

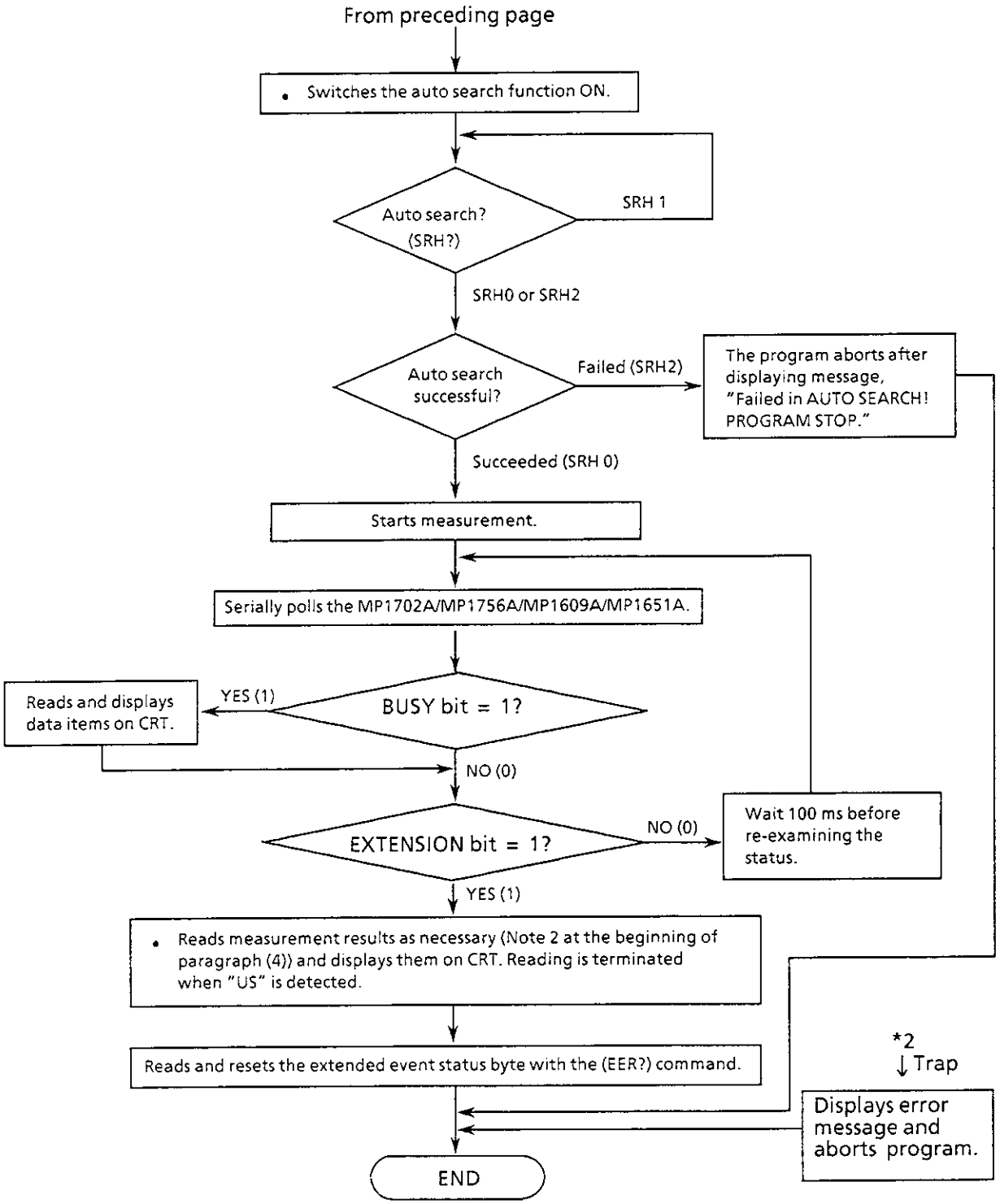
Printout from built-in printer

```
90-02-28/15:37:07 START

**************************
 START 90-02-28/15:37:07
 END   90-02-28/15:37:12
 ELP           0/00:00:05
 ==ERROR MEASUREMENT==
 ERROR RATIO 1.0000E-04
 ERROR COUNT    2500006
 EI                   5
 EFI             0.0000%
 PFI                  0
 CLI                  0
 SLI                  0
**************************
```

(d) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Extended data (threshold measurement and error performance measurement data) are output in standard format at the end of measurement, as determined by serial polling.





Note: See the beginning of this paragraph 15.5 for *1 and *2.

Program listing

```
' ************************************************************
' *                                                          *
' *     MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT     *
' *        -- Standard format , Extension output --          *
' *       -- Threshold EI/EFI , Error performance --         *
' *                                                  ED_M4   *
' ************************************************************
'
'-------------------------------------------------------------'
'                          MAIN  ROUTINE                      '
'-------------------------------------------------------------'
'
common shared IBSTA%,IBERR%,IBCNT%   ' Setup GPIB-PC functions
GOSUB gpinit                         ' Setup GPIB interface
wrt$="DTM 0;CTM 0" : GOSUB wrtcmd2
'
SCREEN 9
PRINT "** MP1702A/MP1609A/MP1651A **"
PRINT "** MEASUREMENT SAMPLE SOFT **"
PRINT
'
GOSUB  MType                         ' Check Measurement type
'
wrt$="EES 64" : GOSUB wrtcmd2        ' EES : MEASURE bit
'
GOSUB DSet                           ' Data setting
GOSUB Mode                           ' Measurement mode
GOSUB Clock                          ' Check Clock loss ?
GOSUB Srch                           ' Auto search : ON
'
wrt$="STA"  : GOSUB wrtcmd2          ' Measurement Start
'
GOSUB SPoll                    ' Serial polling--> Extension bit=1
GOSUB Result                   ' Display Result
'
STOP
'
'-------------------------------------------------------------'
'                          SUB ROUTINE                        '
'-------------------------------------------------------------'
      '
MType :  '---------- Check Measurement type ----------
      '
    DO
        wrt$="MTY?" : GOSUB wrtcmd2 : GOSUB readcmd
        '
        IF MID$(rd$,1,5)="MTY 1" THEN
          LOCATE 23,1
          INPUT "DIP SW 9 (Function 2)-->OFF, then press Enter key",A
          CALL IBCLR(ED%)
          '
        END IF
    LOOP UNTIL MID$(rd$,1,5)="MTY 0"
    '
    RETURN
      '
DSet:  '---------- Data Setting ----------
      '
    wrt$= "LGC 0;PTN 2;MRK 1"     : GOSUB wrtcmd1
    wrt$= "LGC 0;PTN 2;MRK 1"     : GOSUB wrtcmd2
    '
```

```
    wrt$= "CLK 1;RES 1;FRQ 3000" : GOSUB wrtcmd1
    wrt$= "OFS 0;SPD 0"          : GOSUB wrtcmd1
    wrt$= "DAP 1;DOS 1"          : GOSUB wrtcmd1
    wrt$= "CDL 100;CAP 1;COS 1"  : GOSUB wrtcmd1
    wrt$= "EAD 1"                : GOSUB wrtcmd1
    '
    wrt$="CPL 0"                 : GOSUB wrtcmd2
    wrt$="SRH 0"                 : GOSUB wrtcmd2
    wrt$="SYN 1"                 : GOSUB wrtcmd2
    '
    RETURN
    '
Mode: '---------- Measurement mode ----------
    '
    wrt$="BCL"    : GOSUB wrtcmd2  ' Measurement buffer clear
    wrt$="PRN 1" : GOSUB wrtcmd2  ' Printer : ON
    wrt$="FMT 0" : GOSUB wrtcmd2  ' Data output : Standard format
    wrt$="OSC 0" : GOSUB wrtcmd2  ' 1 sec data output : OFF
    wrt$="THR 1" : GOSUB wrtcmd2  ' Threshold output : ON
    wrt$="EPF 1" : GOSUB wrtcmd2  ' Error performance output : ON
    wrt$="ETH 0" : GOSUB wrtcmd2  ' Error performance threshold : 10E-3
    '
    wrt$="PRD 0,0,1,0" : GOSUB wrtcmd2   ' Measurement period 60sec
    wrt$="MOD 1" : GOSUB wrtcmd2  ' Measurement mode : SINGLE
    wrt$="TIM 3" : GOSUB wrtcmd2  ' Display TIMED
    wrt$="DMS 0" : GOSUB wrtcmd2  ' Display ERROR RATIO
    '
    RETURN
    '
Clock: '---------- Set data , Check connection ----------
    '
    DO
        GOSUB Connect              ' Display device connection
        '
        wrt$="CLI?" : GOSUB wrtcmd2 : GOSUB readcmd
        '
        IF MID$(rd$,1,5)="CLI 1" THEN PRINT "** CLOCK LOSS **"
        '
    LOOP UNTIL MID$(rd$,1,5)="CLI 0"
    '
    RETURN
    '
Srch: '---------- Auto search ON ----------
    '
    wrt$="SRH 1" : GOSUB wrtcmd2       ' Set Auto search : ON
    '
    DO
        wrt$="SRH?" : GOSUB wrtcmd2 : GOSUB readcmd
    LOOP UNTIL MID$(rd$,1,5)="SRH 0" or MID$(rd$,1,5)="SRH 2"
    '
    IF MID$(rd$,1,5)="SRH 2"  THEN
        PRINT "<<< Failed in AUTO SEARCH ! >>>"
        PRINT "    ** PROGRAM STOP  **"
        STOP
    END IF
    '
    RETURN
    '
SPoll: '---------- Serial polling--> Measurement END? ----------
    '
    DO
        CALL IBRSP(ED%,SPR%)                        ' Serial polling
        IF IBSTA% < 0 THEN  GOTO trap
```

```
        '
        s=SPR% AND &H10
        q=SPR% AND &H2
        '
        IF s=&H10 THEN GOSUB readcmd : PRINT rd$    ' BUSY bit=1
        '
        tim = 0.1 : GOSUB waidly
    LOOP UNTIL q=&H2                                ' EXTENSION bit=1
    '
    RETURN
        '
Result: '---------- Display Result ----------
        '
    DO
        GOSUB readcmd : PRINT rd$
        '
    LOOP UNTIL MID$(rd$,1,2)="US"
    '
    wrt$="EER?" : GOSUB wrtcmd2          ' RESET Extension byte
    GOSUB readcmd
        '
    RETURN
        '
gpinit: ' ---------- Set up GP-IB functions  ----------
        '
    CALL IBFIND("GPIB0", GPIB0%)         ' Open device (GPIB0)
    IF GPIB0% < 0 THEN GOTO trap         ' system error
    '
    CALL IBFIND("PPG", PPG%)             ' Open device(PPG)
    IF PPG% < 0 THEN GOTO trap           ' system error
    '
    CALL IBFIND("ED", ED%)               ' Open device(ED)
    IF ED% < 0 THEN GOTO trap            ' system error
    '
    CALL IBSIC(GPIB0%)                   ' Interface clear
    IF IBSTA% < 0 THEN GOTO trap         ' system error
    '
    tim = 0.5
    GOSUB waidly
    '
    CALL IBCLR(PPG%)                     ' Device clear(PPG)
    '
    CALL IBCLR(ED%)                      ' Device clear(ED)
    '
    RETURN
        '
wrtcmd1: ' ----------  Write command (PPG) ----------
        '
    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(PPG%, wrt$)               ' Write command (PPG)
    IF IBSTA% < 0 THEN GOTO trap           Trap
    '
    RETURN
        '
wrtcmd2: ' ----------  Write command (ED) ----------
        '
    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(ED%, wrt$)                ' Write command (ED)
    IF IBSTA% < 0 THEN GOTO trap         ' Trap
    '
    RETURN
        '
readcmd: ' ---------- Read command (ED) ----------
        '
    CMD$="?"+chr$(42)+"A"
    CALL IBCMD(GPIB0%,CMD$)
```

```
'
    rd$=SPACE$(25)
    CALL IBRD(GPIB0%, rd$)              ' Read command (GPIB0)
    IF IBSTA% < 0 THEN GOTO trap        ' Trap
'
    RETURN
'
waidly: ' ----------  Wait delay  ----------
'
    stm = TIMER
    etm = TIMER
    WHILE etm - stm < tim
        etm = TIMER
        IF etm < stm THEN etm = etm + 86400
    WEND
'
    RETURN
'
trap:   ' ----------  System trap  ----------
'
    PRINT "IBERR%:" + STR$(IBERR%)
    STOP
'
Connect: '---------- display Connection ----------
'
    WINDOW (-600,-500) -(600,500)
'
    LINE (-50,-50) - (600,500),14,B
    LINE (10,100) - STEP(235,200),,B
    LINE (300,100) - STEP(235,200),,B
'
    CIRCLE (130,130),8
    LINE (130,125) - STEP(0,-55)
    LINE (130,70) - STEP(220,0)
    LINE (350,70) - STEP(0,55)
    CIRCLE (350,130),8
'
    CIRCLE (190,130),8
    LINE (190,125) - STEP(0,-100)
    LINE (190,25) - STEP(220,0)
    LINE (410,25) - STEP(0,100)
    CIRCLE (410,130),8
'
    LOCATE 2,50: PRINT "<<  CONNECTION  >>"
'
    LOCATE 4,42: PRINT "MP1701B/MP1608A     MP1702A/MP1609A"
    LOCATE 5,42: PRINT "    /MP1650A            /MP1651A"
'
    LOCATE 9,45: PRINT "DATA  CLOCK1"
    LOCATE 9,62: PRINT "DATA  CLOCK"
'
    LOCATE 23,1
    INPUT "   Aer you ready ?  Press return key to start    ",A
'
    LOCATE 23,1
    PRINT "                                                  "
'
    LOCATE 4,1
'
    RETURN
'
    END
```

EXECUTED RESULT

Output to GP-IB

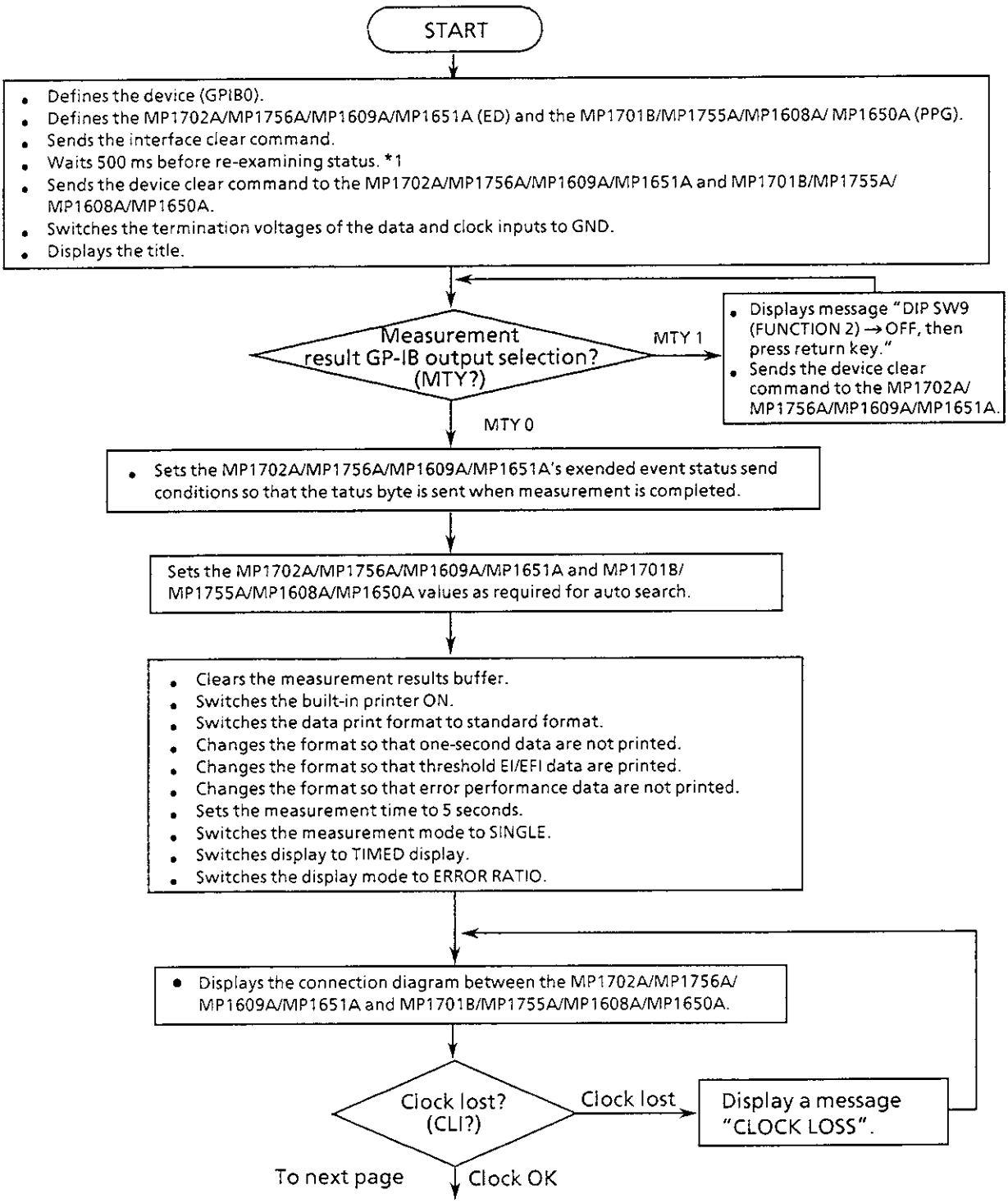
```
STA 90,02,28,15,40,07

STA 90,02,28,15,40,07

END 90,02,28,15,41,07

ELP      0,00,01,00

ER  1.0000E-04

EC     30000073

EI           60

EFI      0.0000

PFI           0

CLI           0

SLI           0

TH3           0,100.0000

TH4          60,   0.0000

TH5          60,   0.0000

TH6          60,   0.0000

TH7          60,   0.0000

TH8          60,   0.0000

TH9           0,100.0000

ES     100.0000

EFS      0.0000

SES      0.0000

DM     100.0000

US       0.0000
```

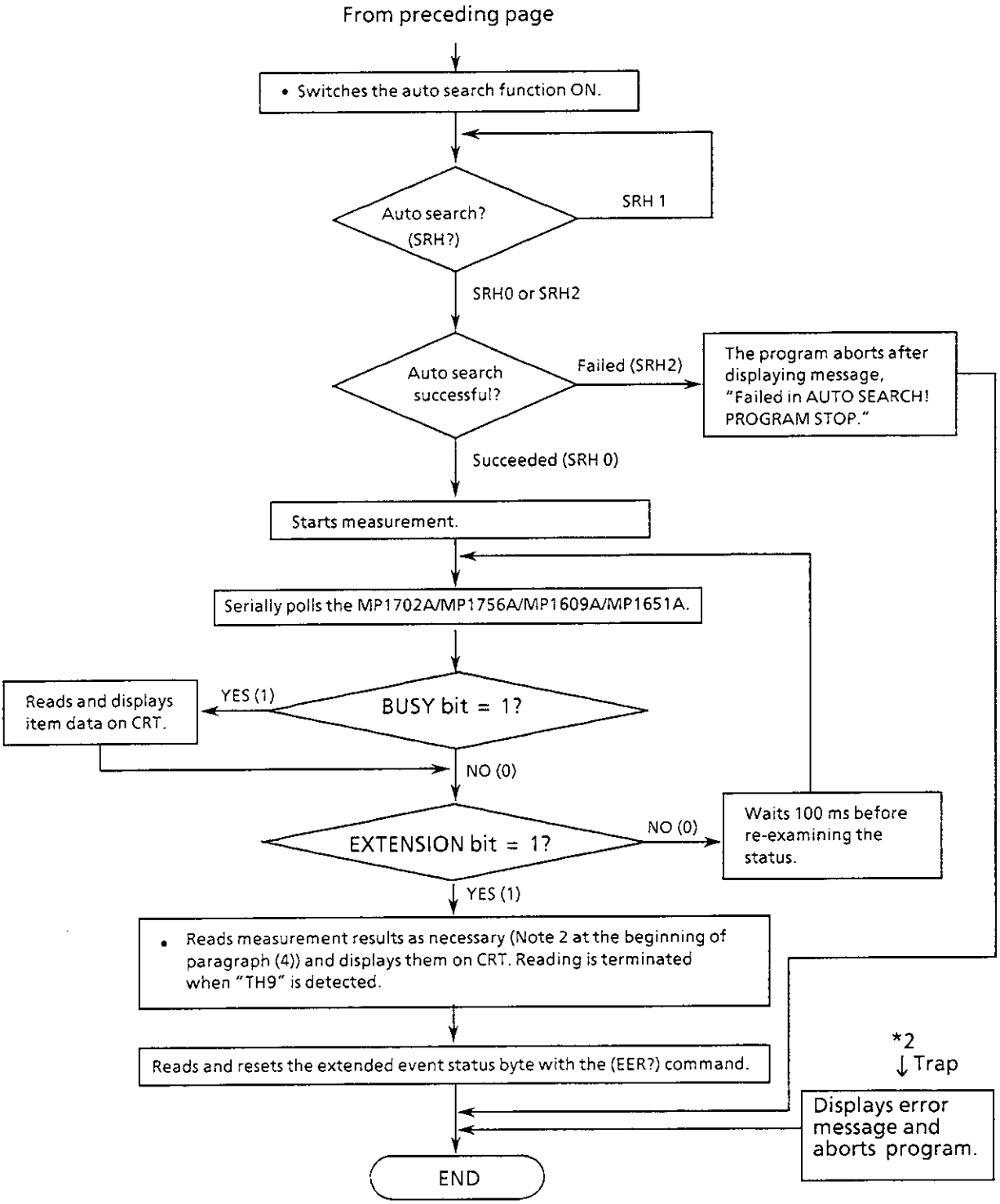
Printout from built-in printer

```
90-02-28/15:40:07 START

************************
 START 90-02-28/15:40:07
 END   90-02-28/15:41:07
 ELP          0/00:01:00
 ==ERROR MEASUREMENT==
 ERROR RATIO 1.0000E-04
 ERROR COUNT   30000073
 EI                   60
 EFI              0.0000%
 PFI                   0
 CLI                   0
 SLI                   0
 == THRESHOLD EI,EFI ==
 ER    EI         EFI
 >-3          0 100.0000%
 >-4         60   0.0000%
 >-5         60   0.0000%
 >-6         60   0.0000%
 >-7         60   0.0000%
 >-8         60   0.0000%
 =<-8         0 100.0000%
 ==ERROR PERFORMANCE==
 ERRORED SEC    100.0000%
 ER FREE SEC      0.0000%
 SES (E-3)        0.0000%
 DM  (E-6)      100.0000%
 UNAVAIL SEC      0.0000%
************************
```

(e) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Extended data (threshold measurement data) are output in standard format at the end of measurement, as determined by serial polling.





Note: See the beginning of this paragraph 15.5 for *1 and *2.

Program listing

```
' ************************************************************
' *                                                          *
' *     MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT     *
' *        -- Standard format , Extension outout --          *
' *                  -- Threshold EI/EFI --                  *
' *                                                ED_M5     *
' ************************************************************
'
'------------------------------------------------------------'
'                          MAIN  ROUTINE                     '
'------------------------------------------------------------'
'
common shared IBSTA%,IBERR%,IBCNT%   ' Setup GPIB-PC functions
GOSUB gpinit                         ' Setup GPIB interface
wrt$="DTM 0;CTM 0" : GOSUB wrtcmd2
'
SCREEN 9
PRINT "** MP1702A/MP1609A/MP1651A **"
PRINT "** MEASUREMENT SAMPLE SOFT **"
PRINT
'
GOSUB  MType                         ' Check Measurement type
'
wrt$="EES 64" : GOSUB wrtcmd2        ' EES : MEASURE bit
'
GOSUB DSet                           ' Data setting
GOSUB Mode                           ' Measurement mode
GOSUB Clock                          ' Check Clock loss ?
GOSUB Srch                           ' Auto search : ON
'
wrt$="STA"  : GOSUB wrtcmd2          ' Measurement Start
'
GOSUB SPoll                    ' Serial polling--> Extension bit=1
GOSUB Result                   ' Display Result
'
STOP
'
'------------------------------------------------------------'
'                          SUB ROUTINE                       '
'------------------------------------------------------------'
'
MType : '---------- Check Measurement type ----------
'
    DO
        wrt$="MTY?" : GOSUB wrtcmd2 : GOSUB readcmd
'
        IF MID$(rd$,1,5)="MTY 1" THEN
          LOCATE 23,1
          INPUT "DIP SW 9 (Function 2)-->OFF, then press Enter key",A
          CALL IBCLR(ED%)
'
        END IF
    LOOP UNTIL MID$(rd$,1,5)="MTY 0"
'
    RETURN
'
DSet: '---------- Data Setting ----------
'
    wrt$= "LGC 0;PTN 2;MPK 1"     : GOSUB wrtcmd1
    wrt$= "LGC 0;PTN 2;MPK 1"     : GOSUB wrtcmd2
'
```

```
    wrt$= "CLK 1;RES 1;FRQ 3000" : GOSUB wrtcmd1
    wrt$= "OFS 0;SPD 0"          : GOSUB wrtcmd1
    wrt$= "DAP 1;DOS 1"          : GOSUB wrtcmd1
    wrt$= "CDL 100;CAP 1;COS 1"  : GOSUB wrtcmd1
    wrt$= "EAD 1"                : GOSUB wrtcmd1
    '
    wrt$="CPL 0"                 : GOSUB wrtcmd2
    wrt$="SRH 0"                 : GOSUB wrtcmd2
    wrt$="SYN 1"                 : GOSUB wrtcmd2
    '
    RETURN
    '
Mode: '---------- Measurement mode ----------
    '
    wrt$="BCL"    : GOSUB wrtcmd2  ' Measurement buffer clear
    wrt$="PRN 1" : GOSUB wrtcmd2  ' Printer : ON
    wrt$="FMT 0" : GOSUB wrtcmd2  ' Data output : Standard format
    wrt$="OSC 0" : GOSUB wrtcmd2  ' 1 sec data output : OFF
    wrt$="THR 1" : GOSUB wrtcmd2  ' Threshold output : ON
    wrt$="EPF 0" : GOSUB wrtcmd2  ' Error performance output : OFF
    '
    wrt$="PRD 0,0,0,5" : GOSUB wrtcmd2    ' Measurement period 5sec
    wrt$="MOD 1" : GOSUB wrtcmd2  ' Measurement mode : SINGLE
    wrt$="TIM 3" : GOSUB wrtcmd2  ' Display TIMED
    wrt$="DMS 0" : GOSUB wrtcmd2  ' Display ERROR RATIO
    '
    RETURN
    '
Clock: '---------- Set data , Check connection ----------
    '
    DO
        GOSUB Connect              ' Display device connection
        '
        wrt$="CLI?" : GOSUB wrtcmd2 : GOSUB readcmd
        '
        IF MID$(rd$,1,5)="CLI 1" THEN PRINT "** CLOCK LOSS **"
        '
    LOOP UNTIL MID$(rd$,1,5)="CLI 0"
    '
    RETURN
    '
Srch: '---------- Auto search ON ----------
    '
    wrt$="SRH 1" : GOSUB wrtcmd2       ' Set Auto search : ON
    '
    DO
        wrt$="SRH?" : GOSUB wrtcmd2 : GOSUB readcmd
    LOOP UNTIL MID$(rd$,1,5)="SRH 0" or MID$(rd$,1,5)="SRH 2"
    '
    IF MID$(rd$,1,5)="SRH 2"  THEN
        PRINT "<<< Failed in AUTO SEARCH ! >>>"
        PRINT "     ** PROGRAM STOP  **"
        STOP
    END IF
    '
    RETURN
    '
SPoll: '---------- Serial polling--> Measurement END? ----------
    '
    DO
        CALL IBRSP(ED%,SPR%)                          Serial polling
        IF IBSTA% < 0 THEN  GOTO trap
        '
        s=SPR% AND &H10
        q=SPR% AND &H2
        '
```

```
        IF s=&H10 THEN GOSUB readcmd : PRINT rd$   ' BUSY bit=1

        tim = 0.1 : GOSUB waidly
    LOOP UNTIL q=&H2                               ' EXTENSION bit=1

    RETURN

Result: '---------- Display Result ----------

    DO
        GOSUB readcmd : PRINT rd$

    LOOP UNTIL MID$(rd$,1,3)="TH?"

    wrt$="EER?" : GOSUB wrtcmd2          ' RESET Extension byte
    GOSUB readcmd

    RETURN

gpinit: ' ---------- Set up GP-IB functions  ----------

    CALL IBFIND("GPIB0", GPIB0%)         ' Open device (GPIB0)
    IF GPIB0% < 0 THEN GOTO trap         ' system error

    CALL IBFIND("PPG", PPG%)             ' Open device(PPG)
    IF PPG% < 0 THEN GOTO trap           ' system error

    CALL IBFIND("ED", ED%)               ' Open device(ED)
    IF ED% < 0 THEN GOTO trap            ' system error

    CALL IBSIC(GPIB0%)                   ' Interface clear
    IF IBSTA% < 0 THEN GOTO trap         ' system error

    tim = 0.5
    GOSUB waidly

    CALL IBCLR(PPG%)                     ' Device clear(PPG)

    CALL IBCLR(ED%)                      ' Device clear(ED)

    RETURN

wrtcmd1: ' ----------  Write command (PPG) ----------

    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(PPG%, wrt$)               ' Write command (PPG)
    IF IBSTA% < 0 THEN GOTO trap         ' Trap

    RETURN

wrtcmd2: ' ----------  Write command (ED) ----------

    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(ED%, wrt$)                ' Write command (ED)
    IF IBSTA% < 0 THEN GOTO trap         ' Trap

    RETURN

readcmd: ' ---------- Read command (ED) ----------

    CMD$="?"+chr$(42)+"A"
    CALL IBCMD(GPIB0%,CMD$)
```

```
    rd$=SPACE$(25)
    CALL IBRD(GPIB0%, rd$)                ' Read command (GPIB0)
    IF IBSTA% < 0 THEN GOTO trap          ' Trap
    '
    RETURN
    '
waidly: ' ----------  Wait delay  ----------
    '
    stm = TIMER
    etm = TIMER
    WHILE etm - stm < tim
        etm = TIMER
        IF etm < stm THEN etm = etm + 86400
    WEND
    '
    RETURN
    '
trap:   ' ----------  System trap  ----------
    '
    PRINT "IBERR%:" + STR$(IBERR%)
     STOP
    '
Connect: '---------- display Connection ----------
    '
    WINDOW (-600,-500) -(600,500)
    '
    LINE (-50,-50) - (600,500),14,B
    LINE (10,100) - STEP(235,200),,B
    LINE (300,100) - STEP(235,200),,B
    '
    CIRCLE (130,130),8
    LINE (130,125) - STEP(0,-55)
    LINE (130,70) - STEP(220,0)
    LINE (350,70) - STEP(0,55)
    CIRCLE (350,130),8
    '
    CIRCLE (190,130),8
    LINE (190,125) - STEP(0,-100)
    LINE (190,25) - STEP(220,0)
    LINE (410,25) - STEP(0,100)
    CIRCLE (410,130),8
    '
    LOCATE 2,50: PRINT "<<  CONNECTION  >>"
    '
    LOCATE 4,42: PRINT "MP1701B/MP1608A    MP1702A/MP1609A"
    LOCATE 5,42: PRINT "    /MP1650A           /MP1651A"
    '
    LOCATE 9,45: PRINT "DATA  CLOCK1"
    LOCATE 9,62: PRINT "DATA  CLOCK"
    '
    LOCATE 23,1
    INPUT "   Aer you ready ?  Press return key to start    ",A
    '
    LOCATE 23,1
    PRINT "                                                  "
    '
    LOCATE 4,1
    '
    RETURN
    '
    END
```

EXECUTED RESULT

Output to GP-IB

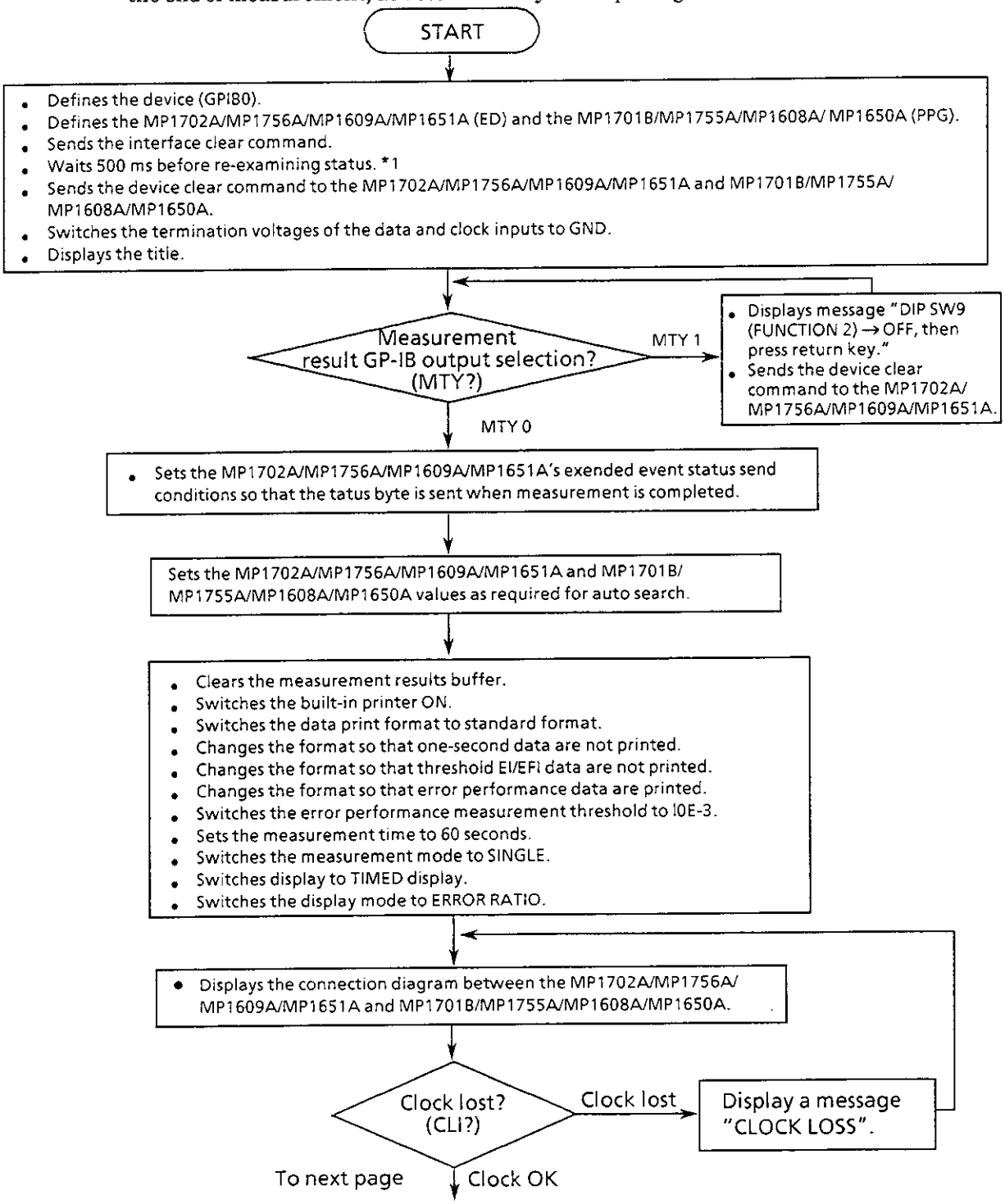
```
STA 90,02,28,15,43,58

STA 90,02,28,15,43,58

END 90,02,28,15,44,03

ELP      0,00,00,05

ER   1.0000E-04

EC       2500006

EI             5

EFI       0.0000

PFI            0

CLI            0

SLI            0

TH3            0,100.0000

TH4            5,   0.0000

TH5            5,   0.0000

TH6            5,   0.0000

TH7            5,   0.0000

TH8            5,   0.0000

TH9            0,100.0000
```

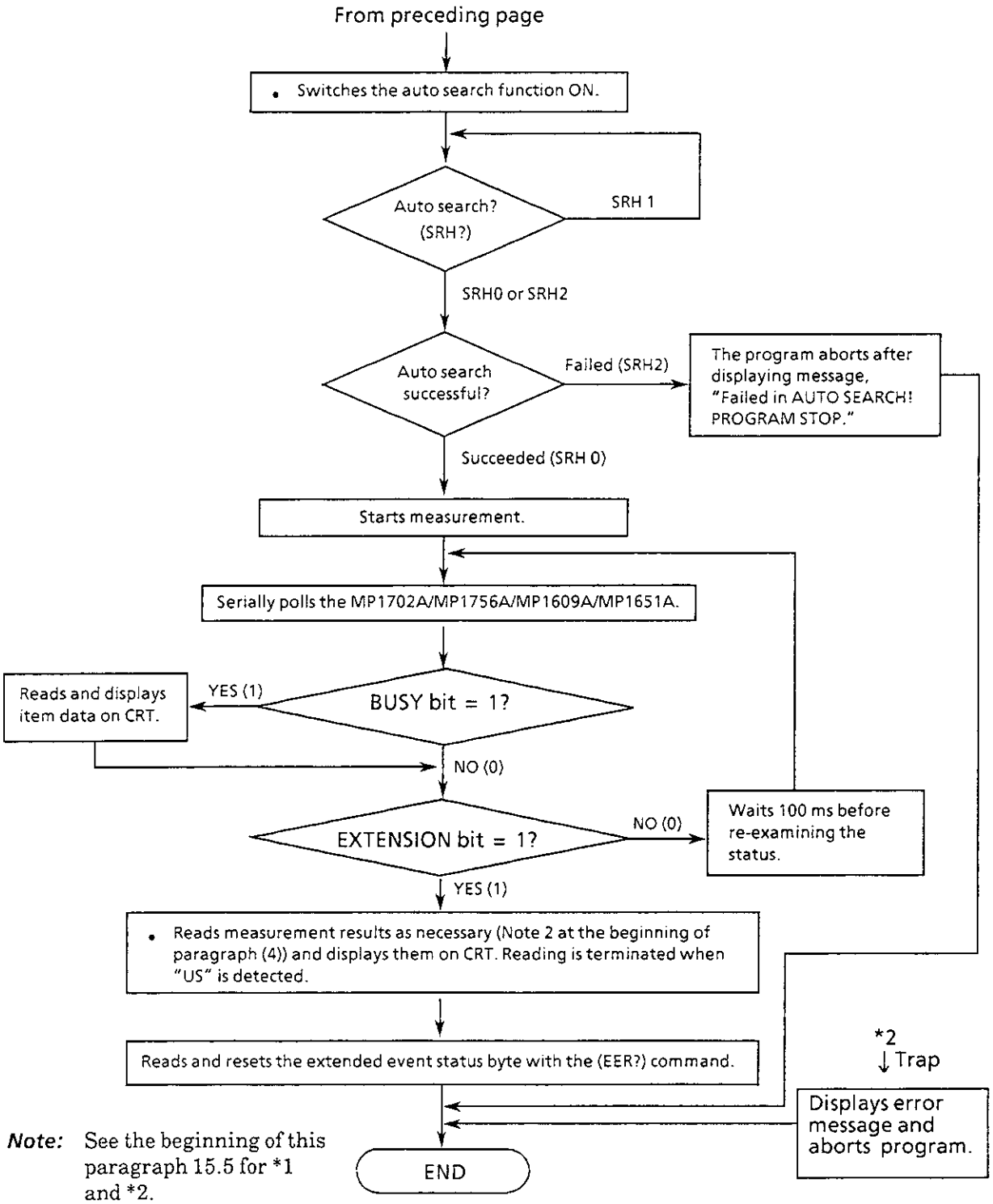
Printout from built-in printer

```
90-02-28/15:43:58 START

**************************
 START 90-02-28/15:43:58
 END   90-02-28/15:44:03
 ELP           0/00:00:05
 ==ERROR MEASUREMENT==
 ERROR RATIO 1.0000E-04
 ERROR COUNT    2500006
 EI                   5
 EFI                0.0000%
 PFI                  0
 CLI                  0
 SLI                  0
 == THRESHOLD EI,EFI ==
 ER    EI          EFI
 >-3          0 100.0000%
 >-4          5   0.0000%
 >-5          5   0.0000%
 >-6          5   0.0000%
 >-7          5   0.0000%
 >-8          5   0.0000%
 =<-8         0 100.0000%
**************************
```

(f) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Extended data (error performance measurement data) are output in standard format at the end of measurement, as determined by serial polling.





Note: See the beginning of this paragraph 15.5 for *1 and *2.

Program listing

```
' ************************************************************
' *                                                          *
' *     MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT     *
' *        -- Standard format , Extension output --          *
' *                  --  Error performance --                *
' *                                                  ED_M6   *
' ************************************************************
'
'------------------------------------------------------------'
'                           MAIN  ROUTINE                    '
'------------------------------------------------------------'
'
common shared IBSTA%,IBERR%,IBCNT%   ' Setup GPIB-PC functions
GOSUB gpinit                         ' Setup GPIB interface
wrt$="DTM 0;CTM 0" : GOSUB wrtcmd2
'
SCREEN 9
PRINT "** MP1702A/MP1609A/MP1651A **"
PRINT "** MEASUREMENT SAMPLE SOFT **"
'
GOSUB  MType                         ' Check Measurement type
'
wrt$="EES 64" : GOSUB wrtcmd2        ' EES : MEASURE bit
'
GOSUB DSet                           ' Data setting
GOSUB Mode                           ' Measurement mode
GOSUB Clock                          ' Check Clock loss ?
GOSUB Srch                           ' Auto search : ON
'
wrt$="STA"  : GOSUB wrtcmd2          ' Measurement Start
'
GOSUB SPoll                    ' Serial polling--> Extension bit=1
GOSUB Result                   ' Display Result
'
STOP
'
'------------------------------------------------------------'
'                           SUB ROUTINE                      '
'------------------------------------------------------------'
      '
MType :  '---------- Check Measurement type ----------
      '
    DO
        wrt$="MTY?" : GOSUB wrtcmd2 : GOSUB readcmd
        '
        IF MID$(rd$,1,5)="MTY 1" THEN
          INPUT "DIP SW 9 (Function 2)-->OFF, then press Enter key",A
          CALL IBCLR(ED%)
          '
        END IF
    LOOP UNTIL MID$(rd$,1,5)="MTY 0"
    '
    RETURN
      '
DSet:  '---------- Data Setting ----------
      '
    wrt$= "LGC 0;PTN 2;MRK 1"     : GOSUB wrtcmd1
    wrt$= "LGC 0;PTN 2;MRK 1"     : GOSUB wrtcmd2
    '
```

```
    wrt$= "CLK 1;RES 1;FRQ 3000" : GOSUB wrtcmd1
    wrt$= "OFS 0;SPD 0"          : GOSUB wrtcmd1
    wrt$= "DAP 1;DOS 1"          : GOSUB wrtcmd1
    wrt$= "CDL 100;CAP 1;COS 1"  : GOSUB wrtcmd1
    wrt$= "EAD 1"                : GOSUB wrtcmd1
    '
    wrt$="CPL 0"                 : GOSUB wrtcmd2
    wrt$="SRH 0"                 : GOSUB wrtcmd2
    wrt$="SYN 1"                 : GOSUB wrtcmd2
    '
    RETURN
    '
Mode: '---------- Measurement mode ----------
    '
    wrt$="BCL"    : GOSUB wrtcmd2   ' Measurement buffer clear
    wrt$="PRN 1" : GOSUB wrtcmd2   ' Printer : ON
    wrt$="FMT 0" : GOSUB wrtcmd2   ' Data output : Standard format
    wrt$="OSC 0" : GOSUB wrtcmd2   ' 1 sec data output : OFF
    wrt$="THR 0" : GOSUB wrtcmd2   ' Threshold output : OFF
    wrt$="EPF 1" : GOSUB wrtcmd2   ' Error performance output : ON
    wrt$="ETH 0" : GOSUB wrtcmd2   ' Error performance threshold : 10E-3
    '
    wrt$="PRD 0,0,1,0" : GOSUB wrtcmd2    ' Measurement period 60sec
    wrt$="MOD 1" : GOSUB wrtcmd2   ' Measurement mode : SINGLE
    wrt$="TIM 3" : GOSUB wrtcmd2   ' Display TIMED
    wrt$="DMS 0" : GOSUB wrtcmd2   ' Display ERROR RATIO
    '
    RETURN
    '
Clock: '---------- Set data , Check connection ----------
    '
    DO
        GOSUB Connect              ' Display device connection
        '
        wrt$="CLI?" : GOSUB wrtcmd2 : GOSUB readcmd
        '
        IF MID$(rd$,1,5)="CLI 1" THEN PRINT "** CLOCK LOSS **"
        '
    LOOP UNTIL MID$(rd$,1,5)="CLI 0"
    '
    RETURN
    '
Srch: '---------- Auto search ON ----------
    '
    wrt$="SRH 1" : GOSUB wrtcmd2       ' Set Auto search : ON
    '
    DO
        wrt$="SRH?" : GOSUB wrtcmd2 : GOSUB readcmd
    LOOP UNTIL MID$(rd$,1,5)="SRH 0" or MID$(rd$,1,5)="SRH 2"
    '
    IF MID$(rd$,1,5)="SRH 2"  THEN
        PRINT "<<< Failed in AUTO SEARCH ! >>>"
        PRINT "    ** PROGRAM STOP  **"
        STOP
    END IF
    '
    RETURN
    '
SPoll: '---------- Serial polling--> Measurement END? ----------
    '
    DO
        CALL IBRSP(ED%,SPR%)                          ' Serial polling
        IF IBSTA% < 0 THEN  GOTO trap
        '
        s=SPR% AND %H10
        q=SPR% AND %H2
```

```
        '
        IF s=&H10 THEN GOSUB readcmd : PRINT rd$    ' BUSY bit=1
        '
        tim = 0.1 : GOSUB waidly
    LOOP UNTIL q=&H2                                ' EXTENSION bit=1
    '
    RETURN
    '
Result: '---------- Display Result ----------
    '
    DO
        GOSUB readcmd : PRINT rd$
    '
    LOOP UNTIL MID$(rd$,1,2)="US"
    '
    wrt$="EER?" : GOSUB wrtcmd2          ' RESET Extension byte
    GOSUB readcmd
    '
    RETURN
    '
gpinit: ' ---------- Set up GP-IB functions  ----------
    '
    CALL IBFIND("GPIB0", GPIB0%)         ' Open device (GPIB0)
    IF GPIB0% < 0 THEN GOTO trap         ' system error
    '
    CALL IBFIND("PPG", PPG%)             ' Open device(PPG)
    IF PPG% < 0 THEN GOTO trap           ' system error
    '
    CALL IBFIND("ED", ED%)               ' Open device(ED)
    IF ED% < 0 THEN GOTO trap            ' system error
    '
    CALL IBSIC(GPIB0%)                   ' Interface clear
    IF IBSTA% < 0 THEN GOTO trap         ' system error
    '
    tim = 0.5
    GOSUB waidly
    '
    CALL IBCLR(PPG%)                     ' Device clear(PPG)
    '
    CALL IBCLR(ED%)                      ' Device clear(ED)
    '
    RETURN
    '
wrtcmd1: ' ----------  Write command (PPG) ----------
    '
    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(PPG%, wrt$)               ' Write command (PPG)
    IF IBSTA% < 0 THEN GOTO trap         ' Trap
    '
    RETURN
    '
wrtcmd2: ' ----------  Write command (ED) ----------
    '
    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(ED%, wrt$)                ' Write command (ED)
    IF IBSTA% < 0 THEN GOTO trap         ' Trap
    '
    RETURN
    '
readcmd: ' ---------- Read command (ED) ----------
    '
    CMD$="?"+chr$(42)+"A"
    CALL IBCMD(GPIB0%,CMD$)
    '
    rd$=SPACE$(23)
    CALL IBRD(GPIB0%, rd$)               ' Read command (GPIB0)
```

```
      IF IBSTA% < 0 THEN GOTO trap          ' Trap
      '
      RETURN
          '
waidly:   ' ----------  Wait delay  ----------
          '
      stm = TIMER
      etm = TIMER
      WHILE etm - stm < tim
          etm = TIMER
          IF etm < stm THEN etm = etm + 86400
      WEND
      '
      RETURN
          '
trap:     ' ----------  System trap  ----------
          '
      PRINT "IBERR%:" + STR$(IBERR%)
       STOP
          '
Connect:  '---------- display Connection ----------
          '
      SCREEN 9
      WINdOW (-600,-500) -(600,500)
      '
      PRINT "  ** MP1702A/MP1609A **"
      PRINT " MEASUREMENT SAMPLE SOFT "
      PRINT
      '
      LINE (-50,-50) - (600,500),14,B
      LINE (10,100) - STEP(235,200),,B
      LINE (300,100) - STEP(235,200),,B
      '
      CIRCLE (130,130),8
      LINE (130,125) - STEP(0,-55)
      LINE (130,70) - STEP(220,0)
      LINE (350,70) - STEP(0,55)
      CIRCLE (350,130),8
      '
      CIRCLE (190,130),8
      LINE (190,125) - STEP(0,-100)
      LINE (190,25) - STEP(220,0)
      LINE (410,25) - STEP(0,100)
      CIRCLE (410,130),8
      '
      LOCATE 2,50: PRINT "<<  CONNECTION  >>"
      '
      LOCATE 4,42: PRINT "MP1701B/MP1608A     MP1702A/MP1609A"
      LOCATE 5,42: PRINT "     /MP1650A            /MP1651A"
      '
      LOCATE 9,45: PRINT "DATA  CLOCK1"
      LOCATE 9,62: PRINT "DATA  CLOCK"
      '
      LOCATE 23,1
      INPUT "Aer you ready ?  Press return key to start ",A
      '
      LOCATE 23,1
      PRINT "                                                "
      '
      LOCATE 4,1
      '
      RETURN
      '
      END
```

EXECUTED RESULT

Output to GP-IB

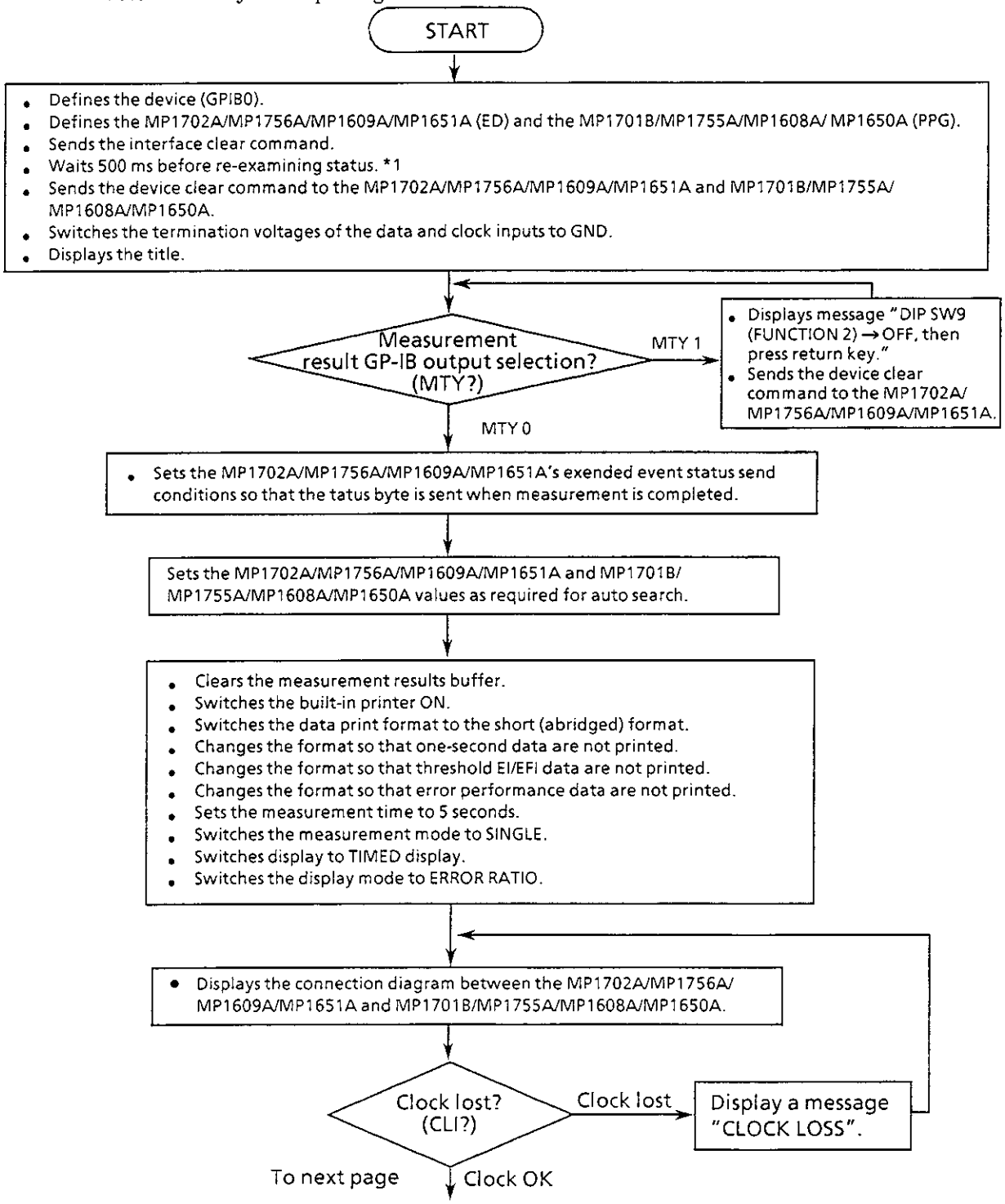
```
STA 90,02,28,15,46,34

STA 90,02,28,15,46,34

END 90,02,28,15,47,34

ELP      0,00,01,00

ER  1.0000E-04

EC     30000073

EI           60

EFI       0.0000

PFI            0

CLI            0

SLI            0

ES     100.0000

EFS       0.0000

SES       0.0000

DM     100.0000

US        0.0000
```

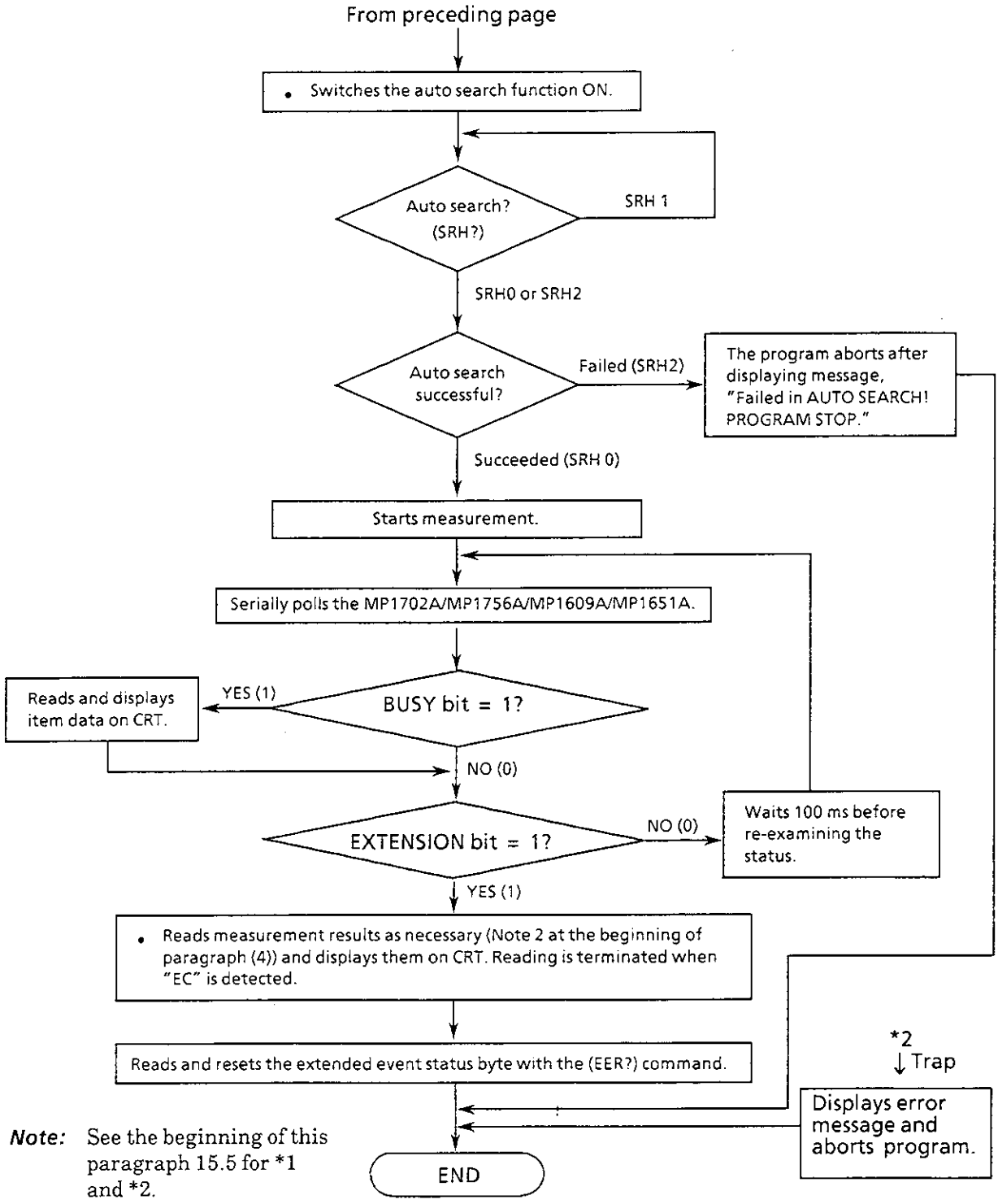
Printout from built-in printer

```
90-02-28/15:46:34 START

**********************
 START 90-02-28/15:46:34
 END   90-02-28/15:47:34
 ELP           0/00:01:00
 ==ERROR MEASUREMENT==
 ERROR RATIO 1.0000E-04
 ERROR COUNT   30000073
 EI                   60
 EFI                0.0000%
 PFI                   0
 CLI                   0
 SLI                   0
 ==ERROR PERFORMANCE==
 ERRORED SEC    100.0000%
 ER FREE SEC      0.0000%
 SES (E-3)        0.0000%
 DM  (E-6)      100.0000%
 UNAVAIL SEC      0.0000%
**********************
```

(g) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Standard data are output in short (abridged) format at the end of measurement, as determined by serial polling.





Note: See the beginning of this paragraph 15.5 for *1 and *2.

Program listing

```
' ************************************************************
' *                                                          *
' *     MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT     *
' *        -- Abridged format : Standard output --           *
' *        -- Measurement end? >> Extension bit --           *
' *                                                  ED_M7   *
' ************************************************************
'
'-------------------------------------------------------------'
'                          MAIN  ROUTINE                      '
'-------------------------------------------------------------'
'
common shared IBSTA%,IBERR%,IBCNT%   ' Setup GPIB-PC functions
GOSUB gpinit                         ' Setup GPIB interface
wrt$="DTM 0;CTM 0"      : GOSUB wrtcmd2
'
SCREEN 9
PRINT "** MP1702A/MP1609A/MP1651A **"
PRINT "** MEASUREMENT SAMPLE SOFT **"
PRINT
'
GOSUB  MType                         ' Check Measurement type
'
wrt$="EES 64" : GOSUB wrtcmd2        ' EES : MEASURE bit
'
GOSUB DSet                           ' Data setting
GOSUB Mode                           ' Measurement mode
GOSUB Clock                          ' Check Clock loss ?
GOSUB Srch                           ' Auto search : ON
'
wrt$="STA"  : GOSUB wrtcmd2          ' Measurement Start
'
GOSUB SPoll                    ' Serial polling--> Extension bit=1
GOSUB Result                   ' Display Result
'
STOP
'
'-------------------------------------------------------------'
'                          SUB ROUTINE                        '
'-------------------------------------------------------------'
'
MType : '---------- Check Measurement type ----------
'
    DO
        wrt$="MTY?" : GOSUB wrtcmd2 : GOSUB readcmd
        '
        IF MID$(rd$,1,5)="MTY 1" THEN
          LOCATE 23,1
          INPUT "DIP SW 9 (Function 2)-->OFF, then press Enter key",A
          CALL IBCLR(ED%)
          '
        END IF
    LOOP UNTIL MID$(rd$,1,5)="MTY 0"
    '
    RETURN
    '
DSet: '---------- Data Setting ----------
    '
    wrt$= "LGC 0;FTN 2;MRK 1"     : GOSUB wrtcmd1
    wrt$= "LGC 0;FTN 2;MRK 1"     : GOSUB wrtcmd2
    '
```

```
    wrt$= "CLK 1;RES 1;FRQ 3000" : GOSUB wrtcmd1
    wrt$= "OFS 0;SPD 0"          : GOSUB wrtcmd1
    wrt$= "DAP 1;DOS 1"          : GOSUB wrtcmd1
    wrt$= "CDL 100;CAP 1;COS 1"  : GOSUB wrtcmd1
    wrt$= "EAD 1"                : GOSUB wrtcmd1
    '
    wrt$="CPL 0"                 : GOSUB wrtcmd2
    wrt$="SRH 0"                 : GOSUB wrtcmd2
    wrt$="SYN 1"                 : GOSUB wrtcmd2
    '
    RETURN
    '
Mode: '---------- Measurement mode ----------
    '
    wrt$="BCL"    : GOSUB wrtcmd2  ' Measurement buffer clear
    wrt$="PRN 1" : GOSUB wrtcmd2  ' Printer : ON
    wrt$="FMT 1" : GOSUB wrtcmd2  ' Data output : Abridged format
    wrt$="OSC 0" : GOSUB wrtcmd2  ' 1 sec data print : OFF
    wrt$="THR 0" : GOSUB wrtcmd2  ' Threshold output : OFF
    wrt$="EPF 0" : GOSUB wrtcmd2  ' Error performance output : OFF
    '
    wrt$="PRD 0,0,0,5" : GOSUB wrtcmd2   ' Measurement period 5sec
    wrt$="MOD 1" : GOSUB wrtcmd2  ' Measurement mode : SINGLE
    wrt$="TIM 3" : GOSUB wrtcmd2  ' Display TIMED
    wrt$="DMS 0" : GOSUB wrtcmd2  ' Display ERROR RATIO
    '
    RETURN
    '
Clock: '---------- Set data , Check connection ----------
    '
    DO
        GOSUB Connect              ' Display device connection
        '
        wrt$="CLI?" : GOSUB wrtcmd2 : GOSUB readcmd
        '
        IF MID$(rd$,1,5)="CLI 1" THEN PRINT "** CLOCK LOSS **"
        '
    LOOP UNTIL MID$(rd$,1,5)="CLI 0"
    '
    RETURN
    '
Srch: '---------- Auto search ON ----------
    '
    wrt$="SRH 1" : GOSUB wrtcmd2      ' Set Auto search : ON
    '
    DO
        wrt$="SRH?" : GOSUB wrtcmd2 : GOSUB readcmd
    LOOP UNTIL MID$(rd$,1,5)="SRH 0" or MID$(rd$,1,5)="SRH 2"
    '
    IF MID$(rd$,1,5)="SRH 2"  THEN
        PRINT "<<< Failed in AUTO SEARCH ! >>>"
        PRINT "     ** PROGRAM STOP  **"
        STOP
    END IF
    '
    RETURN
    '
SPoll: '---------- Serial polling--> Measurement END? ----------
    '
    DO
        CALL IBRSP(ED%,SPR%)                          ' Serial polling
        IF IBSTA% < 0 THEN  GOTO trap
        '
        s=SPR% AND &H10
        q=SPR% AND &H2
        '
```

```
        IF s=&H10 THEN GOSUB readcmd : PRINT rd$    ' BUSY bit=1

        tim = 0.1 : GOSUB waidly
    LOOP UNTIL q=&H2                                ' Extension bit=1

    RETURN

Result: '---------- Display Result ----------

    DO
        GOSUB readcmd : PRINT rd$

    LOOP UNTIL MID$(rd$,1,2)="EC"

    wrt$="EER?" : GOSUB wrtcmd2          ' RESET Extension byte
    GOSUB readcmd

    RETURN

gpinit: ' ---------- Set up GP-IB functions  ----------

    CALL IBFIND("GPIB0", GPIB0%)         ' Open device (GPIB0)
    IF GPIB0% < 0 THEN GOTO trap         ' system error

    CALL IBFIND("PPG", PPG%)             ' Open device(PPG)
    IF PPG% < 0 THEN GOTO trap           ' system error

    CALL IBFIND("ED", ED%)               ' Open device(ED)
    IF ED% < 0 THEN GOTO trap            ' system error

    CALL IBSIC(GPIB0%)                   ' Interface clear
    IF IBSTA% < 0 THEN GOTO trap         ' system error

    tim = 0.5
    GOSUB waidly

    CALL IBCLR(PPG%)                     ' Device clear(PPG)

    CALL IBCLR(ED%)                      ' Device clear(ED)

    RETURN

wrtcmd1: ' ----------  Write command (PPG) ----------

    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(PPG%, wrt$)               ' Write command (PPG)
    IF IBSTA% < 0 THEN GOTO trap         ' Trap

    RETURN

wrtcmd2: ' ----------  Write command (ED) ----------

    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(ED%, wrt$)                ' Write command (ED)
    IF IBSTA% < 0 THEN GOTO trap         ' Trap

    RETURN

readcmd: ' ---------- Read command (ED) ----------

    CMD$="?"+chr$(42)+"A"
    CALL IBCMD(GPIB0%,CMD$)
```

```
    rd$=SPACE$(23)
    CALL IBRD(GPIB0%, rd$)              ' Read command (GPIB0)
    IF IBSTA% < 0 THEN GOTO trap        ' Trap
    '
    RETURN
    '
waidly: ' ----------  Wait delay  ----------
    '
    stm = TIMER
    etm = TIMER
    WHILE etm - stm < tim
        etm = TIMER
        IF etm < stm THEN etm = etm + 86400
    WEND
    '
    RETURN
    '
trap:   ' ----------  System trap  ----------
    '
    PRINT "IBERR%:" + STR$(IBERR%)
     STOP
    '
Connect: '---------- display Connection ----------
    '
    WINDOW (-600,-500) -(600,500)
    '
    LINE (-50,-50) - (600,500),14,B
    LINE (10,100) - STEP(235,200),,B
    LINE (300,100) - STEP(235,200),,B
    '
    CIRCLE (130,130),8
    LINE (130,125) - STEP(0,-55)
    LINE (130,70) - STEP(220,0)
    LINE (350,70) - STEP(0,55)
    CIRCLE (350,130),8
    '
    CIRCLE (190,130),8
    LINE (190,125) - STEP(0,-100)
    LINE (190,25) - STEP(220,0)
    LINE (410,25) - STEP(0,100)
    CIRCLE (410,130),8
    '
    LOCATE 2,50: PRINT "<<  CONNECTION  >>"
    '
    LOCATE 4,42: PRINT "MP1701B/MP1608A    MP1702A/MP1609A"
    LOCATE 5,42: PRINT "    /MP1650A           /MP1651A"
    '
    LOCATE 9,45: PRINT "DATA  CLOCK1"
    LOCATE 9,62: PRINT "DATA  CLOCK"
    '
    LOCATE 23,1
    INPUT "   Aer you ready ?  Press return key to start    ",A
    '
    LOCATE 23,1
    PRINT "                                                  "
    '
    LOCATE 4,1
    '
    RETURN
    '
    END
```

EXECUTED RESULT

Output to GP-IB

```
STA 90,02,28,15,55,52
END 90,02,28,15,55,57
ER  1.0000E-04
EC      2500006
```

Printout from built-in printer

```
90-02-28/15:55:52 START
 END   90-02-28/15:55:57
ER 1.00E-04 EC  2500006
```

(5) Measurement (Measurement type: Output results each time a measurement is completed)

Intermediate measurement results are output each time an (IMD?) request command is issued.

The following programs, (a) to (e), are created for measurement where the measurement result GP-IB output selection is set for output to be made each time a measurement is completed.

Intermediate results are output each time an (IMD) command is issued.

In these programs, measurement is initiated after connecting DATA to DATA and CLOCK1 to CLOCK between the MP1702A/MP1756A/MP1609A/MP1651A and MP1701B/MP1755A/ MP1608A/ MP1650A.

First, the program initializes conditions for auto search and for the appropriate measurement.

Next, it confirms that the clock is not lost and, after confirmation, switches the auto search function ON.

If auto search is successfully completed, measurement is begun. Whenever [RETURN] is pressed, the (IMD?) request command is issued, and measurement results are read and displayed on the CRT. The data read terminate conditions depend on the print format.

If auto search is successfully completed, measurement is begun. Whenever the RETURN key is pressed, the (IMD?) request command is issued, and measurement results are read and displayed on the CRT. The data read terminate conditions depend on the print format.

If auto search fails, the program is aborted.

Table 15-7 shows methods for determining the measurement result print format and the whether one-second data will be printed.

Table 15-7

Paragraph	Data print format	Data output format (Standard or extended)	Print one-second data
(a)	Standard	Standard	As selected
(b)	Standard	Extended: data including threshold and error performance data	No
(c)	Standard	Extended: data including threshold measurement data	No
(d)	Standard	Extended: data including error performance measurement data	No
(e)	Short (abridged)	Standard	No

Note 1: READ command to read measurement results

When detecting end of measurement data on COMPAQ computers with device function mode CALL IBRD (ED%, rd$), it may not be possible to read data because the computer issues untalk and unlisten commands when it detects the measurement terminator. To prevent this, the following adapter function mode should be used to read measurement results via a COMPAQ computer.

< READ command in adapter function mode >

```
100  CMD$="?"+chr$(42)+"A"
110  CALL IBCMD(GPIB0%,CMD$)
120  rd$=SPACE$(○○)
130  CALL IBRD(GPIB0%,rd$)
140  IF IBSTA% <0 THEN GOTO trap
```

100 ? : Unlisten (This command specifies the MP1702A/MP1756A/MP1609A/MP1651A for unlisten mode.)

chr$(42) : Listener address (Corresponds to the primary address of IBCONF adapter (GPIB0) that is set to 10. Change the listener address to match the primary address.)

A : Talker address (Corresponds to the primary address of IBCONF device (ED) and the address of MP1702A/MP1756A/MP1609A/MP1651A, both set to 1. Change the talker address to match the primary address and the MP1702A/MP1756A/MP1609A/MP1651A address.)

110 The commands on line 100 are sent to the (GPIB0) adapter and specify the talker and listener.

120 Variable "rd$" is initialized with spaces.

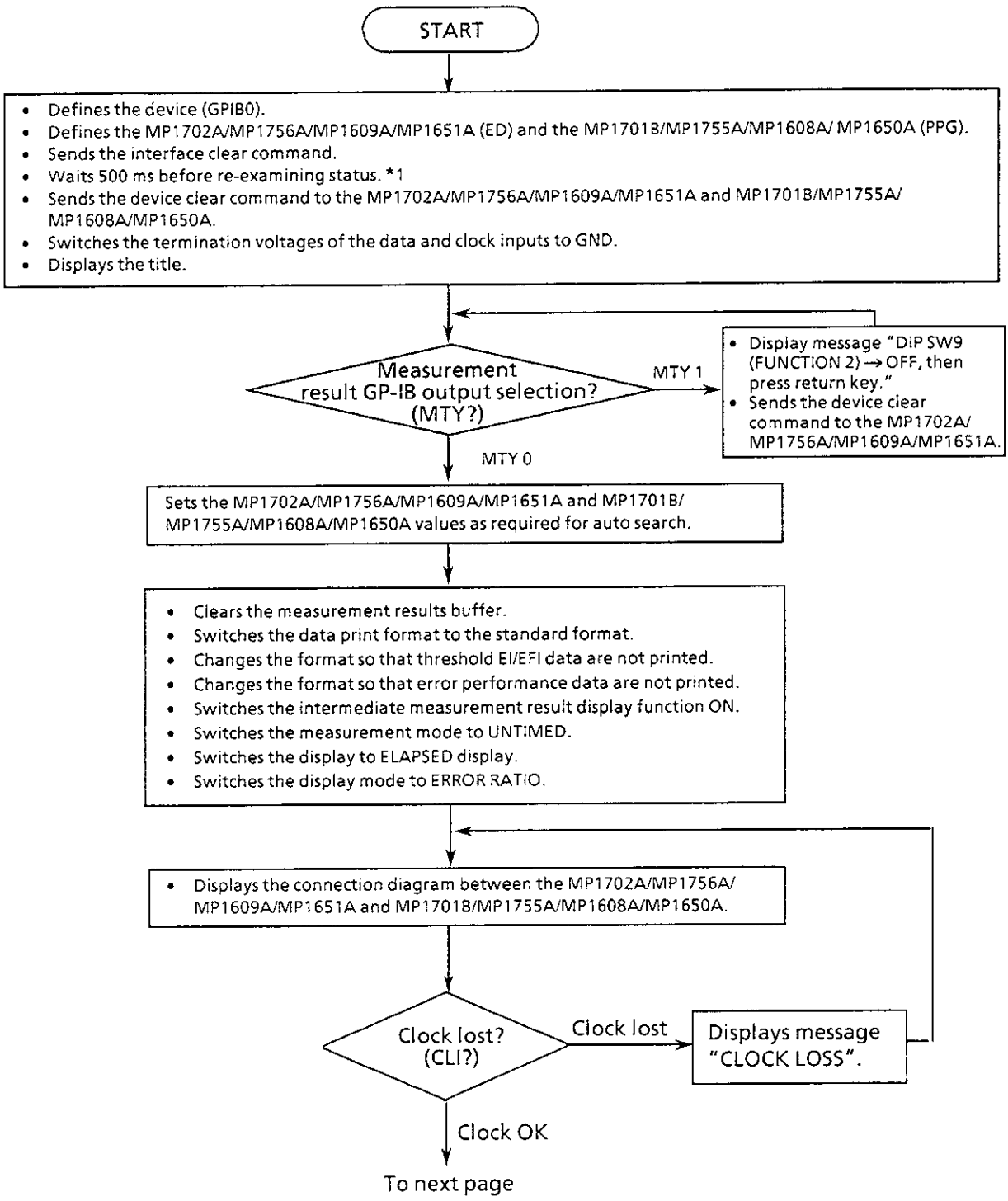
130 The contents read from the talker (MP1702A/MP1756A/MP1609A/MP1651A) by the adapter are stored in rd$.

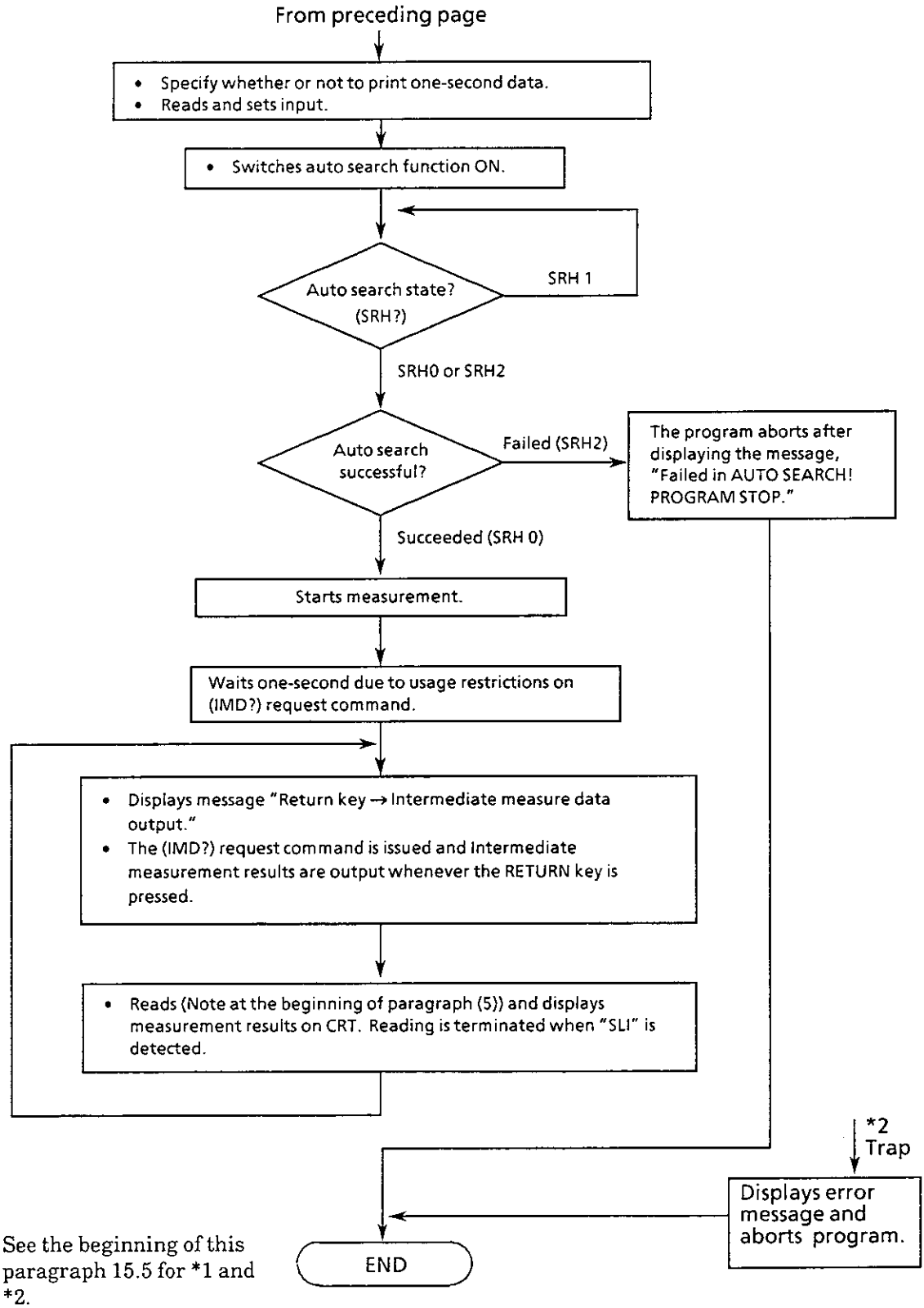
140 Checks whether the GP-IB commands were executed correctly.

Note 2: When using the MP1755A for the transmitter, internal clock frequency setting is ineffective. Set the frequency of the external clock source to 3000 MHz in advance.

(a) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Standard data are output in standard format; intermediate measurement results are output in response to (IMD?) request command.





Note: See the beginning of this paragraph 15.5 for *1 and *2.

Program listing

```
'************************************************************
'*                                                          *
'*     MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT     *
'*          -- Standard format , Standard output--          *
'*              * Intermediate measure data *               *
'*                                                ED2_M1    *
'************************************************************
'
'------------------------------------------------------------
'                          MAIN  ROUTINE                     '
'------------------------------------------------------------
'
common shared IBSTA%,IBERR%,IBCNT%   ' Setup GPIB-PC functions
GOSUB gpinit                         ' Setup GPIB interface
wrt$="DTM 0;CTM 0" : GOSUB wrtcmd2
'
SCREEN 9
PRINT "** MP1702A/MP1609A/MP1651A **"
PRINT "** MEASUREMENT SAMPLE SOFT **"
PRINT
'
GOSUB  MType                         ' Check Measurement type
'
GOSUB DSet                           ' Data setting
GOSUB Mode                           ' Measurement mode
GOSUB Clock                          ' Check Clock loss ?
GOSUB OneSec                         ' 1 sec dataprint ?
GOSUB Srch                           ' Auto search : ON
'
wrt$="STA"  : GOSUB wrtcmd2          ' Measurement Start
'
GOSUB Result                         ' Display Result
'
STOP
'
'------------------------------------------------------------
'                          SUB ROUTINE                       '
'------------------------------------------------------------
    '
MType : '---------- Check Measurement type ----------
    '
    DO
        wrt$="MTY?" : GOSUB wrtcmd2 : GOSUB readcmd
        '
        IF MID$(rd$,1,5)="MTY 1" THEN
          LOCATE 23,1
          INPUT "DIP SW 9 (Function 2)-->OFF, then press Enter key",A
          CALL IBCLR(ED%)
          '
        END IF
    LOOP UNTIL MID$(rd$,1,5)="MTY 0"
    '
    RETURN
    '
DSet: '---------- Data Setting ----------
    '
    wrt$= "LGC 0;PTN 2;MRK 1"     : GOSUB wrtcmd1
    wrt$= "LGC 0;PTN 2;MRK 1"     : GOSUB wrtcmd2
    '
```

```
    wrt$= "CLK 1;RES 1;FRQ 3000" : GOSUB wrtcmd1
    wrt$= "OFS 0;SPD 0"          : GOSUB wrtcmd1
    wrt$= "DAF 1;DOS 1"          : GOSUB wrtcmd1
    wrt$= "CDL 100;CAP 1;COS 1"  : GOSUB wrtcmd1
    wrt$= "EAD 1"                : GOSUB wrtcmd1
    '
    wrt$="CPL 1"                 : GOSUB wrtcmd2
    wrt$="SRH 0"                 : GOSUB wrtcmd2
    wrt$="SYN 1"                 : GOSUB wrtcmd2
    '
    RETURN
    '
Mode: '---------- Measurement mode ----------
    '
    wrt$="BCL"   : GOSUB wrtcmd2  ' Measurement buffer clear
    wrt$="FMT 0" : GOSUB wrtcmd2  ' Data output : Standard format
    wrt$="THR 0" : GOSUB wrtcmd2  ' Threshold output : OFF
    wrt$="EPF 0" : GOSUB wrtcmd2  ' Error performance output : OFF
    '
    wrt$="CUR 1" : GOSUB wrtcmd2  ' Current data : ON
    wrt$="MOD 2" : GOSUB wrtcmd2  ' Measurement mode : UNTIMED
    wrt$="TIM 4" : GOSUB wrtcmd2  ' Display ILAPSED
    wrt$="DMS 0" : GOSUB wrtcmd2  ' Display ERROR RATIO
    '
    RETURN
    '
Clock: '---------- Set data , Check connection ----------
    '
    DO
        GOSUB Connect             ' Display device connection
        '
        wrt$="CLI?" : GOSUB wrtcmd2 : GOSUB readcmd
        '
        IF MID$(rd$,1,5)="CLI 1" THEN PRINT "** CLOCK LOSS **"
        '
    LOOP UNTIL MID$(rd$,1,5)="CLI 0"
    '
    RETURN
    '
OneSec: '---------- One second data output? ----------
    '
    INPUT "1 sec data output [ OFF=0, ON=1 ]";osc$
    wrt$="OSC "+osc$ : GOSUB wrtcmd2
    '
    RETURN
    '
Srch: '---------- Auto search ON ----------
    '
    wrt$="SRH 1" : GOSUB wrtcmd2      ' Set Auto search : ON
    '
    DO
        wrt$="SRH?" : GOSUB wrtcmd2 : GOSUB readcmd
    LOOP UNTIL MID$(rd$,1,5)="SRH 0" or MID$(rd$,1,5)="SRH 2"
    '
    IF MID$(rd$,1,5)="SRH 2"  THEN
        PRINT "<<< Failed in AUTO SEARCH ! >>>"
        PRINT "     ** PROGRAM STOP  **"
        STOP
    END IF
    '
    RETURN
    '
```

```
Result: '---------- Display Result ----------
        '
        tim=1 : GOSUB waidly
        '
        DO
            INPUT "Enter -->Intermediate measure data",A
            wrt$="IMD?" : GOSUB wrtcmd2
            '
            DO
                GOSUB readcmd : PRINT rd$
                '
            LOOP UNTIL MID$(rd$,1,3)="SLI"
            '
        LOOP
        '
        RETURN
        '
gpinit: ' ---------- Set up GP-IB functions  ----------
        '
        CALL IBFIND("GPIB0", GPIB0%)          ' Open device (GPIB0)
        IF GPIB0% < 0 THEN GOTO trap          ' system error
        '
        CALL IBFIND("PPG", PPG%)              ' Open device(PPG)
        IF PPG% < 0 THEN GOTO trap            ' system error
        '
        CALL IBFIND("ED", ED%)                ' Open device(ED)
        IF ED% < 0 THEN GOTO trap             ' system error
        '
        CALL IBSIC(GPIB0%)                    ' Interface clear
        IF IBSTA% < 0 THEN GOTO trap          ' system error
        '
        tim = 0.5
        GOSUB waidly
        '
        CALL IBCLR(PPG%)                      ' Device clear(PPG)
        '
        CALL IBCLR(ED%)                       ' Device clear(ED)
        '
        RETURN
        '
wrtcmd1: ' ----------  Write command (PPG) ----------
        '
        wrt$=wrt$+chr$(13)+chr$(10)
        CALL IBWRT(PPG%, wrt$)                ' Write command (PPG)
        IF IBSTA% < 0 THEN GOTO trap          ' Trap
        '
        RETURN
        '
wrtcmd2: ' ----------  Write command (ED) ----------
        '
        wrt$=wrt$+chr$(13)+chr$(10)
        CALL IBWRT(ED%, wrt$)                 ' Write command (ED)
        IF IBSTA% < 0 THEN GOTO trap          ' Trap
        '
        RETURN
        '
readcmd: ' ---------- Read command (ED) ----------
        '
        CMD$="?"+chr$(42)+"A"
        CALL IBCMD(GPIB0%,CMD$)
        '
        rd$=SPACE$(28)
        CALL IBRD(GPIB0%, rd$)                ' Read command (GPIB0)
        IF IBSTA% < 0 THEN GOTO trap          ' Trap
        '
```

```
    RETURN
          '
waidly:   ' ----------  Wait delay  ----------
          '
    stm = TIMER
    etm = TIMER
    WHILE etm - stm < tim
        etm = TIMER
        IF etm < stm THEN etm = etm + 86400
    WEND
    '
    RETURN
          '
trap:     ' ----------  System trap  ----------
          '
    PRINT "IBERR%:" + STR$(IBERR%)
     STOP
          '
Connect:  '---------- display Connection ----------
          '
    WINDOW (-600,-500) -(600,500)
    '
    LINE (-50,-50) - (600,500),14,B
    LINE (10,100) - STEP(235,200),,B
    LINE (300,100) - STEP(235,200),,B
    '
    CIRCLE (130,130),8
    LINE (130,125) - STEP(0,-55)
    LINE (130,70) - STEP(220,0)
    LINE (350,70) - STEP(0,55)
    CIRCLE (350,130),8
    '
    CIRCLE (190,130),8
    LINE (190,125) - STEP(0,-100)
    LINE (190,25) - STEP(220,0)
    LINE (410,25) - STEP(0,100)
    CIRCLE (410,130),8
    '
    LOCATE 2,50: PRINT "<<  CONNECTION  >>"
    '
    LOCATE 4,42: PRINT "MP1701B/MP1608A    MP1702A/MP1609A"
    LOCATE 5,42: PRINT "    /MP1650A           /MP1651A"
    '
    LOCATE 9,45: PRINT "DATA  CLOCK1"
    LOCATE 9,62: PRINT "DATA  CLOCK"
    '
    LOCATE 23,1
    INPUT "   Aer you ready ?  Press return key to start    ",A
    '
    LOCATE 23,1
    PRINT "                                                  "
    '
    LOCATE 4,1
    '
    RETURN
    '
    END
```

EXECUTED RESULT

① When one-second data is not specified for printing

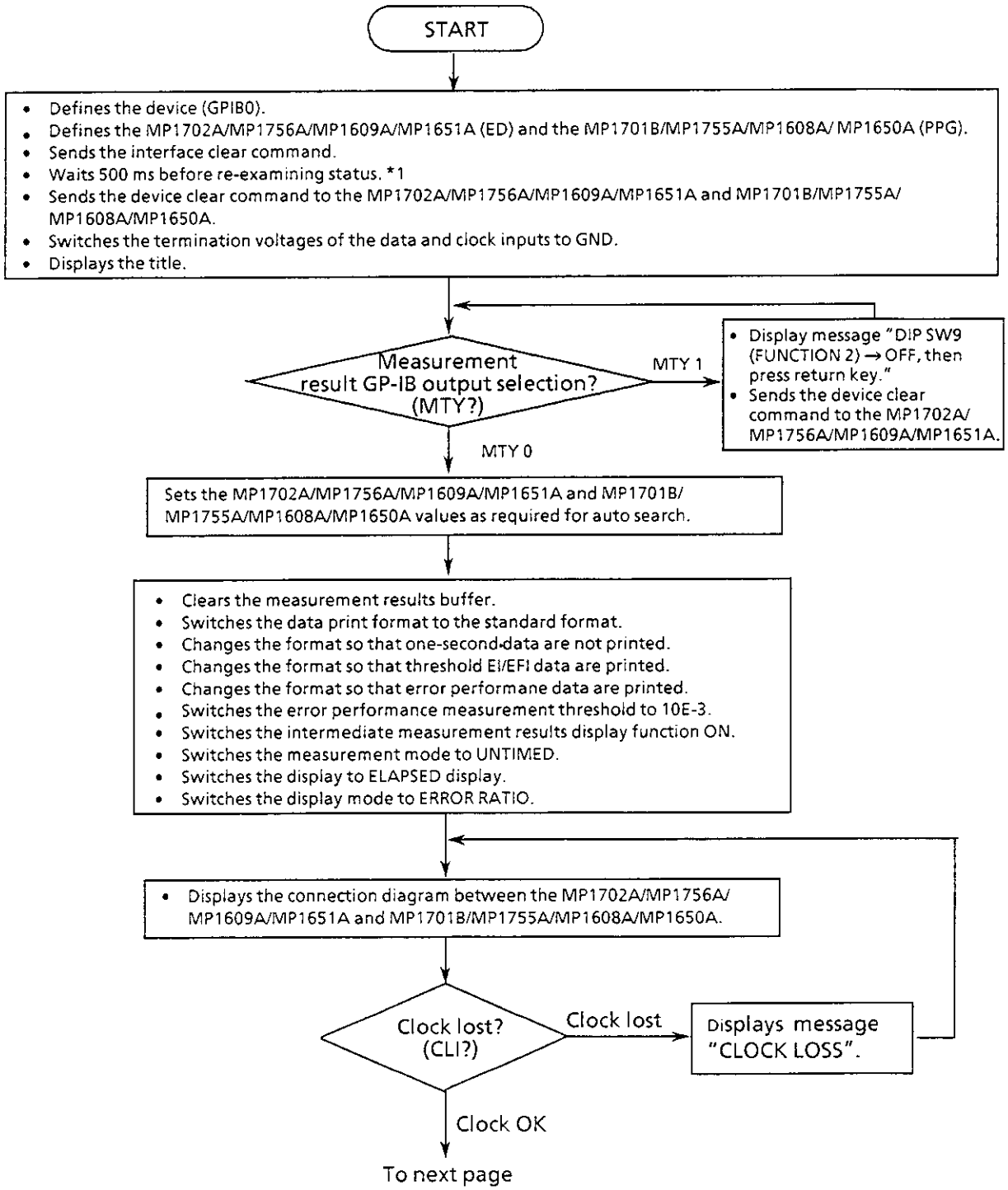
```
STA 90,02,28,16,02,37
STA 90,02,28,16,02,37
INT 90,02,28,16,02,40
ELP      0,00,00,03
ER  1.0000E-04
EC      1500004
EI            3
EFI      0.0000
PFI           0
CLI           0
SLI           0
STA 90,02,28,16,02,37
INT 90,02,28,16,02,47
ELP      0,00,00,10
ER  1.0000E-04
EC      5000012
EI           10
EFI      0.0000
PFI           0
CLI           0
SLI           0
```

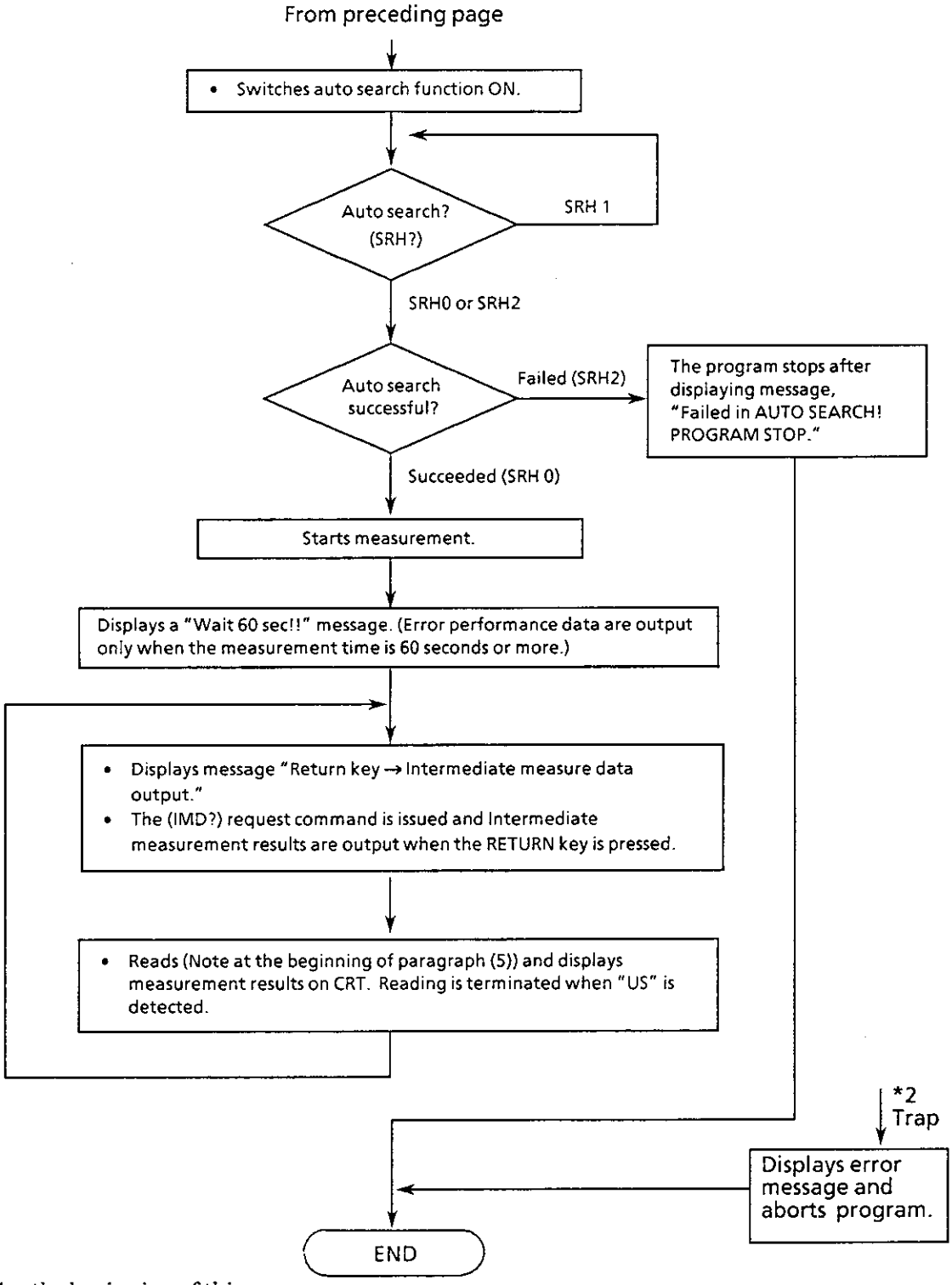
② When one-second data is specified for printing

```
STA 90,02,28,16,27,05
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500002
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
STA 90,02,28,16,27,05
INT 90,02,28,16,27,12
ELP      0,00,00,07
ER  1.0000E-04
EC      3500008
EI            7
EFI      0.0000
PFI           0
CLI           0
SLI           0
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500002
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500002
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500002
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
STA 90,02,28,16,27,05
INT 90,02,28,16,27,25
ELP      0,00,00,07
ER  1.0000E-04
EC      3500008
EI            7
EFI      0.0000
PFI           0
CLI           0
SLI           0
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500002
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500002
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
OSD 1.0000E-04,     500001
STA 90,02,28,16,27,05
INT 90,02,28,16,27,25
ELP      0,00,00,20
ER  1.0000E-04
EC     10000023
EI           20
EFI      0.0000
PFI           0
CLI           0
SLI           0
```

(b) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Extended data (threshold and error performance measurement data) are output in standard format; intermediate measurement results are output when the (IMD?) request command is issued.





Note: See the beginning of this paragraph 15.5 for *1 and *2.

Program listing

```
' ************************************************************
' *                                                          *
' *     MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT     *
' *        -- Standard format , Extension output --          *
' *       -- Threshold EI/EFI , Error performance --         *
' *             * Intermediate measure data *                *
' *                                                 ED2_M2   *
' ************************************************************
'
'-------------------------------------------------------------'
'                          MAIN  ROUTINE                      '
'-------------------------------------------------------------'
'
common shared IBSTA%,IBERR%,IBCNT%   ' Setup GPIB-PC functions
GOSUB gpinit                         ' Setup GPIB interface
wrt$="DTM 0;CTM 0" : GOSUB wrtcmd2
'
SCREEN 9
PRINT "** MP1702A/MP1609A/MP1651A **"
PRINT "** MEASUREMENT SAMPLE SOFT **"
PRINT
'
GOSUB  MType                         ' Check Measurement type
'
GOSUB DSet                           ' Data setting
GOSUB Mode                           ' Measurement mode
GOSUB Clock                          ' Check Clock loss ?
GOSUB Srch                           ' Auto search : ON
'
wrt$="STA"   : GOSUB wrtcmd2         ' Measurement Start
'
GOSUB Result                         ' Display Result
'
STOP
'
'-------------------------------------------------------------'
'                          SUB ROUTINE                        '
'-------------------------------------------------------------'
'
MType :  '---------- Check Measurement type ----------
'
    DO
        wrt$="MTY?" : GOSUB wrtcmd2 : GOSUB readcmd
        '
        IF MID$(rd$,1,5)="MTY 1" THEN
          LOCATE 23,1
          INPUT "DIP SW 9 (Function 2)-->OFF, then press Enter key",A
          CALL IBCLR(ED%)
          '
        END IF
    LOOP UNTIL MID$(rd$,1,5)="MTY 0"
    '
    RETURN
'
DSet:  '---------- Data Setting ----------
'
    wrt$= "LGC 0;PTN 2;MRK 1"     : GOSUB wrtcmd1
    wrt$= "LGC 0;PTN 2;MRK 1"     : GOSUB wrtcmd2
    '
```

```
    wrt$= "CLK 1;RES 1;FRQ 3000" : GOSUB wrtcmd1
    wrt$= "OFS 0;SPD 0"          : GOSUB wrtcmd1
    wrt$= "DAP 1;DOS 1"          : GOSUB wrtcmd1
    wrt$= "CDL 100;CAP 1;COS 1"  : GOSUB wrtcmd1
    wrt$= "EAD 1"                : GOSUB wrtcmd1
    '
    wrt$="CPL 0"                 : GOSUB wrtcmd2
    wrt$="SRH 0"                 : GOSUB wrtcmd2
    wrt$="SYN 1"                 : GOSUB wrtcmd2
    '
    RETURN
    '
Mode: '---------- Measurement mode ----------
    '
    wrt$="BCL"    : GOSUB wrtcmd2  ' Measurement buffer clear
    wrt$="FMT 0" : GOSUB wrtcmd2  ' Data output : Standard format
    wrt$="OSC 0" : GOSUB wrtcmd2  ' 1 sec data output : OFF
    wrt$="THR 1" : GOSUB wrtcmd2  ' Threshold output : ON
    wrt$="EPF 1" : GOSUB wrtcmd2  ' Error performance output : ON
    wrt$="ETH 0" : GOSUB wrtcmd2  ' Error performance threshold : 10E-3
    '
    wrt$="CUR 1" : GOSUB wrtcmd2  ' Current data : ON
    wrt$="MOD 2" : GOSUB wrtcmd2  ' Measurement mode : UNTIMED
    wrt$="TIM 4" : GOSUB wrtcmd2  ' Display ILAPSED
    wrt$="DMS 0" : GOSUB wrtcmd2  ' Display ERROR RATIO
    '
    RETURN
    '
Clock: '---------- Set data , Check connection ----------
    '
    DO
        GOSUB Connect              ' Display device connection
        '
        wrt$="CLI?" : GOSUB wrtcmd2 : GOSUB readcmd
        '
        IF MID$(rd$,1,5)="CLI 1" THEN PRINT "** CLOCK LOSS **"
        '
    LOOP UNTIL MID$(rd$,1,5)="CLI 0"
    '
    RETURN
    '
Srch: '---------- Auto search ON ----------
    '
    wrt$="SRH 1" : GOSUB wrtcmd2       ' Set Auto search : ON
    '
    DO
        wrt$="SRH?" : GOSUB wrtcmd2 : GOSUB readcmd
    LOOP UNTIL MID$(rd$,1,5)="SRH 0" or MID$(rd$,1,5)="SRH 2"
    '
    IF MID$(rd$,1,5)="SRH 2"  THEN
        PRINT "<<< Failed in AUTO SEARCH ! >>>"
        PRINT "     ** PROGRAM STOP  **"
        STOP
    END IF
    '
    RETURN
    '
Result: '---------- Display Result ----------
    '
    PRINT "* Wait 60 sec !! "
    PRINT "   To output Error performance data"
    DO
        INPUT "Enter --> Intermediate measure data",A
        wrt$="IMD?" : GOSUB wrtcmd2
```

```
        '
        DO
            GOSUB readcmd : PRINT rd$
            '
        LOOP UNTIL MID$(rd$,1,2)="US"
        '
    LOOP
    '
    RETURN
        '
gpinit: ' ---------- Set up GP-IB functions  ----------
        '
    CALL IBFIND("GPIB0", GPIB0%)          ' Open device (GPIB0)
    IF GPIB0% < 0 THEN GOTO trap          ' system error
    '
    CALL IBFIND("PPG", PPG%)              ' Open device(PPG)
    IF PPG% < 0 THEN GOTO trap            ' system error
    '
    CALL IBFIND("ED", ED%)                ' Open device(ED)
    IF ED% < 0 THEN GOTO trap             ' system error
    '
    CALL IBSIC(GPIB0%)                    ' Interface clear
    IF IBSTA% < 0 THEN GOTO trap          ' system error
    '
    tim = 0.5
    GOSUB waidly
    '
    CALL IBCLR(PPG%)                      ' Device clear(PPG)
    '
    CALL IBCLR(ED%)                       ' Device clear(ED)
    '
    RETURN
        '
wrtcmd1: ' ----------  Write command (PPG) ----------
        '
    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(PPG%, wrt$)                ' Write command (PPG)
    IF IBSTA% < 0 THEN GOTO trap          ' Trap
    '
    RETURN
        '
wrtcmd2: ' ----------  Write command (ED) ----------
        '
    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(ED%, wrt$)                 ' Write command (ED)
    IF IBSTA% < 0 THEN GOTO trap          ' Trap
    '
    RETURN
        '
readcmd: ' ---------- Read command (ED) ----------
        '
    CMD$="?"+chr$(42)+"A"
    CALL IBCMD(GPIB0%,CMD$)
    '
    rd$=SPACE$(25)
    CALL IBRD(GPIB0%, rd$)                ' Read command (GPIB0)
    IF IBSTA% < 0 THEN GOTO trap          ' Trap
    '
    RETURN
        '
```

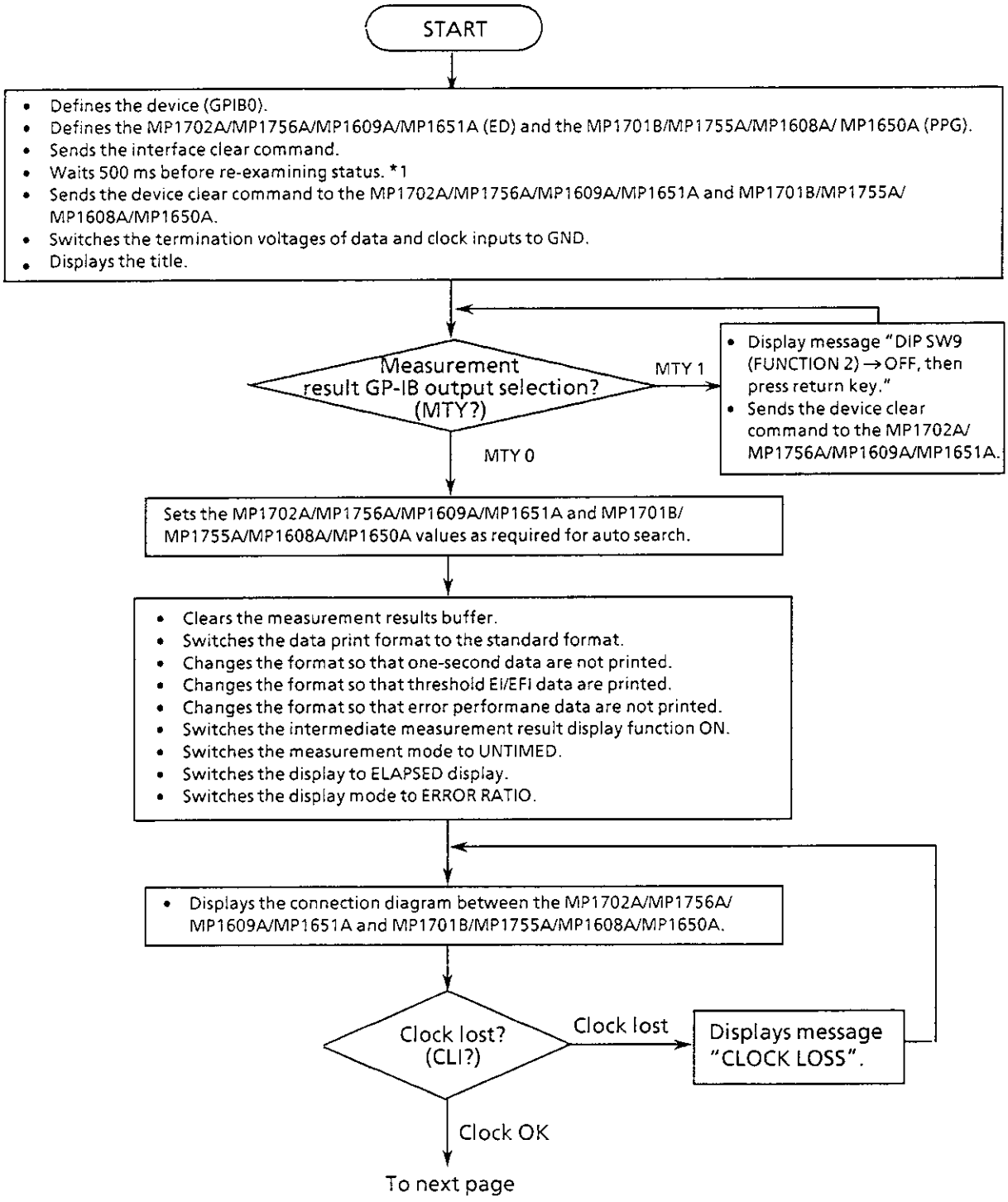
```
waidly:  ' ----------  Wait delay  ----------
     '
     stm = TIMER
     etm = TIMER
     WHILE etm - stm < tim
         etm = TIMER
         IF etm < stm THEN etm = etm + 86400
     WEND
     '
     RETURN
     '
trap:    ' ----------  System trap  ----------
     '
     PRINT "IBERR%:" + STR$(IBERR%)
      STOP
     '
Connect: '---------- display Connection ----------
     '
     WINDOW (-600,-500) -(600,500)
     '
     LINE (-50,-50) - (600,500),14,B
     LINE (10,100) - STEP(235,200),,B
     LINE (300,100) - STEP(235,200),,B
     '
     CIRCLE (130,130),8
     LINE (130,125) - STEP(0,-55)
     LINE (130,70) - STEP(220,0)
     LINE (350,70) - STEP(0,55)
     CIRCLE (350,130),8
     '
     CIRCLE (190,130),8
     LINE (190,125) - STEP(0,-100)
     LINE (190,25) - STEP(220,0)
     LINE (410,25) - STEP(0,100)
     CIRCLE (410,130),8
     '
     LOCATE 2,50: PRINT "<<  CONNECTION  >>"
     '
     LOCATE 4,42: PRINT "MP1701B/MP1608A     MP1702A/MP1609A"
     LOCATE 5,42: PRINT "     /MP1650A            /MP1651A"
     '
     LOCATE 9,45: PRINT "DATA  CLOCK1"
     LOCATE 9,62: PRINT "DATA  CLOCK"
     '
     LOCATE 23,1
     INPUT "   Aer you ready ?  Press return key to start     ",A
     '
     LOCATE 23,1
     PRINT "                                                              "
     '
     LOCATE 4,1
     '
     RETURN
     '
     END
```

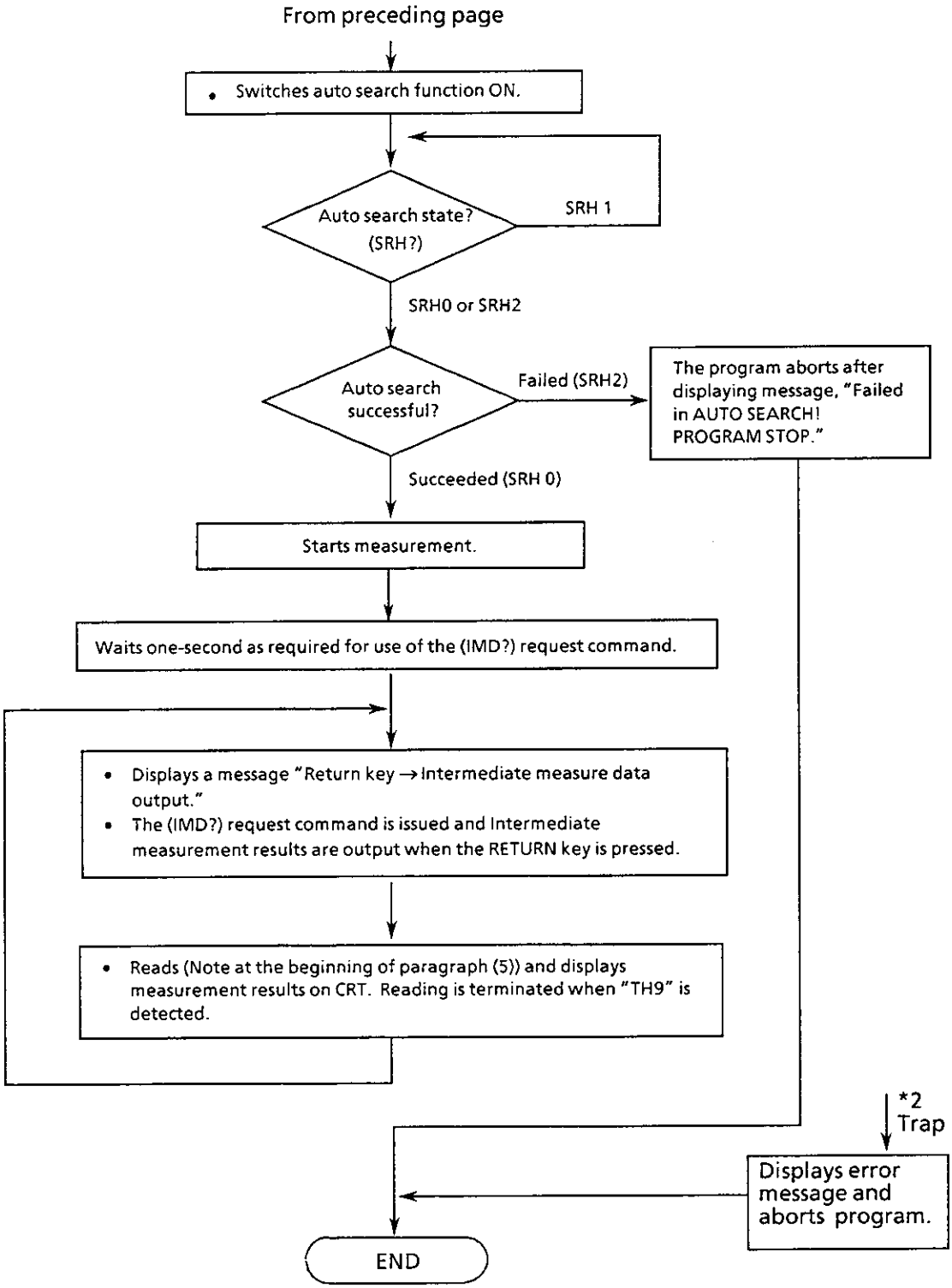
EXECUTED RESULT

```
STA 90,02,28,16,07,34
STA 90,02,28,16,07,34
INT 90,02,28,16,08,39
ELP      0,00,01,05
ER  1.0000E-04
EC     32500077
EI           65
EFI      0.0000
PFI           0
CLI           0
SLI           0
TH3           0,100.0000
TH4          65,   0.0000
TH5          65,   0.0000
TH6          65,   0.0000
TH7          65,   0.0000
TH8          65,   0.0000
TH9           0,100.0000
ES     100.0000
EFS      0.0000
SES      0.0000
DM     100.0000
US       0.0000
STA 90,02,28,16,07,34
INT 90,02,28,16,08,54
ELP      0,00,01,20
ER  1.0000E-04
EC     40000094
EI           80
EFI      0.0000
PFI           0
CLI           0
SLI           0
TH3           0,100.0000
TH4          80,   0.0000
TH5          80,   0.0000
TH6          80,   0.0000
TH7          80,   0.0000
TH8          80,   0.0000
TH9           0,100.0000
ES     100.0000
EFS      0.0000
SES      0.0000
DM     100.0000
US       0.0000
```

(c) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Extended data (threshold measurement data) are output in standard format; intermediate measurement result are output in response to an (IMD?) request command.





Note: See the beginning of this paragraph 15.5 for *1 and *2.

Program listing

```
' ************************************************************
' *                                                          *
' *     MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT     *
' *        -- Standard format : Extension output --          *
' *                  -- Threshold EI/EFI --                  *
' *              * Intermediate measure data *               *
' *                                                 ED2_M3   *
' ************************************************************
'
'-------------------------------------------------------------'
'                          MAIN  ROUTINE                      '
'-------------------------------------------------------------'
'
common shared IBSTA%,IBERR%,IBCNT%   ' Setup GPIB-PC functions
GOSUB gpinit                         ' Setup GPIB interface
wrt$="DTM 0;CTM 0" : GOSUB wrtcmd2
'
SCREEN 9
PRINT "** MP1702A/MP1609A/MP1651A **"
PRINT "** MEASUREMENT SAMPLE SOFT **"
PRINT
'
GOSUB  MType                         ' Check Measurement type
'
GOSUB DSet                           ' Data setting
GOSUB Mode                           ' Measurement mode
GOSUB Clock                          ' Check Clock loss ?
GOSUB Srch                           ' Auto search : ON
'
wrt$="STA"  : GOSUB wrtcmd2          ' Measurement Start
'
GOSUB Result                         ' Display Result
'
STOP
'
'-------------------------------------------------------------'
'                          SUB ROUTINE                        '
'-------------------------------------------------------------'
    '
MType : '---------- Check Measurement type ----------
    '
    DO
        wrt$="MTY?" : GOSUB wrtcmd2 : GOSUB readcmd
        '
        IF MID$(rd$,1,5)="MTY 1" THEN
          LOCATE 23,1
          INPUT "DIP SW 9 (Function 2)-->OFF, then press Enter key",A
          CALL IBCLR(ED%)
          '
        END IF
    LOOP UNTIL MID$(rd$,1,5)="MTY 0"
    '
    RETURN
    '
DSet: '---------- Data Setting ----------
    '
    wrt$= "LGC 0;PTN 2;MRK 1"     : GOSUB wrtcmd1
    wrt$= "LGC 0;PTN 2;MRK 1"     : GOSUB wrtcmd2
    '
```

```
        wrt$= "CLK 1;RES 1;FRQ 3000" : GOSUB wrtcmd1
        wrt$= "OFS 0;SPD 0"          : GOSUB wrtcmd1
        wrt$= "DAP 1;DOS 1"          : GOSUB wrtcmd1
        wrt$= "CDL 100;CAP 1;COS 1"  : GOSUB wrtcmd1
        wrt$= "EAD 1"                : GOSUB wrtcmd1
        '
        wrt$="CPL 0"                 : GOSUB wrtcmd2
        wrt$="SRH 0"                 : GOSUB wrtcmd2
        wrt$="SYN 1"                 : GOSUB wrtcmd2
        '
        RETURN
        '
Mode:   '---------- Measurement mode ----------
        '
        wrt$="BCL"    : GOSUB wrtcmd2   ' Measurement buffer clear
        wrt$="FMT 0" : GOSUB wrtcmd2    ' Data output : Standard format
        wrt$="OSC 0" : GOSUB wrtcmd2    ' 1 sec data output : OFF
        wrt$="THR 1" : GOSUB wrtcmd2    ' Threshold output : ON
        wrt$="EPF 0" : GOSUB wrtcmd2    ' Error performance output : OFF
        '
        wrt$="CUR 1" : GOSUB wrtcmd2    ' Current data : ON
        wrt$="MOD 2" : GOSUB wrtcmd2    ' Measurement mode : UNTIMED
        wrt$="TIM 4" : GOSUB wrtcmd2    ' Display ILAPSED
        wrt$="DMS 0" : GOSUB wrtcmd2    ' Display ERROR RATIO
        '
        RETURN
        '
Clock:  '---------- Set data , Check connection ----------
        '
        DO
            GOSUB Connect               ' Display device connection
            '
            wrt$="CLI?" : GOSUB wrtcmd2 : GOSUB readcmd
            '
            IF MID$(rd$,1,5)="CLI 1" THEN PRINT "** CLOCK LOSS **"
            '
        LOOP UNTIL MID$(rd$,1,5)="CLI 0"
        '
        RETURN
        '
Srch:   '---------- Auto search ON ----------
        '
        wrt$="SRH 1" : GOSUB wrtcmd2        ' Set Auto search : ON
        '
        DO
            wrt$="SRH?" : GOSUB wrtcmd2 : GOSUB readcmd
        LOOP UNTIL MID$(rd$,1,5)="SRH 0" or MID$(rd$,1,5)="SRH 2"
        '
        IF MID$(rd$,1,5)="SRH 2"  THEN
            PRINT "<<< Failed in AUTO SEARCH ! >>>"
            PRINT "     ** PROGRAM STOP  **"
            STOP
        END IF
        '
        RETURN
        '
Result: '---------- Display Result ----------
        '
        tim=1 : GOSUB waidly
        '
        DO
            INPUT "Enter --> Intermediate measure data",A
            wrt$="IMD?" : GOSUB wrtcmd2
            '
```

```
        DO
            GOSUB readcmd : PRINT rd$
        LOOP UNTIL MID$(rd$,1,3)="TH9"
    LOOP
    RETURN
gpinit: ' ---------- Set up GP-IB functions ----------
    CALL IBFIND("GPIB0", GPIB0%)          ' Open device (GPIB0)
    IF GPIB0% < 0 THEN GOTO trap          ' system error
    CALL IBFIND("PPG", PPG%)              ' Open device(PPG)
    IF PPG% < 0 THEN GOTO trap            ' system error
    CALL IBFIND("ED", ED%)                ' Open device(ED)
    IF ED% < 0 THEN GOTO trap             ' system error
    CALL IBSIC(GPIB0%)                    ' Interface clear
    IF IBSTA% < 0 THEN GOTO trap          ' system error
    tim = 0.5
    GOSUB waidly
    CALL IBCLR(PPG%)                      ' Device clear(PPG)
    CALL IBCLR(ED%)                       ' Device clear(ED)
    RETURN
wrtcmd1: ' ----------  Write command (PPG) ----------
    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(PPG%, wrt$)                ' Write command (PPG)
    IF IBSTA% < 0 THEN GOTO trap          ' Trap
    RETURN
wrtcmd2: ' ----------  Write command (ED) ----------
    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(ED%, wrt$)                 ' Write command (ED)
    IF IBSTA% < 0 THEN GOTO trap          ' Trap
    RETURN
readcmd: ' ---------- Read command (ED) ----------
    CMD$="?"+chr$(42)+"A"
    CALL IBCMD(GPIB0%,CMD$)
    rd$=SPACE$(25)
    CALL IBRD(GPIB0%, rd$)                ' Read command (GPIB0)
    IF IBSTA% < 0 THEN GOTO trap          ' Trap
    RETURN
waidly: ' ----------  Wait delay  ----------
    stm = TIMER
    etm = TIMER
    WHILE etm - stm < tim
        etm = TIMER
        IF etm < stm THEN etm = etm + 86400
```

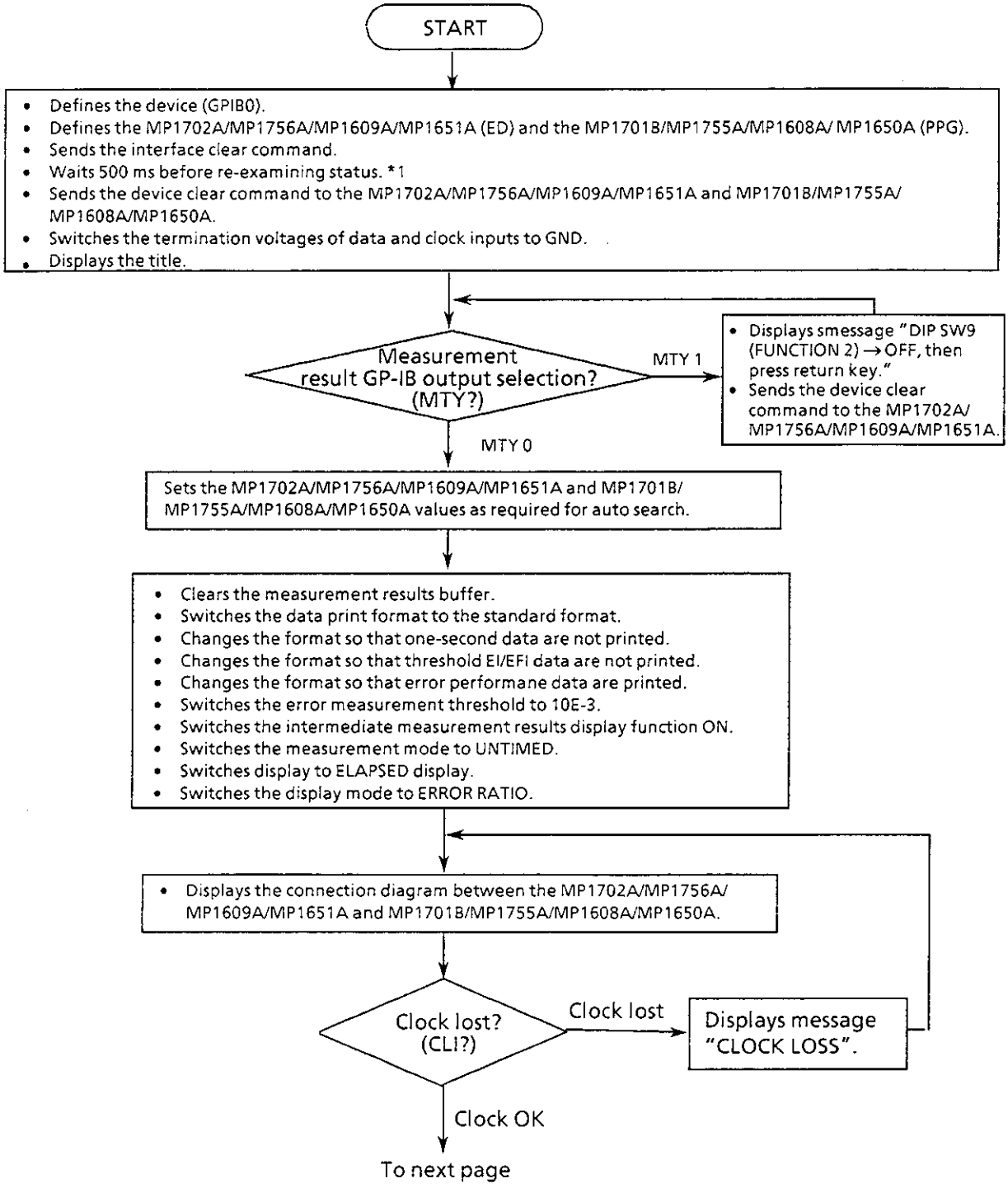
```
    WEND
    '
    RETURN
    '
trap:    ' ----------  System trap  ----------
    '
    PRINT "IBERR%:" + STR$(IBERR%)
     STOP
    '
Connect: '---------- display Connection ----------
    '
    WINDOW (-600,-500) -(600,500)
    '
    LINE (-50,-50) - (600,500),14,B
    LINE (10,100) - STEP(235,200),,B
    LINE (300,100) - STEP(235,200),,B
    '
    CIRCLE (130,130),8
    LINE (130,125) - STEP(0,-55)
    LINE (130,70) - STEP(220,0)
    LINE (350,70) - STEP(0,55)
    CIRCLE (350,130),8
    '
    CIRCLE (190,130),8
    LINE (190,125) - STEP(0,-100)
    LINE (190,25) - STEP(220,0)
    LINE (410,25) - STEP(0,100)
    CIRCLE (410,130),8
    '
    LOCATE 2,50: PRINT "<<  CONNECTION  >>"
    '
    LOCATE 4,42: PRINT "MP1701B/MP1608A    MP1702A/MP1609A"
    LOCATE 5,42: PRINT "    /MP1650A           /MP1651A"
    '
    LOCATE 9,45: PRINT "DATA  CLOCK1"
    LOCATE 9,62: PRINT "DATA  CLOCK"
    '
    LOCATE 23,1
    INPUT "   Aer you ready ?  Press return key to start    ",A
    '
    LOCATE 23,1
    PRINT "                                                  "
    '
    LOCATE 4,1
    '
    RETURN
    '
    END
```

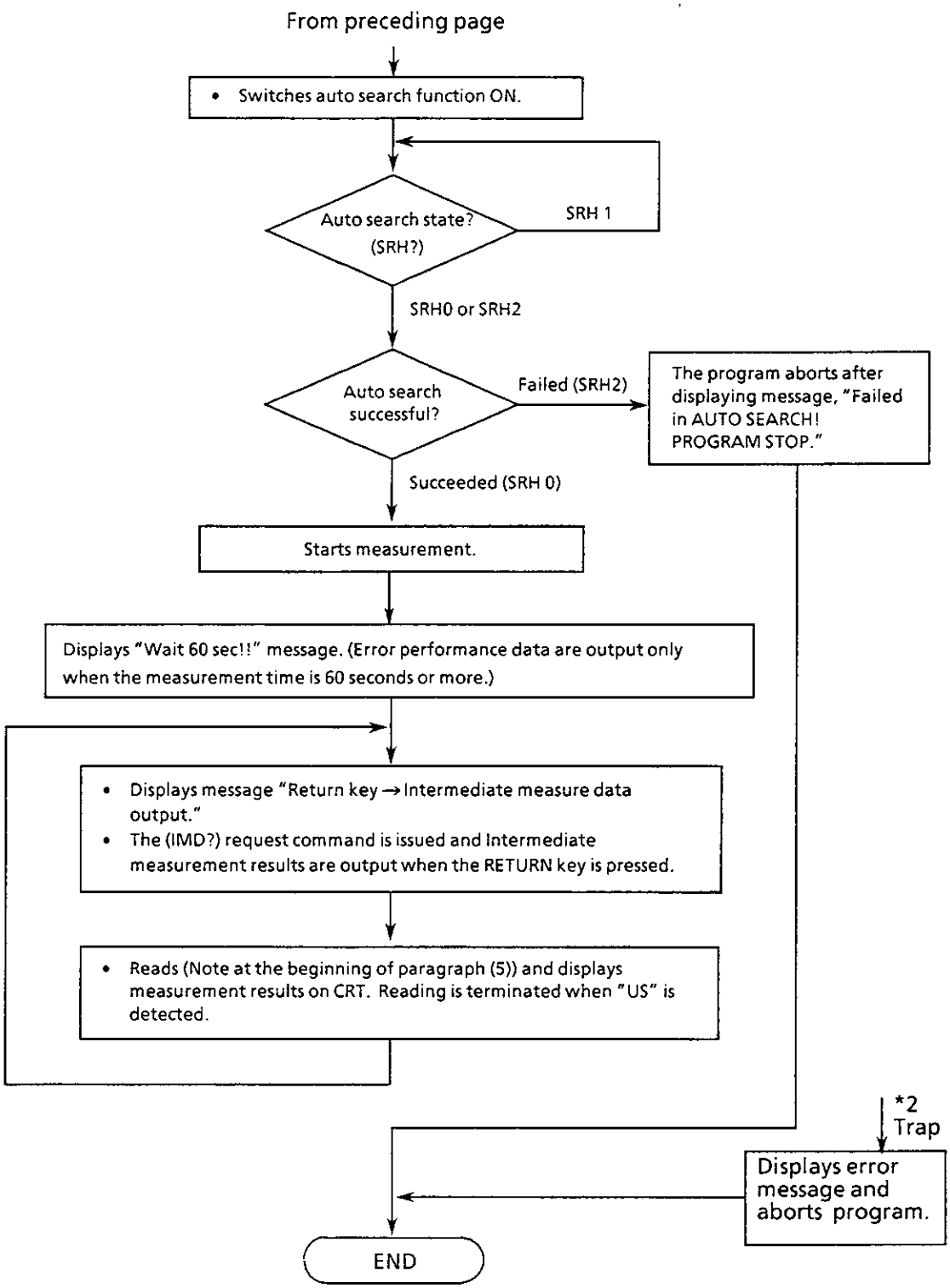
EXECUTED RESULT

```
STA 90,02,28,16,12,31
STA 90,02,28,16,12,31
INT 90,02,28,16,12,35
ELP      0,00,00,04
ER  1.0000E-04
EC      2000005
EI            4
EFI       0.0000
PFI           0
CLI           0
SLI           0
TH3           0,100.0000
TH4           4,   0.0000
TH5           4,   0.0000
TH6           4,   0.0000
TH7           4,   0.0000
TH8           4,   0.0000
TH9           0,100.0000
STA 90,02,28,16,12,31
INT 90,02,28,16,12,50
ELP      0,00,00,19
ER  1.0000E-04
EC      9500023
EI           19
EFI       0.0000
PFI           0
CLI           0
SLI           0
TH3           0,100.0000
TH4          19,   0.0000
TH5          19,   0.0000
TH6          19,   0.0000
TH7          19,   0.0000
TH8          19,   0.0000
TH9           0,100.0000
```

(d) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Extended data (error performance measurement data) are output in standard format; intermediate measurement results are output in response to the (IMD?) request command.





Note: See the beginning of this paragraph 15.5 for *1 and *2.

Program listing

```
' ************************************************************
' *                                                          *
' *     MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT     *
' *        -- Standard format , Extension output --          *
' *                  --  Error performance --                *
' *               * Intermediate measure data *              *
' *                                                 ED2_M4   *
' ************************************************************
'
'-------------------------------------------------------------
'                           MAIN  ROUTINE                     '
'-------------------------------------------------------------
'
common shared IBSTA%,IBERR%,IBCNT%   ' Setup GPIB-PC functions
GOSUB gpinit                         ' Setup GPIB interface
wrt$="DTM 0;CTM 0" : GOSUB wrtcmd2
'
SCREEN 9
PRINT "** MP1702A/MP1609A/MP1651A **"
PRINT "** MEASUREMENT SAMPLE SOFT ** "
PRINT
'
GOSUB  MType                         ' Check Measurement type
'
GOSUB DSet                           ' Data setting
GOSUB Mode                           ' MEasurement mode
GOSUB Clock                          ' Check Clock loss ?
GOSUB Srch                           ' Auto search : ON
'
wrt$="STA"  : GOSUB wrtcmd2          ' Measurement Start
'
GOSUB Result                         ' Display Result
'
STOP
'
'-------------------------------------------------------------
'                           SUB ROUTINE                       '
'-------------------------------------------------------------
'
MType :  '---------- Check Measurement type ----------
'
    DO
        wrt$="MTY?" : GOSUB wrtcmd2 : GOSUB readcmd
        '
        IF MID$(rd$,1,5)="MTY 1" THEN
          LOCATE 23,1
          INPUT "DIP SW 9 (Function 2)-->OFF, then press Enter key",A
          CALL IBCLR(ED%)
          '
        END IF
    LOOP UNTIL MID$(rd$,1,5)="MTY 0"
    '
    RETURN
    '
DSet:  '---------- Data Setting ----------
    '
    wrt$= "LGC 0;PTN 2;MRK 1"     : GOSUB wrtcmd1
    wrt$= "LGC 0;PTN 2;MRK 1"     : GOSUB wrtcmd2
    '
```

```
    wrt$= "CLK 1;RES 1;FRQ 3000" : GOSUB wrtcmd1
    wrt$= "OFS 0;SPD 0"          : GOSUB wrtcmd1
    wrt$= "DAP 1;DOS 1"          : GOSUB wrtcmd1
    wrt$= "CDL 100;CAP 1;COS 1"  : GOSUB wrtcmd1
    wrt$= "EAD 1"                : GOSUB wrtcmd1
    '
    wrt$="CPL 0"                 : GOSUB wrtcmd2
    wrt$="SRH 0"                 : GOSUB wrtcmd2
    wrt$="SYN 1"                 : GOSUB wrtcmd2
    '
    RETURN
    '
Mode: '---------- Measurement mode ----------
    '
    wrt$="BCL"    : GOSUB wrtcmd2  ' Measurement buffer clear
    wrt$="FMT 0" : GOSUB wrtcmd2  ' Data output : Standard format
    wrt$="OSC 0" : GOSUB wrtcmd2  ' 1 sec data output : OFF
    wrt$="THR 0" : GOSUB wrtcmd2  ' Threshold output : OFF
    wrt$="EPF 1" : GOSUB wrtcmd2  ' Error performance output : ON
    wrt$="ETH 0" : GOSUB wrtcmd2  ' Error performance threshold : 10E-3
    '
    wrt$="CUR 1" : GOSUB wrtcmd2  ' Current data : ON
    wrt$="MOD 2" : GOSUB wrtcmd2  ' Measurement mode : UNTIMEED
    wrt$="TIM 4" : GOSUB wrtcmd2  ' Display ILAPSED
    wrt$="DMS 0" : GOSUB wrtcmd2  ' Display ERROR RATIO
    '
    RETURN
    '
Clock: '---------- Set data , Check connection ----------
    '
    DO
        GOSUB Connect              ' Display device connection
        '
        wrt$="CLI?" : GOSUB wrtcmd2 : GOSUB readcmd
        '
        IF MID$(rd$,1,5)="CLI 1" THEN PRINT "** CLOCK LOSS **"
        '
    LOOP UNTIL MID$(rd$,1,5)="CLI 0"
    '
    RETURN
    '
Srch: '---------- Auto search ON ----------
    '
    wrt$="SRH 1" : GOSUB wrtcmd2       ' Set Auto search : ON
    '
    DO
        wrt$="SRH?" : GOSUB wrtcmd2 : GOSUB readcmd
    LOOP UNTIL MID$(rd$,1,5)="SRH 0" or MID$(rd$,1,5)="SRH 2"
    '
    IF MID$(rd$,1,5)="SRH 2"  THEN
        PRINT "<<< Failed in AUTO SEARCH ! >>>"
        PRINT "     ** PROGRAM STOP  **"
        STOP
    END IF
    '
    RETURN
    '
Result: '---------- Display Result ----------
    '
    PRINT "* Wait 60 sec !!"
    PRINT "    To output Error performance data"
    '
```

```
    DO
        INPUT "Enter --> Intermediate measure data",A
        wrt$="IMD?" : GOSUB wrtcmd2
        '
        DO
            GOSUB readcmd : PRINT rd$
            '
        LOOP UNTIL MID$(rd$,1,2)="US"
        '
    LOOP
    '
    RETURN
    '
gpinit: ' ---------- Set up GP-IB functions  ----------
    '
    CALL IBFIND("GPIB0", GPIB0%)          ' Open device (GPIB0)
    IF GPIB0% < 0 THEN GOTO trap          ' system error
    '
    CALL IBFIND("PPG", PPG%)              ' Open device(PPG)
    IF PPG% < 0 THEN GOTO trap            ' system error
    '
    CALL IBFIND("ED", ED%)                ' Open device(ED)
    IF ED% < 0 THEN GOTO trap             ' system error
    '
    CALL IBSIC(GPIB0%)                    ' Interface clear
    IF IBSTA% < 0 THEN GOTO trap          ' system error
    '
    tim = 0.5
    GOSUB waidly
    '
    CALL IBCLR(PPG%)                      ' Device clear(PPG)
    '
    CALL IBCLR(ED%)                       ' Device clear(ED)
    '
    RETURN
    '
wrtcmd1: ' ----------  Write command (PPG) ----------
    '
    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(PPG%, wrt$)                ' Write command (PPG)
    IF IBSTA% < 0 THEN GOTO trap          ' Trap
    '
    RETURN
    '
wrtcmd2: ' ----------  Write command (ED) ----------
    '
    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(ED%, wrt$)                 ' Write command (ED)
    IF IBSTA% < 0 THEN GOTO trap          ' Trap
    '
    RETURN
    '
readcmd: ' ---------- Read command (ED) ----------
    '
    CMD$="?"+chr$(42)+"A"
    CALL IBCMD(GPIB0%,CMD$)
    '
    rd$=SPACE$(23)
    CALL IBRD(GPIB0%, rd$)                ' Read command (GPIB0)
    IF IBSTA% < 0 THEN GOTO trap          ' Trap
    '
    RETURN
    '
```

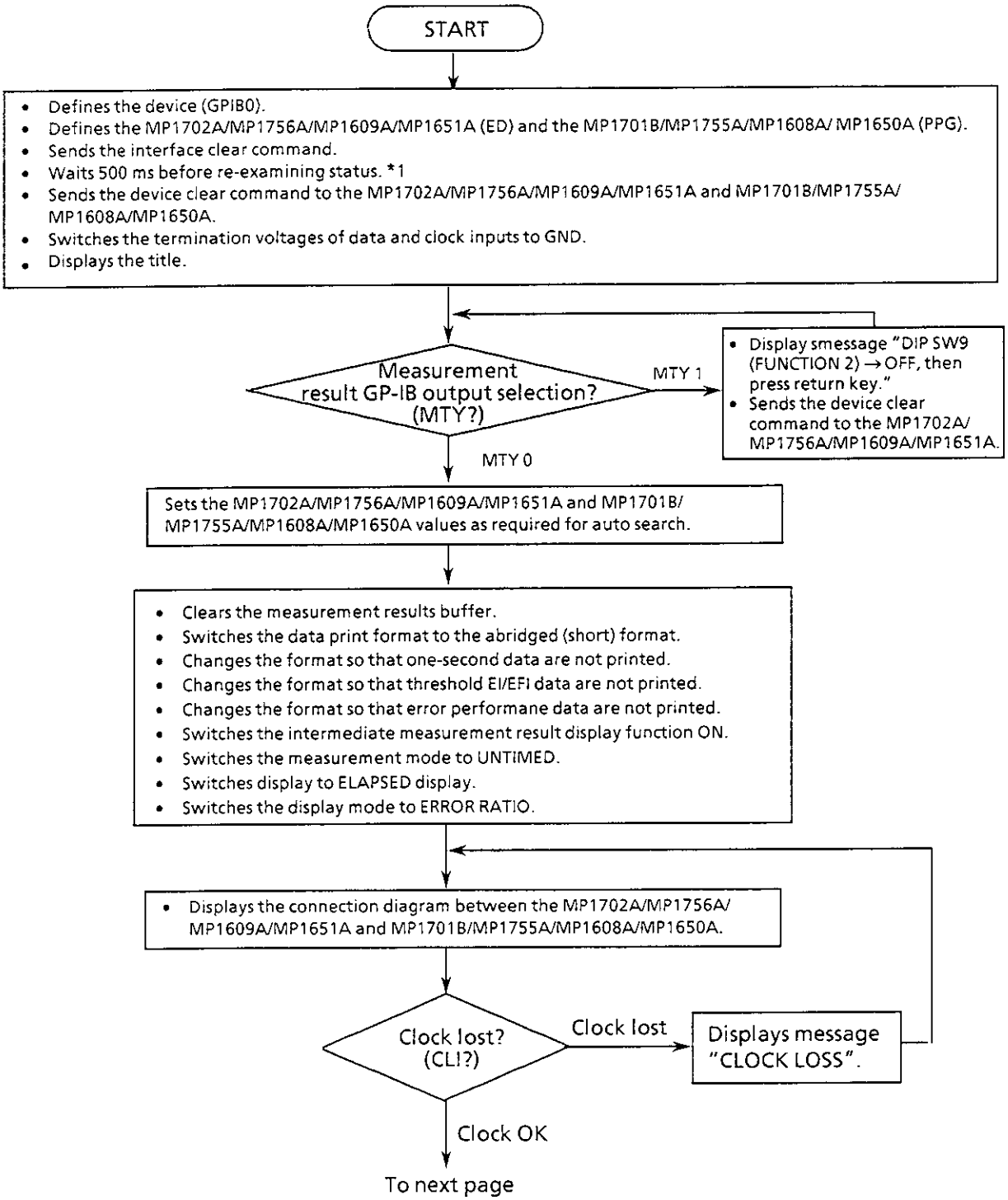
```
waidly: ' ----------  Wait delay  ----------
        '
     stm = TIMER
     etm = TIMER
     WHILE etm - stm < tim
         etm = TIMER
         IF etm < stm THEN etm = etm + 86400
     WEND
     '
     RETURN
         '
trap:    ' ----------  System trap  ----------
         '
     PRINT "IBERR%:" + STR$(IBERR%)
      STOP
         '
Connect: '---------- display Connection ----------
         '
     WINDOW (-600,-500) -(600,500)
     '
     LINE (-50,-50) - (600,500),14,B
     LINE (10,100) - STEP(235,200),,B
     LINE (300,100) - STEP(235,200),,B
     '
     CIRCLE (130,130),8
     LINE (130,125) - STEP(0,-55)
     LINE (130,70) - STEP(220,0)
     LINE (350,70) - STEP(0,55)
     CIRCLE (350,130),8
     '
     CIRCLE (190,130),8
     LINE (190,125) - STEP(0,-100)
     LINE (190,25) - STEP(220,0)
     LINE (410,25) - STEP(0,100)
     CIRCLE (410,130),8
     '
     LOCATE 2,50: PRINT "<<  CONNECTION  >>"
     '
     LOCATE 4,42: PRINT "MP1701B/MP1608A     MP1702A/MP1609A"
     LOCATE 5,42: PRINT "     /MP1650A            /MP1651A"
     '
     LOCATE 9,45: PRINT "DATA  CLOCK1"
     LOCATE 9,62: PRINT "DATA  CLOCK"
     '
     LOCATE 23,1
     INPUT "   Aer you ready ?  Press return key to start     ",A
     '
     LOCATE 23,1
     PRINT "                                                  "
     '
     LOCATE 4,1
     '
     RETURN
     '
     END
```

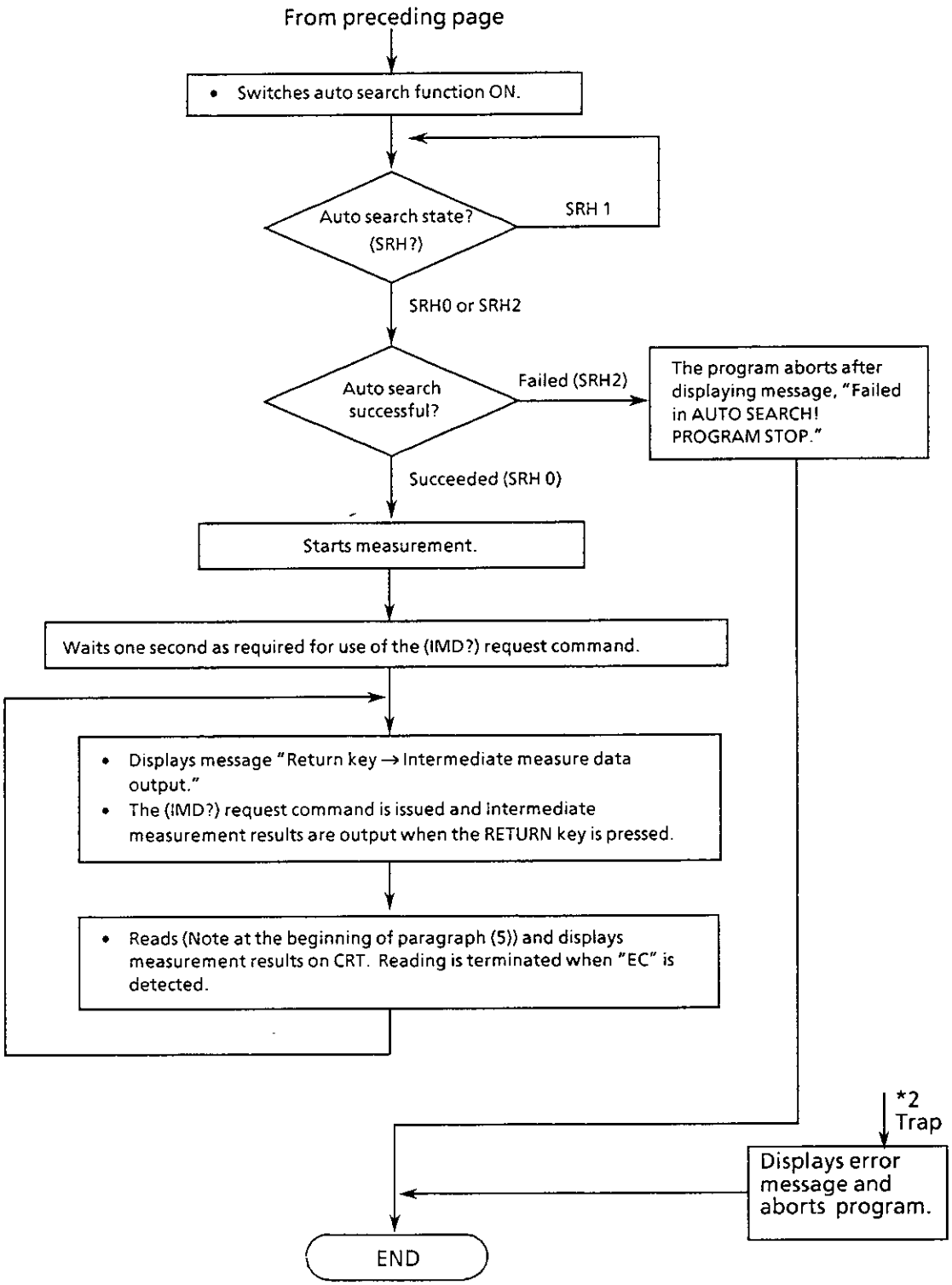
EXECUTED RESULT

```
STA 90,02,28,16,14,50
STA 90,02,28,16,14,50
INT 90,02,28,16,15,52
ELP      0,00,01,02
ER  1.0000E-04
EC     31000073
EI           62
EFI      0.0000
PFI           0
CLI           0
SLI           0
ES     100.0000
EFS      0.0000
SES      0.0000
DM     100.0000
US       0.0000
STA 90,02,28,16,14,50
INT 90,02,28,16,16,01
ELP      0,00,01,11
ER  1.0000E-04
EC     35500084
EI           71
EFI      0.0000
PFI           0
CLI           0
SLI           0
ES     100.0000
EFS      0.0000
SES      0.0000
DM     100.0000
US       0.0000
```

(e) Measurement (Measurement type: Measurement results are output each time a measurement is completed)

Standard data are output in short (abridged) format; intermediate measurement results are output in response to (IMD?) request command.





Note: See the beginning of this paragraph 15.5 for *1 and *2.

Program listing

```
' ************************************************************
' *                                                          *
' *     MP1702A/MP1609A/MP1651A  MEASUREMENT SAMPLE SOFT     *
' *        -- Abridged format , Standard output --           *
' *        -- Measurement end? >> Extension bit --           *
' *             * Intermediate measure data *                *
' *                                                ED2_M5    *
' ************************************************************

'-------------------------------------------------------------'
'                          MAIN  ROUTINE                      '
'-------------------------------------------------------------'

common shared IBSTA%,IBERR%,IBCNT%   ' Setup GPIB-PC functions
GOSUB gpinit                         ' Setup GPIB interface
wrt$="DTM 0;CTM 0" : GOSUB wrtcmd2

SCREEN 9
PRINT "** MP1702A/MP1609A/MP1651A **"
PRINT "** MEASUREMENT SAMPLE SOFT **"
PRINT

GOSUB MType                          ' Check Measurement type

GOSUB DSet                           ' Data setting
GOSUB Mode                           ' Measurement mode
GOSUB Clock                          ' Check Clock loss ?
GOSUB Srch                           ' Auto search : ON

wrt$="STA"  : GOSUB wrtcmd2          ' Measurement Start

GOSUB Result                         ' Display Result

STOP

'-------------------------------------------------------------'
'                          SUB ROUTINE                        '
'-------------------------------------------------------------'

MType : '---------- Check Measurement type ----------

    DO
        wrt$="MTY?" : GOSUB wrtcmd2 : GOSUB readcmd

        IF MID$(rd$,1,5)="MTY 1" THEN
          LOCATE 23,1
          INPUT "DIP SW 9 (Function 2)-->OFF, then press Enter key",A
          CALL IBCLR(ED%)

        END IF
    LOOP UNTIL MID$(rd$,1,5)="MTY 0"

    RETURN

DSet: '---------- Data Setting ----------

    wrt$= "LGC 0;PTN 2;MRK 1"     : GOSUB wrtcmd1
    wrt$= "LGC 0;PTN 2;MRK 1"     : GOSUB wrtcmd2
```

```
    wrt$= "CLK 1;RES 1;FRQ 3000" : GOSUB wrtcmd1
    wrt$= "OFS 0;SPD 0"          : GOSUB wrtcmd1
    wrt$= "DAP 1;DOS 1"          : GOSUB wrtcmd1
    wrt$= "CDL 100;CAP 1;COS 1"  : GOSUB wrtcmd1
    wrt$= "EAD 1"                : GOSUB wrtcmd1
    '
    wrt$="CFL 0"                 : GOSUB wrtcmd2
    wrt$="SRH 0"                 : GOSUB wrtcmd2
    wrt$="SYN 1"                 : GOSUB wrtcmd2
    '
    RETURN
    '
Mode: '---------- Measurement mode ----------
    '
    wrt$="BCL"   : GOSUB wrtcmd2  ' Measurement buffer clear
    wrt$="FMT 1" : GOSUB wrtcmd2  ' Data output : Abridged format
    wrt$="OSC 0" : GOSUB wrtcmd2  ' 1 sec data output : OFF
    wrt$="THR 0" : GOSUB wrtcmd2  ' Threshold output : OFF
    wrt$="EPF 0" : GOSUB wrtcmd2  ' Error performance output : OFF
    '
    wrt$="CUR 1" : GOSUB wrtcmd2  ' Current data : ON
    wrt$="MOD 2" : GOSUB wrtcmd2  ' Measurement mode : UNTIMED
    wrt$="TIM 4" : GOSUB wrtcmd2  ' Display ILAPSED
    wrt$="DMS 0" : GOSUB wrtcmd2  ' Display ERROR RATIO
    '
    RETURN
    '
Clock: '---------- Set data , Check connection ----------
    '
    DO
        GOSUB Connect             ' Display device connection
        '
        wrt$="CLI?" : GOSUB wrtcmd2 : GOSUB readcmd
        '
        IF MID$(rd$,1,5)="CLI 1" THEN PRINT "** CLOCK LOSS **"
        '
    LOOP UNTIL MID$(rd$,1,5)="CLI 0"
    '
    RETURN
    '
Srch: '---------- Auto search ON ----------
    '
    wrt$="SRH 1" : GOSUB wrtcmd2      ' Set Auto search : ON
    '
    DO
        wrt$="SRH?" : GOSUB wrtcmd2 : GOSUB readcmd
    LOOP UNTIL MID$(rd$,1,5)="SRH 0" or MID$(rd$,1,5)="SRH 2"
    '
    IF MID$(rd$,1,5)="SRH 2"  THEN
        PRINT "<<< Failed in AUTO SEARCH ! >>>"
        PRINT "    ** PROGRAM STOP  **"
        STOP
    END IF
    '
    RETURN
    '
Result: '---------- Display Result ----------
    '
    tim=1 : GOSUB waidly
    DO
        INPUT "Enter --> Intermediate measure data",A
        wrt$="IMD?" : GOSUB wrtcmd2
        '
```

```
        DO
            GOSUB readcmd : PRINT rd$
        LOOP UNTIL MID$(rd$,1,2)="EC"
    LOOP
    RETURN
gpinit: ' ---------- Set up GP-IB functions  ----------
    CALL IBFIND("GPIB0", GPIB0%)          ' Open device (GPIB0)
    IF GPIB0% < 0 THEN GOTO trap          ' system error
    CALL IBFIND("PPG", PPG%)              ' Open device(PPG)
    IF PPG% < 0 THEN GOTO trap            ' system error
    CALL IBFIND("ED", ED%)                ' Open device(ED)
    IF ED% < 0 THEN GOTO trap             ' system error
    CALL IBSIC(GPIB0%)                    ' Interface clear
    IF IBSTA% < 0 THEN GOTO trap          ' system error
    tim = 0.5
    GOSUB waidly
    CALL IBCLR(PPG%)                      ' Device clear(PPG)
    CALL IBCLR(ED%)                       ' Device clear(ED)
    RETURN
wrtcmd1: ' ----------  Write command (PPG) ----------
    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(PPG%, wrt$)                ' Write command (PPG)
    IF IBSTA% < 0 THEN GOTO trap          ' Trap
    RETURN
wrtcmd2: ' ----------  Write command (ED) ----------
    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(ED%, wrt$)                 ' Write command (ED)
    IF IBSTA% < 0 THEN GOTO trap          ' Trap
    RETURN
readcmd: ' ---------- Read command (ED) ----------
    CMD$="?"+chr$(42)+"A"
    CALL IBCMD(GPIB0%,CMD$)
    rd$=SPACE$(23)
    CALL IBRD(GPIB0%, rd$)                ' Read command (GPIB0)
    IF IBSTA% < 0 THEN GOTO trap          ' Trap
    RETURN
```

```
waidly:  ' ----------  Wait delay  ----------
         '
     stm = TIMER
     etm = TIMER
     WHILE etm - stm < tim
         etm = TIMER
         IF etm < stm THEN etm = etm + 86400
     WEND
     '
     RETURN
         '
trap:    ' ----------  System trap  ----------
         '
     PRINT "IBERR%:" + STR$(IBERR%)
      STOP
         '
Connect: '---------- display Connection ----------
         '
     WINDOW (-600,-500) -(600,500)
     '
     LINE (-50,-50) - (600,500),14,B
     LINE (10,100) - STEP(235,200),,B
     LINE (300,100) - STEP(235,200),,B
     '
     CIRCLE (130,130),8
     LINE (130,125) - STEP(0,-55)
     LINE (130,70) - STEP(220,0)
     LINE (350,70) - STEP(0,55)
     CIRCLE (350,130),8
     '
     CIRCLE (190,130),8
     LINE (190,125) - STEP(0,-100)
     LINE (190,25) - STEP(220,0)
     LINE (410,25) - STEP(0,100)
     CIRCLE (410,130),8
     '
     LOCATE 2,50: PRINT "<<  CONNECTION  >>"
     '
     LOCATE 4,42: PRINT "MP1701B/MP1608A    MP1702A/MP1609A"
     LOCATE 5,42: PRINT "    /MP1650A           /MP1651A"
     '
     LOCATE 9,45: PRINT "DATA  CLOCK1"
     LOCATE 9,62: PRINT "DATA  CLOCK"
     '
     LOCATE 23,1
     INPUT "   Aer you ready ?  Press return key to start    ",A
     '
     LOCATE 23,1
     PRINT "                                                  "
     '
     LOCATE 4,1
     '
     RETURN
     '
     END
```

EXECUTED RESULT

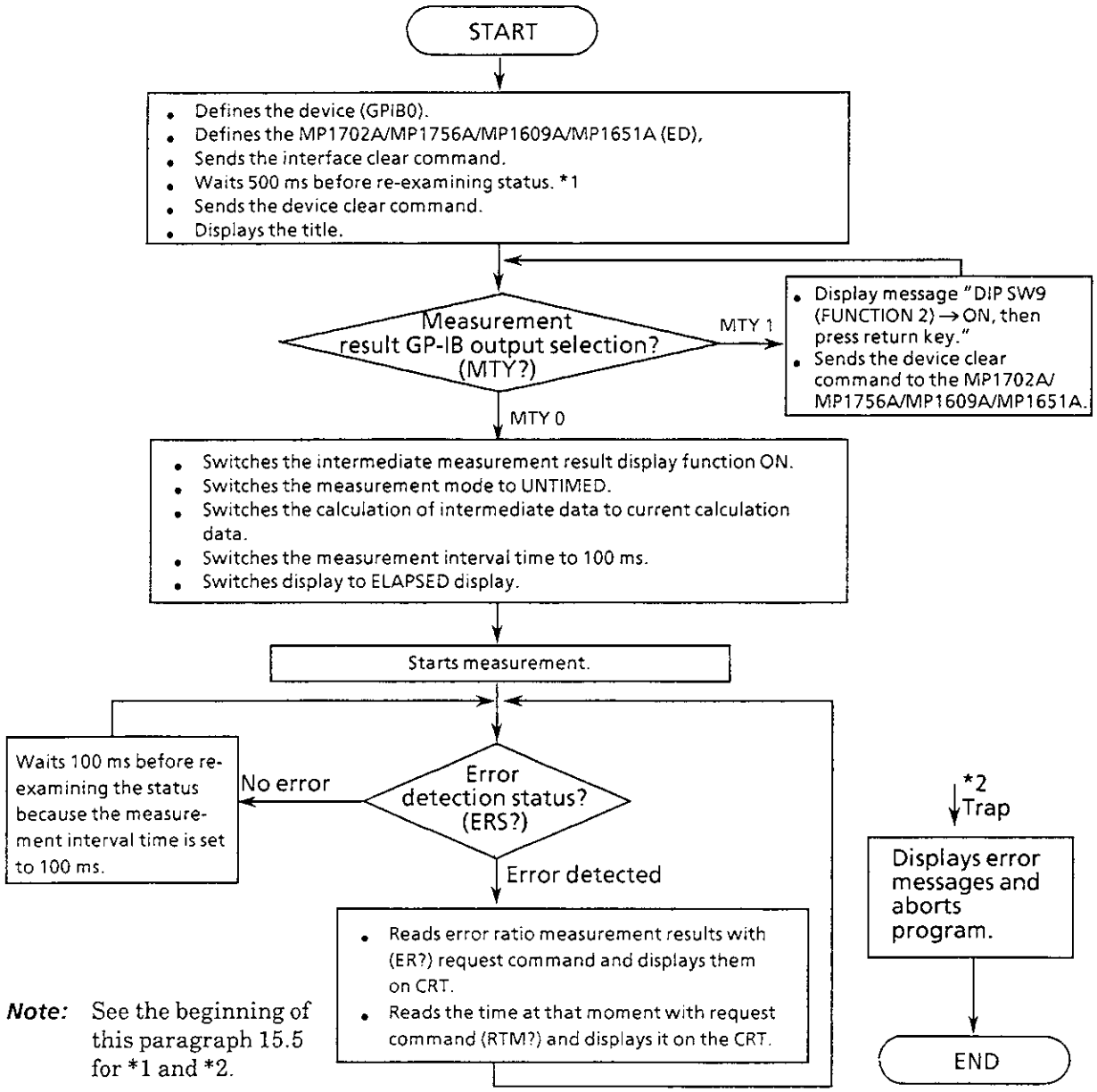
```
STA 90,02,28,16,18,08
INT 90,02,28,16,18,11
ER  1.0000E-04
EC      1500003
INT 90,02,28,16,18,14
ER  1.0000E-04
EC      3000007
INT 90,02,28,16,18,18
ER  1.0000E-04
EC      5000012
```

(6) Measurement (Measurement type: Measurement results are output when a data request command is received.)

(a) Error ratio measurement

This program reads error ratio measurement results when measurement results GP-IB output selection is set for output by the data request command.

Measurement is started after the parameters are appropriately initialized. The program checks the error detection status with the (ERS?) request command. When an error is detected, the error ratio measurement results are read with the (ER?) request command and displayed on CRT along with the time.



Note: See the beginning of this paragraph 15.5 for *1 and *2.

Program listing

```
' ************************************************************
' *                                                          *
' *      MP1702A/MP1609A/MP1651A   ERROR RATIO  SAMPLE SOFT  *
' *         -- Result output >> Data request command --      *
' *                                                          *
' *                                                 ED_ERR   *
' ************************************************************
'
'-----------------------------------------------------------'
'                        MAIN ROUTINE                       '
'-----------------------------------------------------------'
'
common shared IBSTA%,IBERR%,IBCNT%    ' Setup GPIB-PC functions
GOSUB gpinit                          ' Setup GPIB interface
'
PRINT "** MP1702A/MP1609A/MP1651A  ERROR RATIO SAMPLE SOFT ** "
PRINT
'
GOSUB MType                     ' Check measurement type
'
wrt$="CUR 1" : GOSUB wrtcmd   ' Current data : ON
wrt$="MOD 2" : GOSUB wrtcmd   ' Measurement mode : UNTIMED
wrt$="CAL 1" : GOSUB wrtcmd   ' Current data caluculation: EMEDIATE
wrt$="ITV 0" : GOSUB wrtcmd   ' Interval time : 100 ms
wrt$="DMS 0" : GOSUB wrtcmd   ' Display ERROR RATIO
'
GOSUB Measure
'
STOP
'
'-----------------------------------------------------------'
'                        SUB ROUTINE                        '
'-----------------------------------------------------------'
   '
MType:  '---------- Measurement type ? ----------
   '
    DO
      wrt$ = "MTY?" : GOSUB wrtcmd : GOSUB readcmd
      '
      IF MID$(rd$,1,5)="MTY 0" THEN
         INPUT "DIP SW 9 (Function 2)-->ON ,then press Enter key",A
         '
         CALL IBCLR(ED%)                  ' Device clear
         '
      END IF
    '
    LOOP UNTIL MID$(rd$,1,5)="MTY 1"
    '
    RETURN
   '
Measure:  '---------- Measurement ----------
   '
    wrt$="STA" : GOSUB wrtcmd                ' Measurement start
    '
    DO
        wrt$="ERS?" : GOSUB wrtcmd : GOSUB readcmd     ' Errors ?
        '
        IF MID$(rd$,1,5)="ERS 1" THEN
            GOSUB Result
```

```
        ELSE
            tim = 0.1 : GOSUB Waidly
        END IF
        '
    LOOP
    '
    RETURN
    '
Result: '---------- Print Result ----------
    '
    wrt$="ER?" : GOSUB wrtcmd              ' Error ratio ?
    GOSUB readcmd : PRINT MID$(rd$,1,14)
    '
    wrt$="RTM?" : GOSUB wrtcmd             ' Real time ?
    GOSUB readcmd : PRINT MID$(rd$,1,21)
    '
    RETURN
    '
gpinit: ' ----------  Setup GPIB interface  ----------
    '
    CALL IBFIND("GPIB0", GPIB0%)           ' Open device (GPIB0)
    IF GPIB0% < 0 THEN GOTO trap           ' system error
    '
    CALL IBFIND("ED", ED%)                 ' Open device (ED)
    IF ED% < 0 THEN GOTO trap              ' system error
    '
    CALL IBSIC(GPIB0%)                     ' Interface clear
    IF IBSTA% < 0 THEN GOTO trap           ' system error
    '
    tim = 0.5
    GOSUB waidly
    '
    CALL IBCLR(ED%)                        ' Device clear
    '
    RETURN
    '
wrtcmd: ' ----------  Write command  ----------
    '
    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(ED%, wrt$)                  ' Write command
    IF IBSTA% < 0 THEN GOTO trap           ' Trap
    '
    RETURN
    '
readcmd:' ----------  Read command  ----------
    '
    rd$=SPACE$(23)
    CALL IBRD(ED%, rd$)                    ' Read command
    IF IBSTA% < 0 THEN GOTO trap           ' Trap
    '
    RETURN
    '
waidly: ' ----------  Wait delay  ----------
    '
    stm = TIMER
    etm = TIMER
    WHILE etm - stm < tim
        etm = TIMER
        IF etm < stm THEN etm = etm + 86400
    WEND
    '
    RETURN
```

```
'
trap:    ' ----------  System trap  ----------
'
     PRINT "IBERR%:" + STR$(IBERR%)
     STOP
     '
     '
     END
```

EXECUTED RESULT

```
ER  1.0000E-04
RTM 90,02,28,16,49,00
ER  1.0000E-04
RTM 90,02,28,16,49,00
ER  1.0000E-04
RTM 90,02,28,16,49,00
ER  1.0000E-04
RTM 90,02,28,16,49,01
ER  1.0000E-04
RTM 90,02,28,16,49,01
ER  1.0000E-04
RTM 90,02,28,16,49,01
ER  1.0000E-04
RTM 90,02,28,16,49,01
ER  1.0000E-04
RTM 90,02,28,16,49,02
ER  1.0000E-04
RTM 90,02,28,16,49,02
ER  1.0000E-04
RTM 90,02,28,16,49,02
ER  1.0000E-04
RTM 90,02,28,16,49,02
ER  1.0000E-04
RTM 90,02,28,16,49,02
ER  1.0000E-04
RTM 90,02,28,16,49,03
ER  1.0000E-04
RTM 90,02,28,16,49,03
ER  1.0000E-04
RTM 90,02,28,16,49,03
ER  1.0000E-04
RTM 90,02,28,16,49,03
ER  1.0000E-04
RTM 90,02,28,16,49,04
```

(b) Measurementr result output

This program was written for applications in which the measurement result GP-IB output selection is set for output by data request commands.

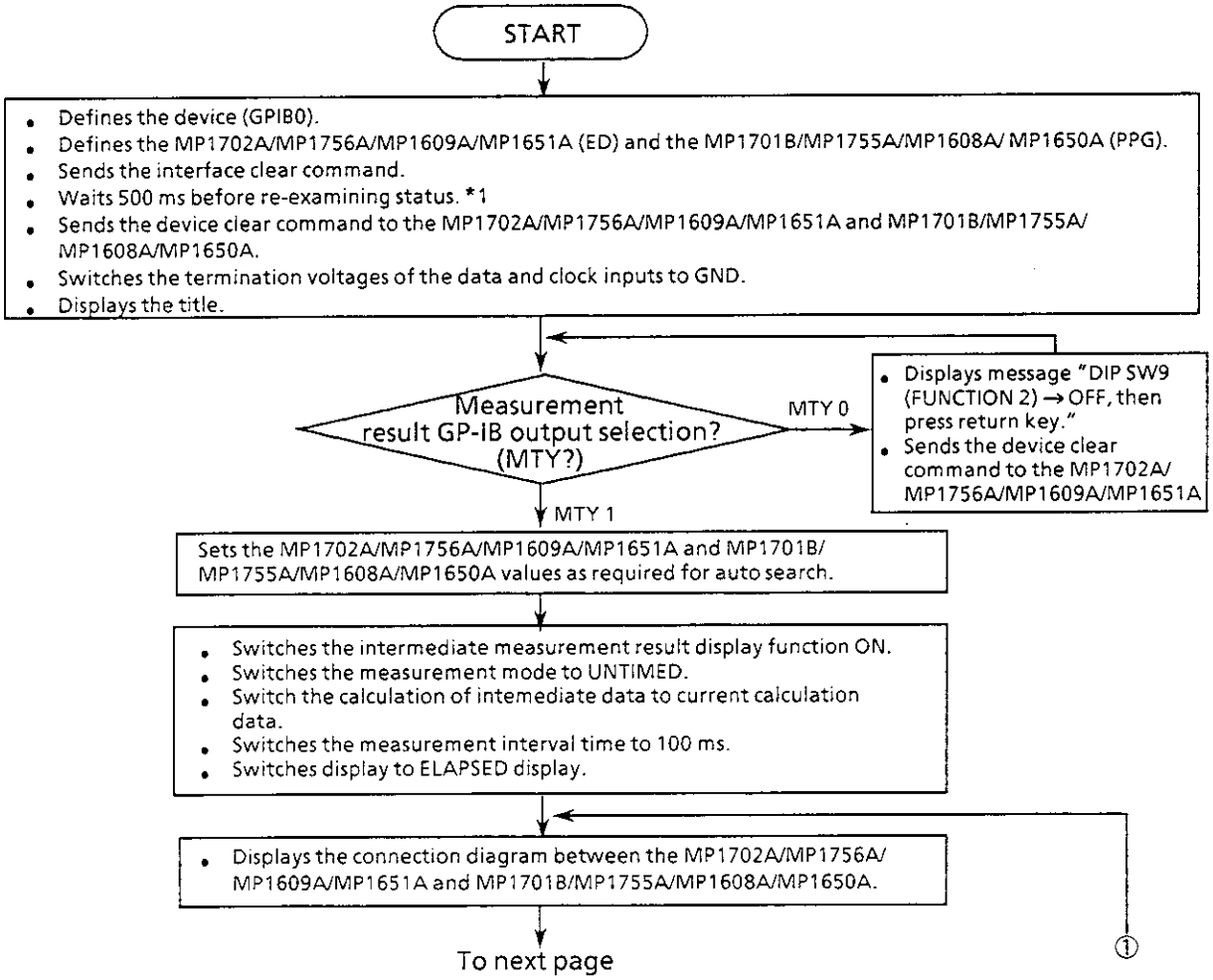
Measurement is initiated after connecting DATA to DATA and CLOCK1 to CLOCK between the MP1702A/MP1756A/MP1609A/MP1651A and MP1701B/MP1755A/MP1608A/MP1650A.

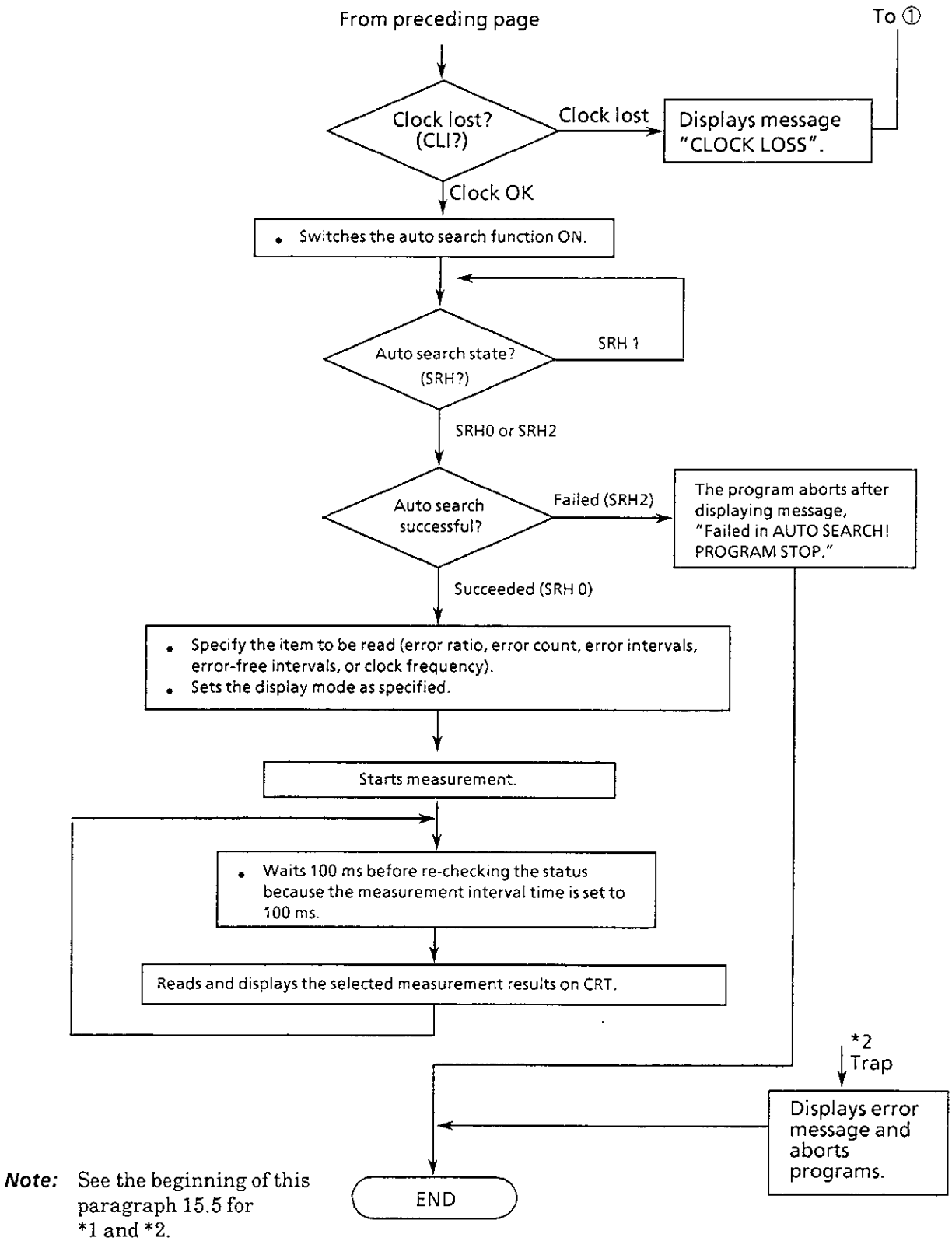
First, the program initializes the parameters as necessary for auto search and measurement. Next, it confirms that the clock is not lost and, after confirmation, it initiates an auto search.

If the auto search in completed successfully, the item to be read (error ratio, error count, error intervals, error-free intervals, or clock frequency) is selected and the program starts measurement after setting the appropriate display mode. After waiting 100 ms , the measurement results of the selected item are read via a request command and displayed on CRT.

On the other hand, if auto search fails, the program aborts.

Note: When using the MP1755A for the transmitter, internal clock frequency setting is ineffective. Set the frequency of the external clock source to 3000 MHz in advance.





Note: See the beginning of this paragraph 15.5 for *1 and *2.

Program listing

```
' ************************************************************
' *                                                          *
' *     MP1702A/MP1609A/MP1651A  MEASUREMENT  SAMPLE SOFT    *
' *       -- Result output >> Data request command --        *
' *                                                          *
' *                                                ED_MTY    *
' ************************************************************
'
'------------------------------------------------------------'
'                          MAIN  ROUTINE                     '
'------------------------------------------------------------'
'
common shared IBSTA%,IBERR%,IBCNT%   ' Setup GPIB-PC functions
GOSUB gpinit                         ' Setup GPIB interface
wrt$="DTM 0;CTM 0" : GOSUB wrtcmd2
'
SCREEN 9
PRINT "** MP1702A/MP1609A/MP1651A **"
PRINT "** MEASUREMENT SAMPLE SOFT **"
'
GOSUB MType                          ' Check Measurement type
'
GOSUB DSet                           ' Data setting
GOSUB Mode                           ' Measurement mode
GOSUB Clock                          ' Check clock loss
GOSUB Srch                           ' Auto search : ON
GOSUB Slct                           ' Select Output Result
'
wrt$="STA" : GOSUB wrtcmd2           ' Measurement start
'
GOSUB Result                         ' Print Result
'
STOP
'
'------------------------------------------------------------'
'                          SUB ROUTINE                       '
'------------------------------------------------------------'
'
MType: '---------- Check Measurement type ----------
'
   DO
      wrt$="MTY?" : GOSUB wrtcmd2 : GOSUB readcmd
      '
      IF MID$(rd$,1,5)="MTY 0" THEN
        LOCATE 23,1
        INPUT "DIP SW 9 (Function 2)-->ON,then press Enter key",A
        CALL IBCLR(ED%)                      ' Device clear(ED)
      END IF
   LOOP UNTIL MID$(rd$,1,5)="MTY 1"
   '
   RETURN
   '
DSet: '---------- Data setting ----------
   '
   wrt$="LGC 0;PTN 2;MRK 1"       : GOSUB wrtcmd1
   wrt$="LGC 0;PTN 2;MRK 1"       : GOSUB wrtcmd2
   '
   wrt$= "CLK 1;RES 1 ;FRQ 3000" : GOSUB wrtcmd1
   wrt$= "OFS 0;SPD 0"            : GOSUB wrtcmd1
```

```
    wrt$= "DAF 1;DOS 1"              : GOSUB wrtcmd1
    wrt$= "CDL 100;CAF 1;COS 1"      : GOSUB wrtcmd1
    wrt$= "EAD 1"                    : GOSUB wrtcmd1
    '
    wrt$="CFL 0"                     : GOSUB wrtcmd2
    wrt$="SRH 0"                     : GOSUB wrtcmd2
    wrt$="SYN 1"                     : GOSUB wrtcmd2
    '
    RETURN
    '
Mode: '---------- Measurement mode ----------
    '
    wrt$="CUR 1" : GOSUB wrtcmd2   ' Current data : ON
    wrt$="MOD 2" : GOSUB wrtcmd2   ' Measurement mode : UNTIMED
    wrt$="CAL 1" : GOSUB wrtcmd2   ' Current data culculation : IMEDIATE
    wrt$="ITV 0" : GOSUB wrtcmd2   ' Interval time : 100 ms
    wrt$="TIM 4" : GOSUB wrtcmd2   ' Display elapsed time
    '
    RETURN
    '
Clock: '---------- Check connection ----------
    '
    DO
       GOSUB Connect                ' Display Device connection
       '
       wrt$="CLI?" : GOSUB wrtcmd2 : GOSUB readcmd
       '
       IF MID$(rd$,1,5)="CLI 1" THEN PRINT "** CLOCK LOSS **"
       '
    LOOP UNTIL MID$(rd$,1,5)="CLI 0"
    '
    RETURN
    '
Srch: '---------- Auto search ON ----------
    '
    wrt$="SRH 1" : GOSUB wrtcmd2       ' Set Auto search : ON
    '
    DO
        wrt$="SRH?" : GOSUB wrtcmd2 : GOSUB readcmd
    LOOP UNTIL MID$(rd$,1,5)="SRH 0" or MID$(rd$,1,5)="SRH 2"
    '
    IF MID$(rd$,1,5)="SRH 2"  THEN
        PRINT "-- Failed in AUTO SEARCH ! --"
        PRINT "   ** PROGRAM STOP  **"
        STOP
    END IF
    '
    RETURN
    '
Slct: '---------- Select output result ----------
    '
    PRINT "--- OUTPUT DATA SELECT ---"
    PRINT "    * ERROR RATIO = [0]"
    PRINT "    * ERROR COUNT = [1]"
    PRINT "    * ERROR INTERVALS = [2]"
    PRINT "    * ERROR FREE INTERVALS = [3]"
    PRINT "    * CLOCK FREQUENCY = [ 4]"
    PRINT
    '
    DO
        INPUT "SELECT OUTPUT DATA [ 0 ~ 4 ] ";dta
    LOOP UNTIL dta>=0 AND dta<=4
    '
```

```
    SELECT CASE dta
    '
    CASE 0
        dms$="0"
        cmd$="ER?"
        '
    CASE 1
        dms$="1"
        cmd$="EC?"
        '
    CASE 2
        dms$="2"
        cmd$="EI?"
        '
    CASE 3
        dms$="3"
        cmd$="EFI?"
        '
    CASE 4
        dms$="4"
        cmd$="FRQ?"
        '
    END SELECT
    '
    wrt$="DMS "+dms$ : GOSUB wrtcmd2
    '
    RETURN
        '
Result: '---------- Display Result ----------
        '
    DO
        tim = 0.1 : GOSUB Waidly
        wrt$=cmd$ : GOSUB wrtcmd2
        GOSUB readcmd : PRINT rd$
        '
    LOOP
    '
    RETURN
        '
gpinit: ' ---------- Set up GP-IB functions  ----------
        '
    CALL IBFIND("GPIB0", GPIB0%)          ' Open device (GPIB0)
    IF GPIB0% < 0 THEN GOTO trap          ' system error
    '
    CALL IBFIND("PPG", PPG%)              ' Open device(PPG)
    IF PPG% < 0 THEN GOTO trap            ' system error
    '
    CALL IBFIND("ED", ED%)                ' Open device(ED)
    IF ED% < 0 THEN GOTO trap             ' system error
    '
    CALL IBSIC(GPIB0%)                    ' Interface clear
    IF IBSTA% < 0 THEN GOTO trap          ' system error
    '
    tim = 0.5
    GOSUB waidly
    '
    CALL IBCLR(PPG%)                      ' Device clear(PPG)
    '
    CALL IBCLR(ED%)                       ' Device clear(ED)
    '
    RETURN
        '
```

```
wrtcmd1:  ' ----------  Write command (PPG) ----------
          '
     wrt$=wrt$+chr$(13)+chr$(10)
     CALL IBWRT(PPG%, wrt$)                ' Write command (PPG)
     IF IBSTA% < 0 THEN GOTO trap          ' Trap
     '
     RETURN
          '
wrtcmd2:  ' ----------  Write command (ED) ----------
          '
     wrt$=wrt$+chr$(13)+chr$(10)
     CALL IBWRT(ED%, wrt$)                 ' Write command (ED)
     IF IBSTA% < 0 THEN GOTO trap          ' Trap
     '
     RETURN
          '
readcmd:  ' ---------- Read command (ED) ----------
          '
     rd$=SPACE$(16)
     CALL IBRD(ED%, rd$)                   ' Read command (ED)
     IF IBSTA% < 0 THEN GOTO trap          ' Trap
     '
     RETURN
          '
waidly:   ' ----------  Wait delay  ----------
          '
     stm = TIMER
     etm = TIMER
     WHILE etm - stm < tim
         etm = TIMER
         IF etm < stm THEN etm = etm + 86400
     WEND
     '
     RETURN
          '
trap:     ' ----------  System trap  ----------
          '
     PRINT "IBERR%:" + STR$(IBERR%)
      STOP
          '
Connect:  '---------- Display Connection ----------
          '
     WINDOW (-600,-500) -(600,500)
     '
     LINE (-50,-50) - (600,500),14,B
     LINE (10,100) - STEP(235,200),,B
     LINE (300,100) - STEP(235,200),,B
     '
     CIRCLE (130,130),8
     LINE (130,125) - STEP(0,-55)
     LINE (130,70) - STEP(220,0)
     LINE (350,70) - STEP(0,55)
     CIRCLE (350,130),8
     '
     CIRCLE (190,130),8
     LINE (190,125) - STEP(0,-100)
     LINE (190,25) - STEP(220,0)
     LINE (410,25) - STEP(0,100)
     CIRCLE (410,130),8
     '
     LOCATE 2,50: PRINT "<<  CONNECTION  >>"
     '
```

```
LOCATE 4,42: PRINT "MP1701B/MP1608A    MP1702A/MP1609A"
LOCATE 5,42: PRINT "    /MP1650A           /MP1651A"
'
LOCATE 9,45: PRINT "DATA  CLOCK1"
LOCATE 9,62: PRINT "DATA  CLOCK"
'
LOCATE 23,1
INPUT "   Aer you ready ? Press return key to start.    ",A
'
LOCATE 23,1
PRINT "                                                  "
LOCATE 4,1
'
RETURN
'
'
END
```

EXECUTED RESULT

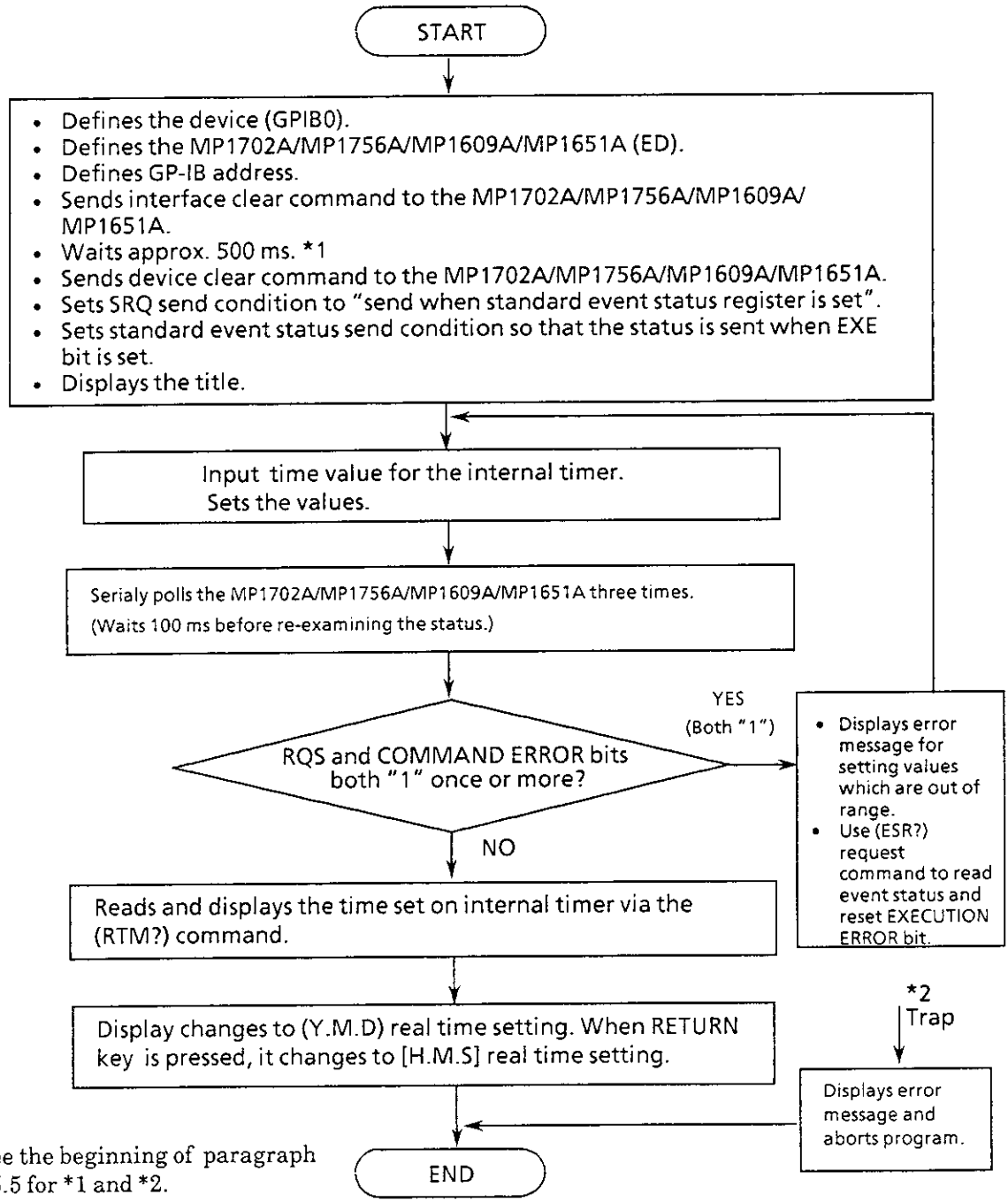
[ERROR RATIO]		[ERROR COUNT]		[ERROR INTERVALS]	
ER	1.0000E-04	EC	50000	EI	1.0000E-99
ER	1.0000E-04	EC	50000	EI	1.0000E-99
ER	1.0000E-04	EC	50000	EI	1.0000E-99
ER	1.0000E-04	EC	50000	EI	1.0000E-99
ER	1.0000E-04	EC	50000	EI	1.0000E-99
ER	1.0000E-04	EC	50000	EI	1.0000E-99
ER	1.0000E-04	EC	50000	EI	1
ER	1.0000E-04	EC	50000	EI	1
ER	1.0000E-04	EC	50000	EI	1
ER	1.0000E-04			EI	1
ER	1.0000E-04			EI	2
				EI	2
				EI	2

[ERROR FREE INTERVALS]		[CLOCK FREQUENCY]	
EFI	1.0000E-99	FRQ	3000.011
EFI	1.0000E-99	FRQ	3000.011
EFI	1.0000E-99	FRQ	3000.011
EFI	1.0000E-99	FRQ	3000.011
EFI	1.0000E-99	FRQ	3000.011
EFI	1.0000E-99	FRQ	3000.011
EFI	0.0000	FRQ	3000.011
EFI	0.0000	FRQ	3000.011
EFI	0.0000	FRQ	3000.011
EFI	0.0000	FRQ	3000.011
EFI	0.0000	FRQ	3000.011
EFI	0.0000	FRQ	3000.011
EFI	0.0000	FRQ	3000.011
EFI	0.0000	FRQ	3000.011

(7) Internal timer setting

This program sets the internal timer, reads its value by data request command, and then displays this value on the CRT.

The range of the input value is monitored by checking the standard event status byte.



Note:

See the beginning of paragraph 15.5 for *1 and *2.

Program listing

```
' ************************************************************
' *                                                          *
' *   MP1702A/MP1609A/MP1651A REAL TIME SETTING  SAMPLE SOFT  *
' *                                                          *
' *                                                ED_RTM    *
' ************************************************************
'
'------------------------------------------------------------
'                          MAIN  ROUTINE                     '
'------------------------------------------------------------
'
common shared IBSTA%,IBERR%,IBCNT%     ' Setup GPIB-PC functions
GOSUB gpinit                           ' Setup GPIB interface
'
wrt$ = "SRQ 32"   : GOSUB wrtcmd       ' SRQ : COMMAND ERROR bit
wrt$ = "ESE 16"   : GOSUB wrtcmd       ' Execution error bit
'
GOSUB DSet
GOSUB Dsply
'
STOP
'
'------------------------------------------------------------
'                          SUB  ROUTINE                      '
'------------------------------------------------------------
'
DSet:    '----------  Set real time ----------
'
    DO
      CLS
      PRINT "    ** MP1702A/MP1609A/MP1651A **"
      PRINT "** REAL TIME SETTING SAMPLE SOFT **"
      PRINT
      INPUT "Real time setting data * YEAR *    [ 0~99 ] ";Y$
      INPUT "Real time setting data * MONTH *   [ 1~12 ] ";M$
      INPUT "Real time setting data * DAY *     [ 1~31 ] ";D$
      INPUT "Real time setting data * HOURE *   [ 0~23 ] ";H$
      INPUT "Real time setting data * MINUTE *  [ 0~59 ] ";Mi$
      INPUT "Real time setting data * SECOND *  [ 0~59 ] ";S$
      PRINT
      '
      rtm$=Y$+","+M$+","+D$+","+H$+","+Mi$+","+S$
      wrt$="RTM "+rtm$   : GOSUB wrtcmd
      '
      GOSUB check
      '
    LOOP UNTIL Result$="OK"
    '
    RETURN
        '
Dsply:   '---------- Display real time ----------
        '
    wrt$="RTM?" : GOSUB wrtcmd
    GOSUB readcmd : PRINT rd$
    '
    wrt$="TIM 0" : GOSUB wrtcmd
    INPUT "* Press ENTER key to display [ H,M,S ] ",A
    wrt$="TIM 1" : GOSUB wrtcmd
    '
    RETURN
        '
```

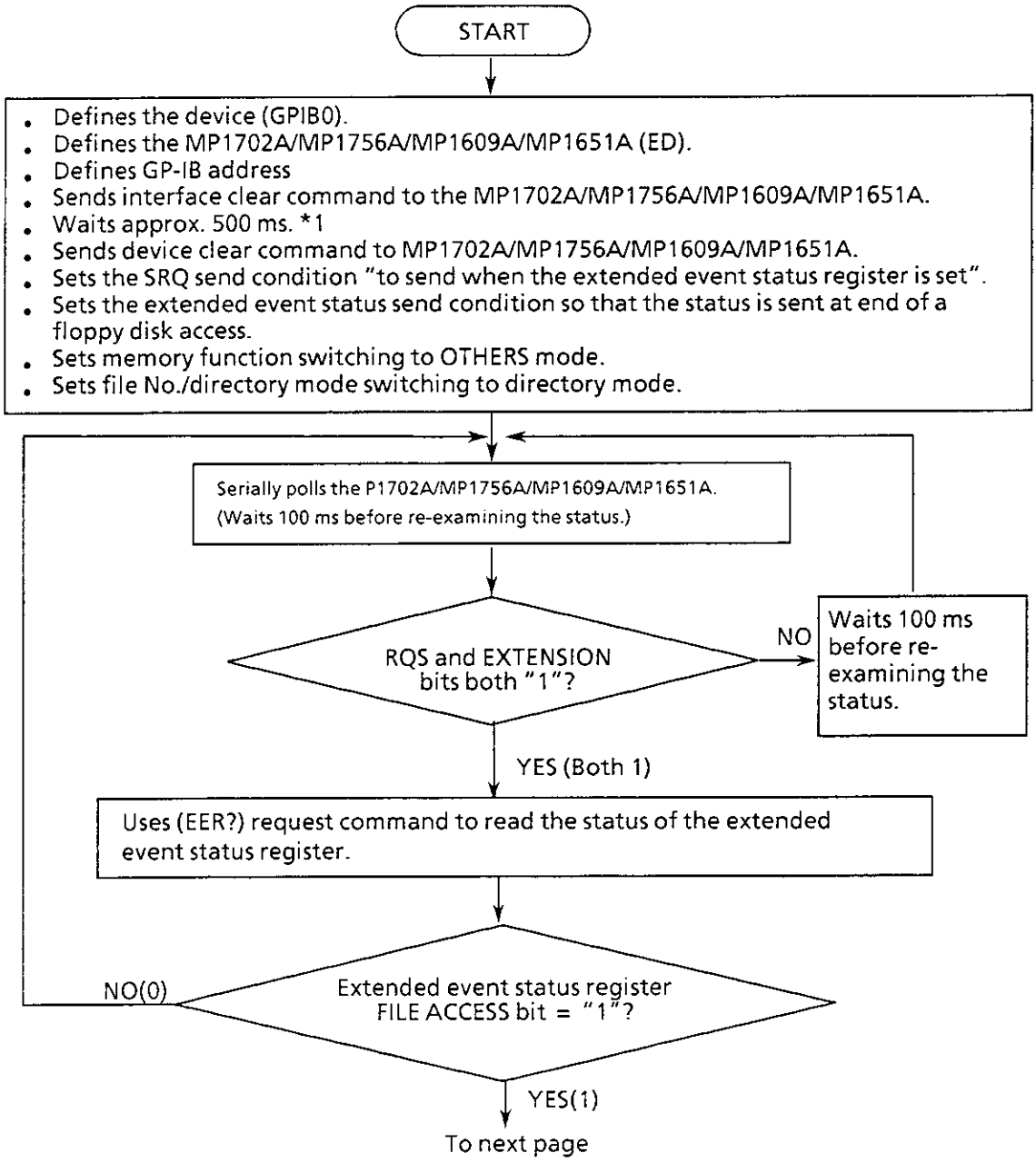
```
check:   '---------- Check Set data ----------
         '
     rslt=0
     '
     FOR I=0 TO 2
     '
       CALL IBRSP(ED%,SPR%)                ' Serial polling
       IF IBSTA% < 0 THEN GOTO trap
       '
       s=SPR% AND &H40
       q=SPR% AND &H20
       '
       tim = 0.1 : GOSUB waidly
       '
       IF s=&H40 AND q=&H20 THEN rslt=1   ' RQS,COMMAND ERROR bit=1
       '
     NEXT I
     '
     IF rslt=1 THEN
       INPUT " EXECUTION ERROR !!  Press ENTER key ",A
       wrt$="ESR?" : GOSUB wrtcmd         ' Reset Execution error bit
     ELSE
         Result$="OK"
     END IF
     '
     RETURN
         '
gpinit:  ' ---------- Setup GPIB interface  ----------
         '
     CALL IBFIND("GPIB0", GPIB0%)          ' Open device (GPIB0)
     IF GPIB0% < 0 THEN GOTO trap          ' system error
     '
     CALL IBFIND("ED", ED%)                ' Open device (ED)
     IF ED% < 0 THEN GOTO trap             ' system error
     '
     CALL IBSIC(GPIB0%)                    ' Interface clear
     IF IBSTA% < 0 THEN GOTO trap          ' system error
     '
     tim = 0.5
     GOSUB waidly
     '
     CALL IBCLR(ED%)                       ' Device clear
     '
     RETURN
         '
wrtcmd:  ' ----------  Write command  ----------
         '
     wrt$=wrt$+chr$(13)+chr$(10)
     CALL IBWRT(ED%, wrt$)                 ' Write command
     IF IBSTA% < 0 THEN GOTO trap          ' Trap
     '
     RETURN
         '
readcmd: ' ----------  Read command  ----------
         '
     rd$=SPACE$(23)
     CALL IBRD(ED%, rd$)                   ' Read command
     IF IBSTA% < 0 THEN GOTO trap          ' Trap
     '
     RETURN
         '
```

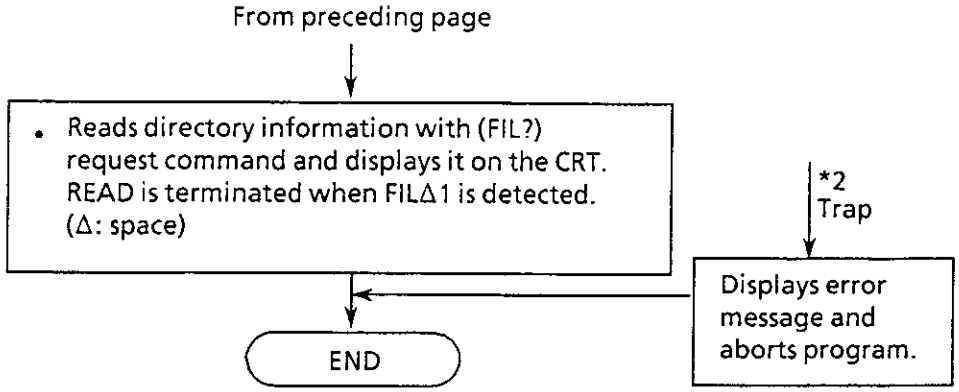
```
waidly:  ' ----------  Wait delay  ----------
         '
     stm = TIMER
     etm = TIMER
     '
     WHILE etm - stm < tim
         etm = TIMER
         IF etm < stm THEN etm = etm + 86400
     WEND
     '
     RETURN
         '
trap:    ' ----------  System trap  ----------
         '
     PRINT "IBERR%:" + STR$(IBERR%)
     STOP
     '
     '
     END
```

(8) Floppy disk file information read
(Floppy disk access status check by serial polling)

This program checks the directory information for files stored on the floppy disk, and display it on the CRT. Afterwards, an example of program output is listed.

Floppy disk access is checked by serial polling and by data request command that checks the extended event status information.





Note: See the beginning of paragraph 15.5 for *1 and *2.

EXECUTED RESULTS

```
R00      ,PTN,    65664,89-12-14,21:05
R01      ,PTN,     1920,89-12-14,20:13
R01      ,OTH,      144,90-01-18,18:54
R03      ,PTN,      130,89-12-15,20:18
R03      ,OTH,      144,90-01-18,18:55
R04      ,PTN,     6016,90-01-23,10:59
R05      ,PTN,    65664,90-01-11,13:06
R05      ,OTH,      144,89-12-28,14:25
R06      ,PTN,    65664,90-01-11,13:08
R07      ,PTN,     2176,90-01-11,13:09
R08      ,PTN,     2176,90-01-11,13:10
R10      ,PTN,    65664,90-01-29,21:08
R50      ,PTN,      144,90-01-16,11:10
R50      ,OTH,      144,90-01-18,18:55
R80      ,PTN,      132,90-01-29,21:08
FIL 1
```

Program listing

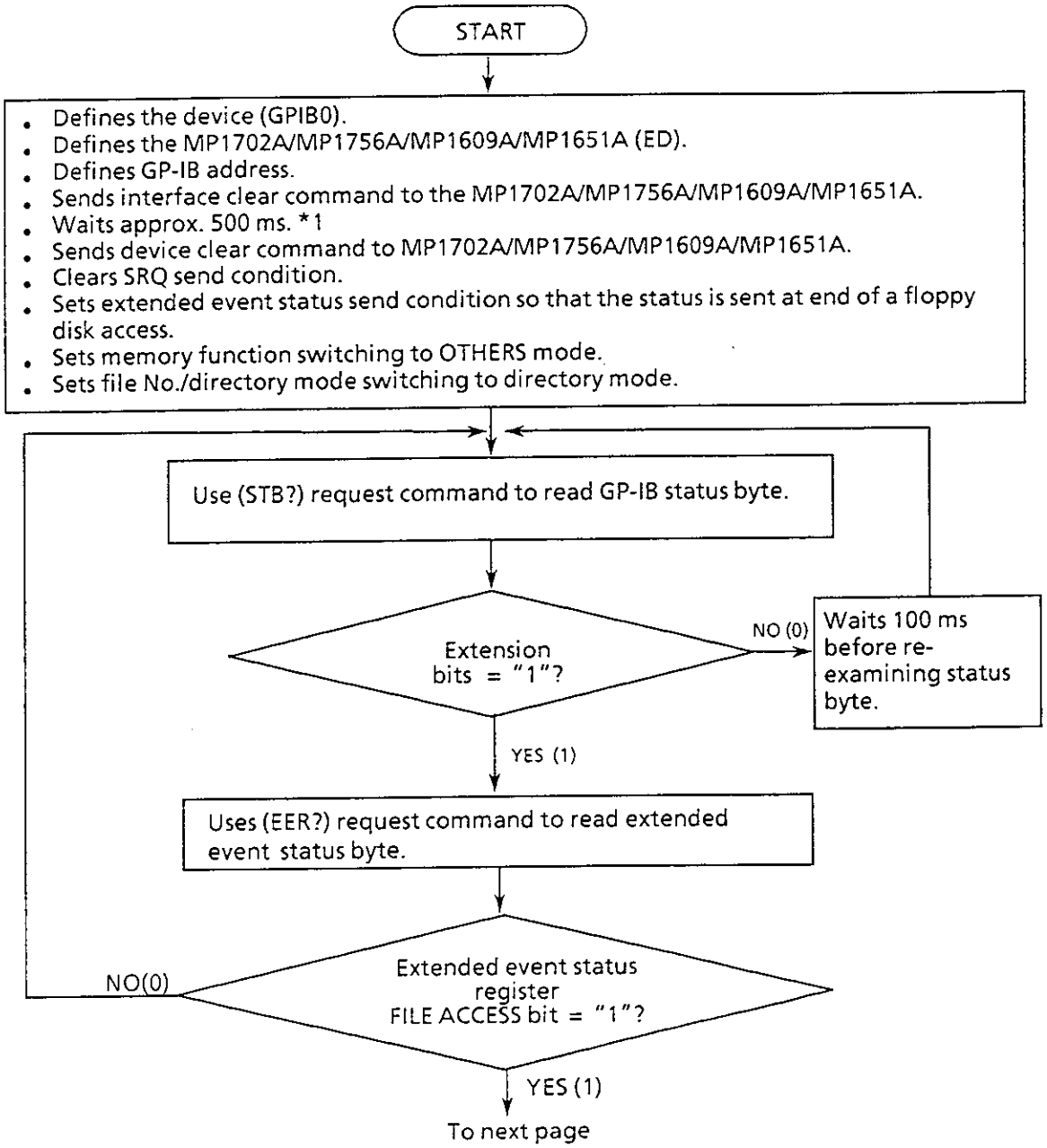
```
'*****************************************************************
'*                                                               *
'*  MP1702A/MP1609A/MP1651A FILE DIRECTORY READ SAMPLE SOFT_1    *
'*                                                               *
'*                                                      ED_DIR   *
'*****************************************************************
'
'---------------------------------------------------------------
'                          MAIN  ROUTINE                        '
'---------------------------------------------------------------
'
common shared IBSTA%,IBERR%,IBCNT%      ' Setup GPIB-PC functions
GOSUB gpinit                            ' Setup GPIB interface
'
wrt$ = "SRQ 2"   : GOSUB wrtcmd         ' SRQ : Extension bit
wrt$ = "EES 32"  : GOSUB wrtcmd         ' Floppy access end
wrt$ = "MEM 1"   : GOSUB wrtcmd         ' Memory mode : OTHERS
wrt$ = "FIL 1"   : GOSUB wrtcmd         ' Directory mode
'
GOSUB SpollEer
GOSUB FileDir
'
 STOP
'
'---------------------------------------------------------------
'                          SUB  ROUTINE                         '
'---------------------------------------------------------------
'
SpollEer:'----------  Check Status Byte  ----------
'
    DO
      DO
        CALL IBRSP(ED%,SPR%)            ' Send Serial poll
        IF IBSTA% < 0 THEN GOTO trap
'
        s=SPR% AND &H40
        q=SPR% AND &H2
'
        tim = 0.1 : GOSUB waidly
'
      LOOP UNTIL s=&H40 AND q=&H2       ' RQS bit,Extension bit = 1
'
      wrt$ = "EER?"
      GOSUB wrtcmd              ' REQUEST Extension Event Register ?
      GOSUB readcmd             ' READ Extension Event Register
'
    LOOP UNTIL MID$(rd$,9,1) = "1"      ' File Access bit = 1
'
    RETURN
'
FileDir:' ----------  File directory  ----------
'
    wrt$ = "FIL?" : GOSUB wrtcmd        ' REQUEST Directory ?
'
    DO
      GOSUB readcmd                     ' READ    Directory
      PRINT rd$
    LOOP UNTIL MID$(rd$,1,5) = "FIL 1"
'
    RETURN
```

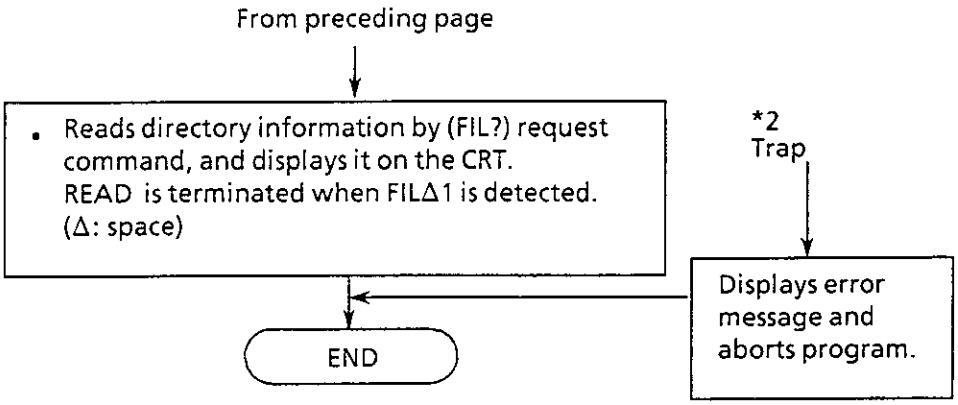
```
gpinit:  ' ---------- Setup GPIB interface  ----------
         '
     CALL IBFIND("GPIB0", GPIB0%)          ' Open device (GPIB0)
     IF GPIB0% < 0 THEN GOTO trap          ' system error
     '
     CALL IBFIND("ED", ED%)                ' Open device (ED)
     IF ED% < 0 THEN GOTO trap             ' system error
     '
     CALL IBSIC(GPIB0%)                    ' Interface clear
     IF IBSTA% < 0 THEN GOTO trap          ' system error
     '
     tim = 0.5
     GOSUB waidly
     '
     CALL IBCLR(ED%)                       ' Device clear (ED)
     '
     RETURN
         '
wrtcmd:  ' ----------  Write command  ----------
         '
     wrt$=wrt$+chr$(13)+chr$(10)
     CALL IBWRT(ED%, wrt$)                 ' Write command
     IF IBSTA% < 0 THEN GOTO trap          ' Trap
     '
     RETURN
         '
readcmd: ' ----------  Read command  ----------
         '
     rd$ = SPACE$(38)
     CALL IBRD(ED%,rd$)                    ' Read command
     IF IBSTA% < 0 THEN GOTO trap          ' Trap
     '
     RETURN
         '
waidly:  ' ----------  Wait delay  ----------
         '
     stm = TIMER
     etm = TIMER
     '
     WHILE etm - stm < tim
         etm = TIMER
         IF etm < stm THEN etm = etm + 86400
     WEND
     '
     RETURN
         '
trap:    ' ----------  System trap  ----------
         '
     PRINT "IBERR%:" + STR$(IBERR%)

     STOP
     '
     '
     END
```

(9) Floppy disk file information read
(Floppy disk access status check by (STB?) request command)

This program checks the directory information for files stored on floppy disk, and displays it on the CRT. Afterwards, an example of program output is shown.

Floppy disk access is monitored with data request commands that check the GP-IB status byte and the extended event status byte.





Note: See the beginning of paragraph 15.5 for *1 and *2.

EXECUTED RESULTS

```
R00      ,PTN,    65664,89-12-14,21:05
R01      ,PTN,     1920,89-12-14,20:13
R01      ,OTH,      144,90-01-18,18:54
R03      ,PTN,      130,89-12-15,20:18
R03      ,OTH,      144,90-01-18,18:55
R04      ,PTN,     6016,90-01-23,10:59
R05      ,PTN,    65664,90-01-11,13:06
R05      ,OTH,      144,89-12-28,14:25
R06      ,PTN,    65664,90-01-11,13:08
R07      ,PTN,     2176,90-01-11,13:09
R08      ,PTN,     2176,90-01-11,13:10
R10      ,PTN,    65664,90-01-29,21:08
R50      ,PTN,      144,90-01-16,11:10
R50      ,OTH,      144,90-01-18,18:55
R80      ,PTN,      132,90-01-29,21:08
FIL 1
```

Program listing

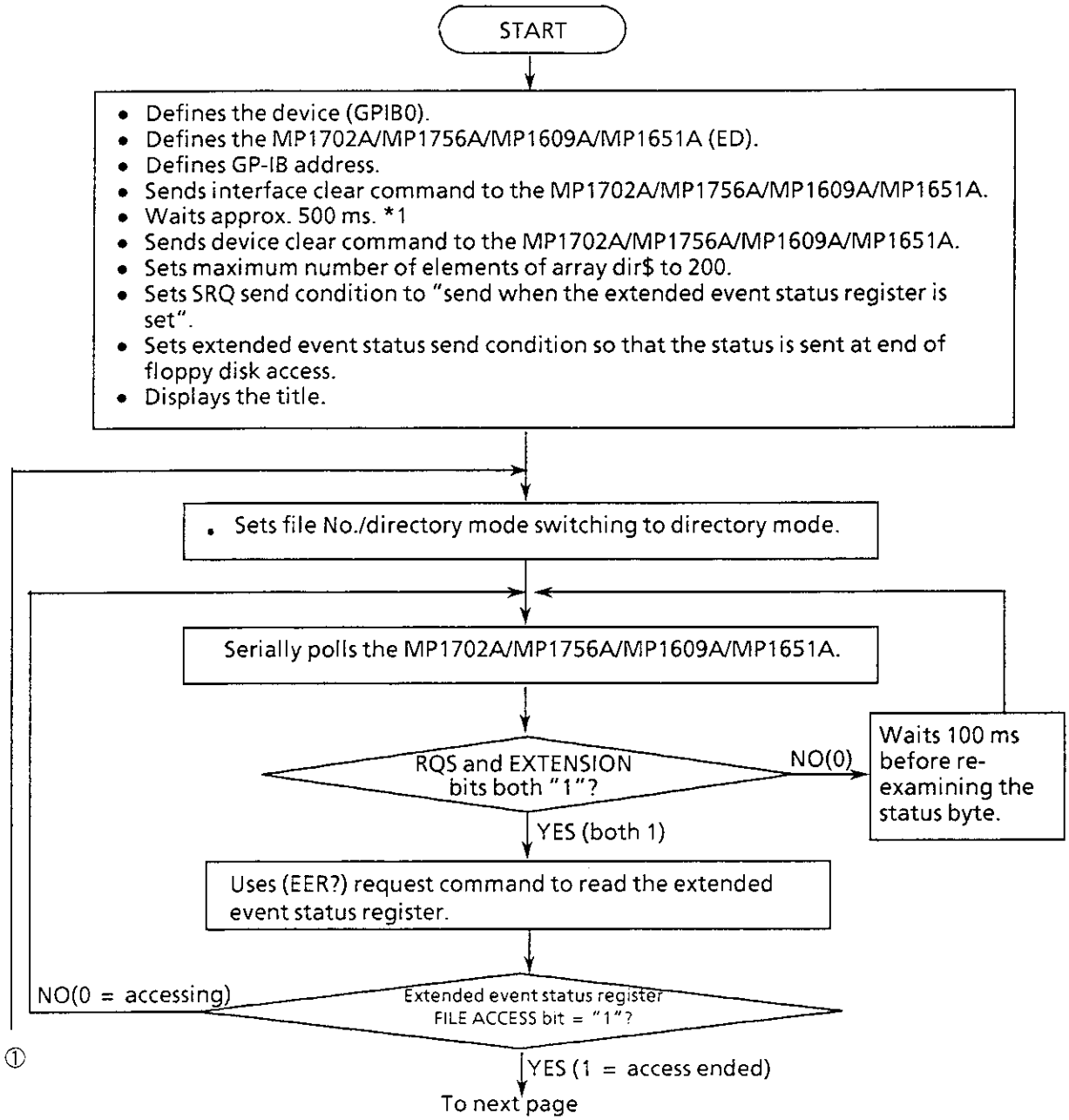
```
' ********************************************************************
' *                                                                  *
' *  MP1702A/MP1609A/MP1651A  FILE DIRECTORY READ SAMPLE SOFT_2      *
' *                                                                  *
' *                                                        ED_DIR2   *
' ********************************************************************
'
'------------------------------------------------------------------'
'                            MAIN  ROUTINE                         '
'------------------------------------------------------------------'
'
common shared IBSTA%,IBERR%,IBCNT%    ' Set up GP-IB functions
GOSUB gpinit
'
wrt$ = "SRQ 0"   : GOSUB wrtcmd      ' SRQ : Off
wrt$ = "EES 32"  : GOSUB wrtcmd      ' EES : Floppy access end
wrt$ = "MEM 1"   : GOSUB wrtcmd      ' FIL : Memory mode : OTHERS
wrt$ = "FIL 1"   : GOSUB wrtcmd      ' FIL : Directory mode
'
'
GOSUB StbReg
GOSUB FileDir
'
STOP
'
'------------------------------------------------------------------'
'                          SUB  ROUTINE                            '
'------------------------------------------------------------------'
'
StbReg: '----------  Check Status Byte  ----------
'
  DO
    DO
      wrt$ = "STB?" : GOSUB wrtcmd   ' REQUEST Status Byte Register ?
      '
      GOSUB readcmd                  ' READ Status Byte Register
      tim = 0.1 : GOSUB waidly
      '
    LOOP UNTIL MID$(rd$,13,1) = "1"  ' Extension bit = 1
    '
  wrt$ = "EER?" : GOSUB wrtcmd       ' REQUEST Extension Register ?
  '
  GOSUB readcmd                      ' READ Extension Register
  '
  LOOP UNTIL MID$(rd$,9,1) = "1"     ' File Access bit = 1
  '
  RETURN
'
FileDir: ' ----------  File directory   ----------
'
  wrt$ = "FIL?" : GOSUB wrtcmd         ' REQUEST Directory ?
  '
  DO
    '
    GOSUB readcmd                      ' READ Directory
    PRINT rd$
    '
  LOOP UNTIL MID$(rd$,1,5) = "FIL 1"   '  File Access bit = 1
  '
  RETURN
'
```

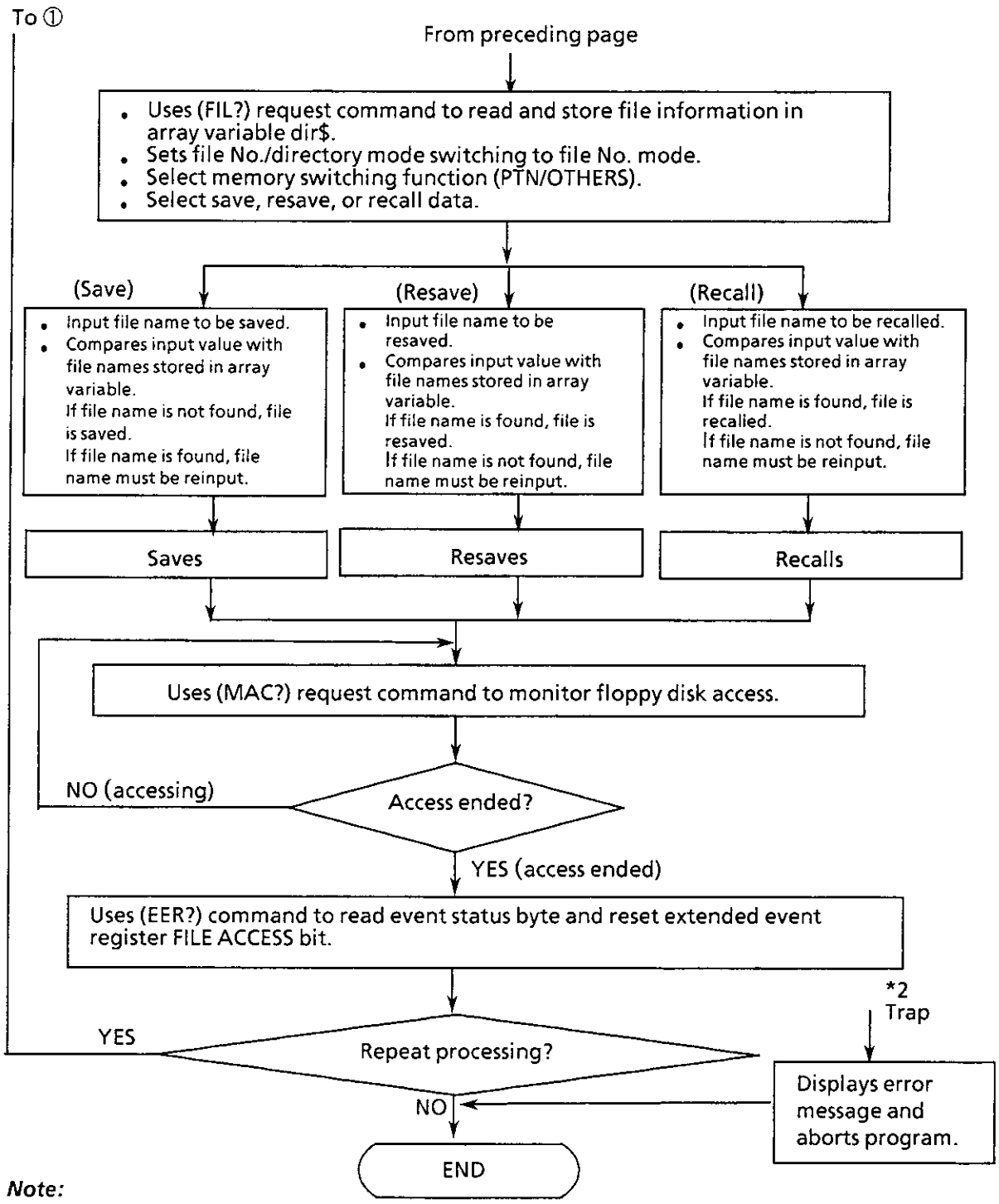
```
gpinit: ' ---------- Set up GP-IB functions  ----------
        '
    CALL IBFIND("GPIB0", GPIB0%)          ' Open device (GPIB0)
    IF GPIB0% < 0 THEN GOTO trap          ' system error
    '
    CALL IBFIND("ED", ED%)                ' Open device(ED)
    IF ed% < 0 THEN GOTO trap             ' system error
    '
    CALL IBSIC(GPIB0%)                    ' Open device
    IF IBSTA% < 0 THEN GOTO trap          ' system error
    '
    tim = 0.5
    GOSUB waidly
    '
    CALL IBCLR(ED%)                       ' Device clear
    '
    RETURN
        '
wrtcmd: ' ----------  Write command  ----------
        '
    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(ED%, wrt$)                 ' Write command
    IF IBSTA% < 0 THEN GOTO trap          ' Trap
    '
    RETURN
        '
readcmd:' ----------  Read command  ----------
        '
    rd$=SPACE$(38)
    CALL IBRD(ED%, rd$)                   ' Read command
    IF IBSTA% < 0 THEN GOTO trap          ' Trap
    '
    RETURN
        '
waidly: ' ----------  Wait delay  ----------
        '
    stm = TIMER
    etm = TIMER
    '
    WHILE etm - stm < tim
        etm = TIMER
        IF etm < stm THEN etm = etm + 86400
    WEND
    '
    RETURN
        '
trap:   ' ----------  System trap  ----------
        '
    PRINT "IBERR%:" + STR$(IBERR%)
    STOP
    '
    '
    END
```

(10) Data save, resave, and recall

This program reads and recalls directory and file information for files stored on floppy disk. It also saves or resaves the status of the MP1702A/MP1756A/MP1609A/MP1651A corresponding to the memory switching function (PTN/OTHERS).

The access status of the floppy disk to obtain the directory information is checked by serial polling and by data request command to check the extended event status. During save, resave, and recall, floppy disk access is checked by the MAC? request command.





Note:

See the beginning of paragraph 15.5 for *1 and *2.

Program listing

```
' ************************************************************
' *                                                          *
' *        MP1702A/MP1609A/MP1651A  MEMORY  SAMPLE SOFT      *
' *                                                          *
' *                                                ED_MEMO   *
' ************************************************************
'
'------------------------------------------------------------'
'                          MAIN  ROUTINE                     '
'------------------------------------------------------------'
'
common shared IBSTA%,IBERR%,IBCNT%   ' Setup GPIB-PC  functions
GOSUB gpinit                         ' Setup GPIB interface
'
DIM dir$(199)
wrt$ = "SRQ 2"  : GOSUB wrtcmd       ' SRQ : Extension bit
wrt$ = "EES 32" : GOSUB wrtcmd       ' Floppy access end
'
PRINT "** MP1702A/MP1609A/MP1651A  MEMORY  SAMPLE SOFT **"
PRINT
DO
    GOSUB SPoll : GOSUB DSet : GOSUB Floppy
    '
    INPUT " NEXT DATA SET [ YES=0 , NO=1 ] ";loop$
    PRINT
    '
LOOP UNTIL loop$="1"
'
STOP
'
'------------------------------------------------------------'
'                          SUB  ROUTINE                      '
'------------------------------------------------------------'
'
SPoll:  '////////////////////////////////////////////////////////
        '/                                                      /
        '/   Serial polling                                    /
        '/     RQS,Extension bit = 1  ,  File access bit = 1   /
        '/                                                      /
        '////////////////////////////////////////////////////////
        '
    wrt$="FIL 1" : GOSUB wrtcmd          ' Directory mode
    '
    DO
      DO
        CALL IBRSP(ED%,SPR%)             ' Serial polling
        IF IBSTA < 0 THEN GOTO TRAP
        '
        s=SPR% AND &H40
        q=SPR% AND &H2
        '
        tim = 0.1 : GOSUB waidly
        '
      LOOP UNTIL s=&H40 AND q=&H2        ' RQS,Extension bit = 1
      '
      wrt$="EER?" : GOSUB wrtcmd  ' REQUEST Extension event register
      GOSUB readcmd               ' READ Extension event register
      '
    LOOP UNTIL MID$(rd$,9,1)="1"  ' File access bit = 1
    '
    RETURN
    '
```

```
DSet:   '////////////////////////////////////////////////////////////
        '/                                                          /
        '/  Read file directory , Set Memory mode   &               /
        '/             Select  SAVE  or  RESAVE  or  RECALL         /
        '/                                                          /
        '////////////////////////////////////////////////////////////
        '
    '------------------------------------ Read File directory
    '
    I=0
    '
    '
    wrt$="FIL?" :GOSUB wrtcmd              ' REQUEST File directory
    '
    DO
      GOSUB readcmd                        ' READ File directory
      dir$(I)=MID$(rd$,1,12)
      I=I+1
    LOOP UNTIL MID$(rd$,1,5)="FIL 1"       ' File end
    '
    '------------------------------------------ Set Memory mode
    '
    wrt$="FIL 0" : GOSUB wrtcmd
    '
    DO
      INPUT " MEMORY MODE SELECT [ PTN=0 , OTHERS=1 ] ";mem
    LOOP UNTIL mem=0 OR mem=1
    '
    wrt$="MEM "+STR$(mem) : GOSUB wrtcmd
    '
    IF mem=0 THEN mem$="PTN"
    IF mem=1 THEN mem$="OTH"
    '
    '---------------------------- Select SAVE or RESAVE or RECALL
    '
    DO
      INPUT " SELECT [ SAVE=0 , RESAVE=1 , RECALL=2 ]";dta
    LOOP UNTIL dta=0 OR dta=1 OR dta=2
    '
    IF dta=0 THEN GOSUB DSave
    IF dta=1 THEN GOSUB DResave
    IF dta=2 THEN GOSUB DRecall
    '
    RETURN
    '
DSave:  '////////////////////////////////////////////////////////////
        '/                                                          /
        '/   *  Data Save *                                        /
        '/              SAME Memory mode & File name --> ERROR !    /
        '/                                                          /
        '////////////////////////////////////////////////////////////
        '
    DO
      I=0
      result$=SPACE$(4)
      INPUT " ** Data Save **  File Name [ 0~99 ] ";nam
      '
      DO
        IF mem$=MID$(dir$(I),10,3) AND nam=VAL(MID$(dir$(I),2,2)) THEN
        result$="SAME"
        END IF
        '
        I=I+1
```

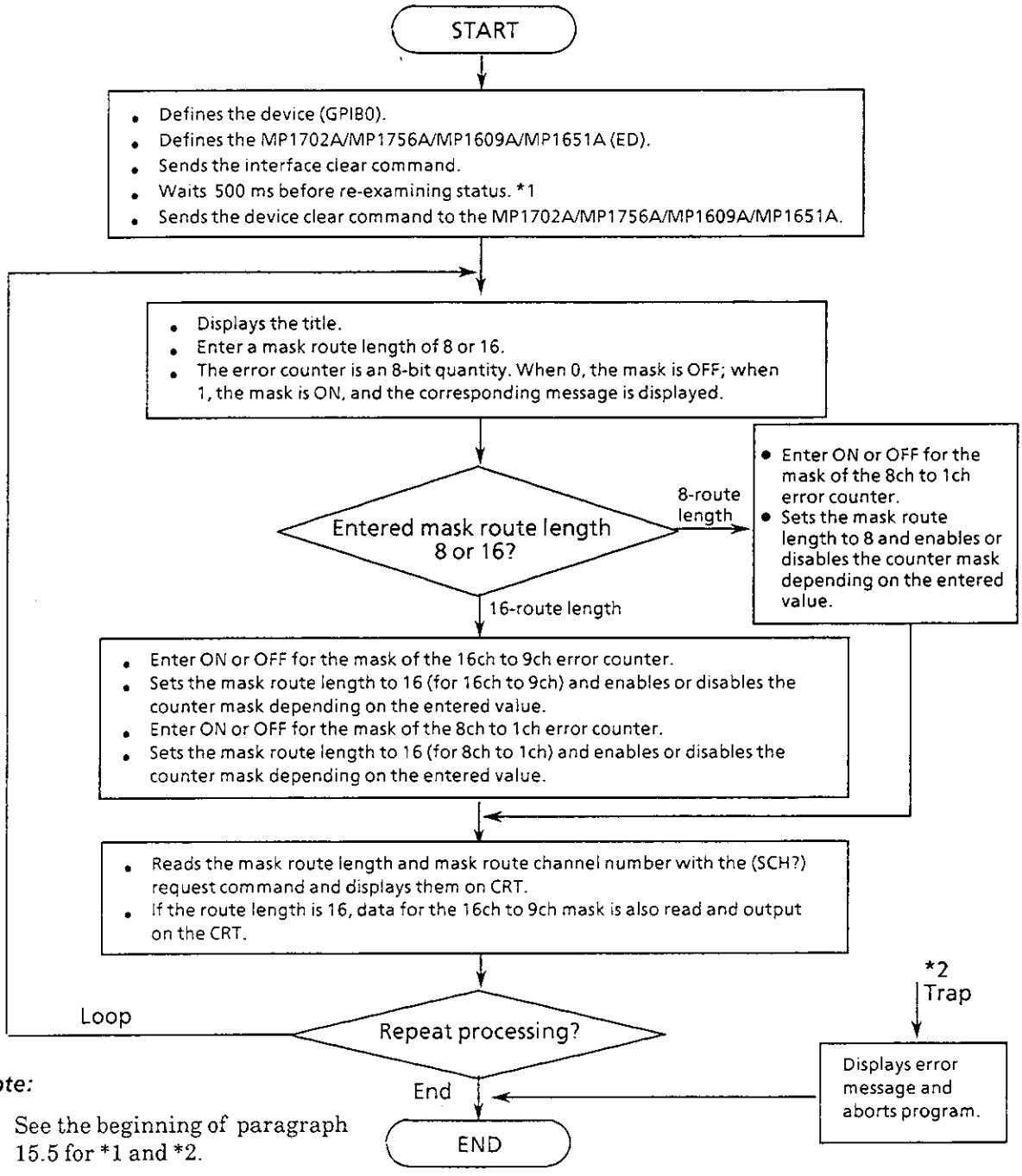
```
      LOOP UNTIL MID$(dir$(I),1,5)="FIL 1"
    LOOP UNTIL result$ <> "SAME"
    wrt$="SAV "+STR$(nam) : GOSUB wrtcmd
    RETURN
DResave:'////////////////////////////////////////////////////////////
        '/                                                          /
        '/   *  Data Resave *                                      /
        '/              SAME Memory mode & File name --> OK !      /
        '/                                                         /
        '////////////////////////////////////////////////////////////
    DO
      I=0
      result$=SPACE$(4)
      INPUT " ** Data Resave **  File Name [ 0~99 ] ";nam
      DO
        IF mem$=MID$(dir$(I),10,3) AND nam=VAL(MID$(dir$(I),2,2)) THEN
        result$="SAME"
        END IF
        I=I+1
      LOOP UNTIL MID$(dir$(I),1,5)="FIL 1"
    LOOP UNTIL result$ = "SAME"
    wrt$="RSV "+STR$(nam) : GOSUB wrtcmd
    RETURN
DRecall:'////////////////////////////////////////////////////////////
        '/                                                          /
        '/   *  Data Recall *                                      /
        '/              SAME Memory mode & File name --> OK !      /
        '/                                                         /
        '////////////////////////////////////////////////////////////
    DO
      I=0
      result$=SPACE$(4)
      INPUT " ** Data Recall **  File Name [ 0~99 ] ";nam
      DO
        IF mem$=MID$(dir$(I),10,3) AND nam=VAL(MID$(dir$(I),2,2)) THEN
        result$="SAME"
        END IF
        I=I+1
      LOOP UNTIL MID$(dir$(I),1,5)="FIL 1"
    LOOP UNTIL result$ = "SAME"
    wrt$="RCL "+STR$(nam) : GOSUB wrtcmd
    RETURN
```

```
Floppy: '////////////////////////////////////////////////////////
        '/                                                      /
        '/     Memory Access Condition ?                        /
        '/                    &  Reset  FILE ACCESS bit         /
        '/                                                      /
        '////////////////////////////////////////////////////////
    DO
        wrt$="MAC?" : GOSUB wrtcmd : GOSUB readcmd
    LOOP UNTIL MID$(rd$,1,5)="MAC 0"
    '
    wrt$="EER?" : GOSUB wrtcmd : GOSUB readcmd
    RETURN
    '
gpinit: ' ----------  Setup GPIB interface  ----------
    '
    CALL IBFIND("GPIB0", GPIB0%)          ' Open device (GPIB0)
    IF GPIB0% < 0 THEN GOTO trap          ' system error
    '
    CALL IBFIND("ED", ED%)                ' Open device (ED)
    IF ED% < 0 THEN GOTO trap             ' system error
    '
    CALL IBSIC(GPIB0%)                    ' Interface clear
    IF IBSTA% < 0 THEN GOTO trap          ' system error
    '
    tim = 0.5: GOSUB waidly
    '
    CALL IBCLR(ED%)                       ' Device clear
    '
    RETURN
    '
wrtcmd: ' ----------  Write command  ----------
    '
    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(ED%, wrt$)                       ' Write command
    IF IBSTA% < 0 THEN GOTO trap                ' Trap
    '
    RETURN
    '
readcmd:' ----------  Read command  ----------
    '
    rd$=SPACE$(38)
    CALL IBRD(ED%, rd$ )                        ' Read command
    IF IBSTA% < 0 THEN GOTO trap                ' Trap
    '
    RETURN
    '
waidly: ' ----------  Wait delay  ----------
    '
    stm = TIMER
    etm = TIMER
    '
    WHILE etm - stm < tim
        etm = TIMER
        IF etm < stm THEN etm = etm + 86400
    WEND
    '
    RETURN
    '
trap:   ' ----------  System trap  ----------
    '
    PRINT "IBERR%:" + STR$(IBERR%)
    STOP
    '
    END
```

(11) Setting the mask route length and channnel number

This program sets the mask route length and channel number.

First, the program selects the mask route length and turns the mask for each error counter ON or OFF depending on the selected mask route length. Then, it reads the mask route length value with (SCH?) request command and displays it on CRT.



Note:

See the beginning of paragraph 15.5 for *1 and *2.

Program listing

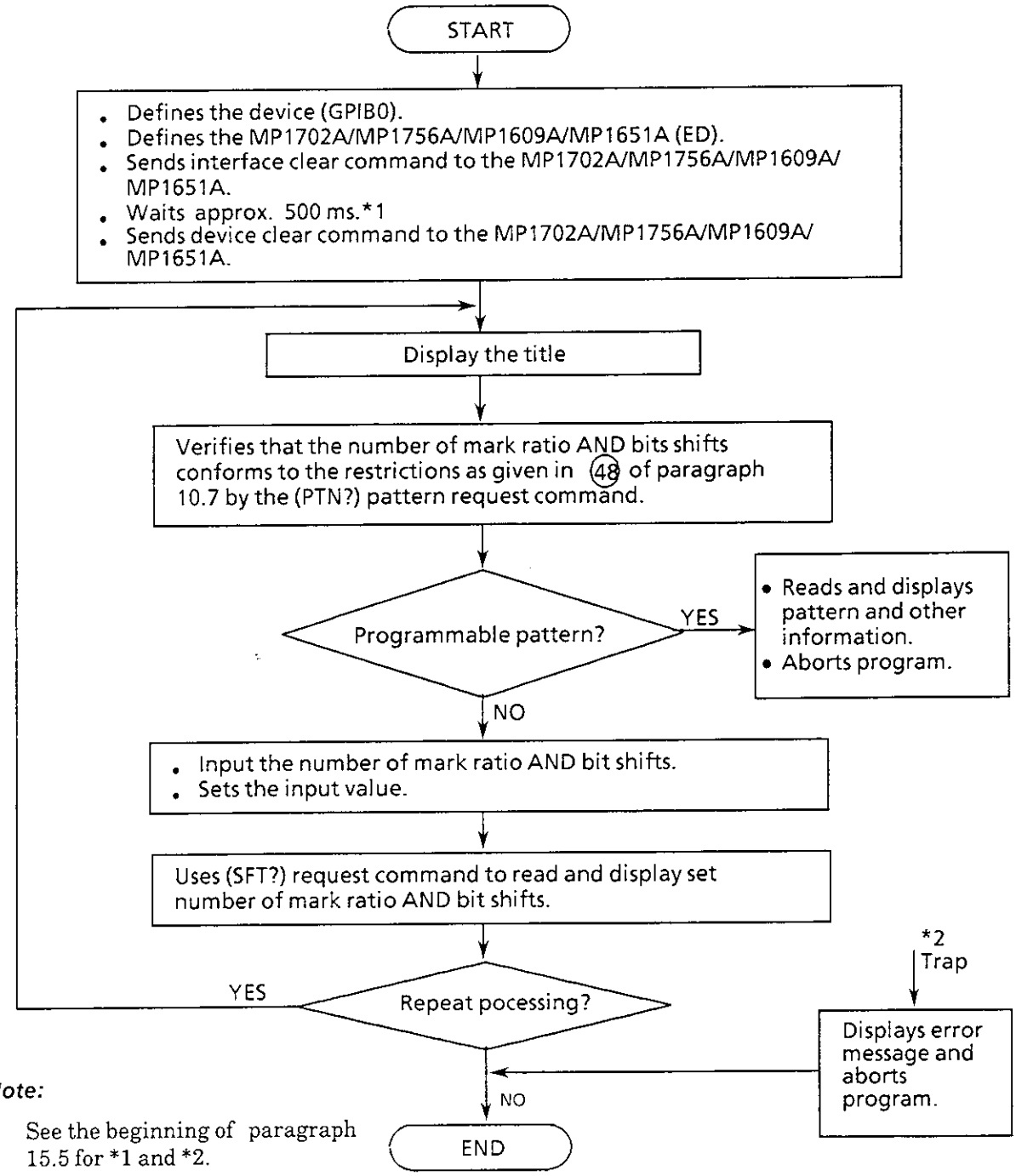
```
' ************************************************************
' *                                                          *
' *   MP1702A/MP1609A/MP1651A  MEASURE CH MASK  SAMPLE SOFT  *
' *                                                          *
' *                                                ED_MASK   *
' ************************************************************
'
'------------------------------------------------------------'
'                          MAIN  ROUTINE                     '
'------------------------------------------------------------'
'
common shared IBSTA%,IBERR%,IBCNT%      ' Setup GPIB-PC functions
GOSUB gpinit                            ' Setup GPIB interface
'
DO
    CLS
    '
    PRINT "*** MP1702A/MP1609A/MP1651A MEASURE CH MASK SAMPLE ***"
    PRINT
    '
    GOSUB Route
    GOSUB Dsply
    '
    INPUT "NEXT DATA SET [ YES=0 , NO=1 ] ";loop$
    PRINT
LOOP UNTIL loop$="1"
'
STOP
'
'------------------------------------------------------------'
'                          SUB  ROUTINE                      '
'------------------------------------------------------------'
'
Route:  ' ---------- Set Mask route length & Channel ----------
'
DO
    INPUT "MASK ROUTE LENGTH [ 8 LENGTH=0 , 16 LENGTH=1 ] ";msk
LOOP UNTIL msk=0 or msk=1
'
PRINT "ERROR COUNTER = 8 bit [ 0=MASK OFF , 1=MASK ON ]"
PRINT
'
IF msk=1 THEN
    PRINT "** MASK ROUTE LENGTH =16 **"
    '
    INPUT "CHANNEL SET DATA BIT8 (16ch)-->BIT1 (9ch) [0/1]";sch2$
    wrt$ = "MSK 2;SCH #B"+sch2$ : GOSUB wrtcmd
    '
    INPUT "CHANNEL SET DATA BIT8 (8ch)-->BIT1 (1ch)  [0/1] ";sch1$
    wrt$ = "MSK 1;SCH #B"+sch1$ : GOSUB wrtcmd
ELSE
    PRINT "** MASK ROUTE LENGTH =8 **"
    '
    INPUT "CHANNEL SET DATA BIT8 (8ch)-->BIT1 (1ch)  [0/1] ";sch1$
    wrt$ = "MSK 0;SCH #B"+sch1$ : GOSUB wrtcmd
    '
END IF
'
RETURN
```

```
Dsply:  '---------- Display set data ----------

        wrt$="SCH?"    : GOSUB wrtcmd
        GOSUB readcmd : PRINT MID$(rd$,1,20); : PRINT "   ( 8ch~1ch)"

        IF MID$(rd$,5,1)="1" THEN
            GOSUB readcmd : PRINT MID$(rd$,1,20); :PRINT "   (16ch~9ch)"
        END IF

        PRINT

        RETURN

gpinit: ' ----------  Setup GPIB interface  ----------

      CALL IBFIND("GPIB0", GPIB0%)          ' Open device (GPIB0)
      IF GPIB0% < 0 THEN GOTO trap          ' system error

      CALL IBFIND("ED", ED%)                ' Open device (ED)
      IF ED% < 0 THEN GOTO trap             ' system error

      CALL IBSIC(GPIB0%)                    ' Interface clear
      IF IBSTA% < 0 THEN GOTO trap          ' system error

      tim = 0.5
      GOSUB waidly

      CALL IBCLR(ED%)                       ' Device clear

      RETURN

wrtcmd: ' ----------  Write command  ----------

      wrt$=wrt$+chr$(13)+chr$(10)
      CALL IBWRT(ED%, wrt$)                 ' Write command
      IF IBSTA% < 0 THEN GOTO trap          ' Trap

      RETURN

readcmd:' ----------  Read command  ----------

      rd$=SPACE$(22)
      CALL IBRD(ED%, rd$)                   ' Read command
      IF IBSTA% < 0 THEN GOTO trap          ' Trap

      RETURN

waidly: ' ----------  Wait delay  ----------

      stm = TIMER
      etm = TIMER
      WHILE etm - stm < tim
          etm = TIMER
          IF etm < stm THEN etm = etm + 86400
      WEND

      RETURN

trap:   ' ----------  System trap  ----------

      PRINT "IBERR%:" + STR$(IBERR%)
      STOP

      END
```

(12) Setting the number of mark ratio AND bit shifts

This program sets the number of mark ratio AND bit shifts.

The number of mark ratio AND bit shifts cannot be set for programmable patterns. If attempted, the information for the pattern will be displayed on the CRT and program execution will be aborted.



Note:

See the beginning of paragraph 15.5 for *1 and *2.

Program listing

```
' ************************************************************
' *                                                          *
' *              MP1702A / MP1609A / MP1651A                 *
' *         MARK RATIO  AND BIT SHIFT  SAMPLE SOFT           *
' *                                                ED_MARK   *
' ************************************************************
'
'------------------------------------------------------------'
'                          MAIN  ROUTINE                     '
'------------------------------------------------------------'
'
common shared IBSTA%,IBERR%,IBCNT%      ' Setup GPIB-PC functions
GOSUB gpinit                            ' Setup GPIB interface
'
DO
    CLS
    PRINT "* MP1702A/MP1609A/MP1651A  MARK RATIO  AND BIT SHIFT *"
    PRINT
    '
    GOSUB Ptn                           ' Check pattern mode
    '
    GOSUB Mark
    '
    INPUT "NEXT DATA SET [ YES=0 , NO=1 ]";loop$
    PRINT
    '
LOOP UNTIL loop$="1"
'
STOP
'
'------------------------------------------------------------'
'                          SUB  ROUTINE                      '
'------------------------------------------------------------'
    '
Ptn:    ' ----------  Check pattern mode ----------
    '
    wrt$ = "PTN?" : GOSUB wrtcmd        ' REQUEST Pattern mode ?
    GOSUB readcmd :ptn$=MID$(rd$,5,1)
    '
    IF ptn$="0" OR ptn$="1" THEN
        GOSUB Ero                       ' Word/Data mode-->Error
        PRINT "** PROGRAM STOP ** "
        STOP
    END IF
    '
    RETURN
    '
Mark:   ' ----------  Set Mark ratio bit shift ----------
    '
    DO
      INPUT "MARK RATIO [ 1 BIT Shift=0,  3 BIT Shift=1 ] ";sft
    LOOP UNTIL sft=0 OR sft=1
    '
    wrt$ = "SFT " +STR$(sft)    : GOSUB wrtcmd
    wrt$="SFT?" : GOSUB wrtcmd : GOSUB readcmd
    '
    IF MID$(rd$,5,1)="0" THEN
     rd$="1"
    ELSE
      rd$="3"
    END IF
    '
```

```
    PRINT "MARK RATIO BIT SHIFT= "+rd$
    '
    RETURN
    '
Ero:  ' ----------  Error --> Display Pattern data  ----------
    '
    wrt$="FRM?" : GOSUB wrtcmd : GOSUB readcmd
    '
    IF ptn$="0" THEN
        '
        PRINT "ERROR !!  Pattern mode = PROG.WORD "
        PRINT
        '
        IF MID$(rd$,1,5)="FRM 0" THEN
            '
            wrt$ = "WNB?" : GOSUB wrtcmd : GOSUB readcmd
            PRINT "NUMBER OF WORD ="+MID$(rd$,4,6)
            '
            wrt$ = "WLN?" : GOSUB wrtcmd : GOSUB readcmd
            PRINT "WORD LENGTH ="+MID$(rd$,4,3)
            '
        ELSE
            GOSUB Fln
        END IF
        '
    ELSE
        PRINT "ERROR !!  Pattern mode = PROG.DATA "
        PRINT
        '
        IF MID$(rd$,1,5)="FRM 0" THEN
            '
            wrt$ = "DLN?" : GOSUB wrtcmd : GOSUB readcmd
            PRINT "DATA LENGTH ="+MID$(rd$,4,7)
            '
        ELSE
            GOSUB Fln
        END IF
        '
    END IF
    '
    wrt$ = "PAG?" : GOSUB wrtcmd : GOSUB readcmd
    PRINT "PAGE ="+MID$(rd$,4,6)
    '
    RETURN
    '
Fln:  '---------- Frame length mode ----------
    '
    wrt$="FLN?" : GOSUB wrtcmd : GOSUB readcmd
    PRINT "FRAME LENGTH ="+MID$(rd$,4,3)
    '
    RETURN
    '
gpinit:  ' ----------  Set up GP-IB functions  ----------
    '
    CALL IBFIND("GPIB0", GPIB0%)          ' Open device (GPIB0)
    IF GPIB0% < 0 THEN GOTO trap          ' system error
    '
    CALL IBFIND("ED", ED%)                ' Open device(ED)
    IF ED% < 0 THEN GOTO trap             ' system error
    '
    CALL IBSIC(GPIB0%)                    ' Interface clear
    IF IBSTA% < 0 THEN GOTO trap          ' system error
    '
```

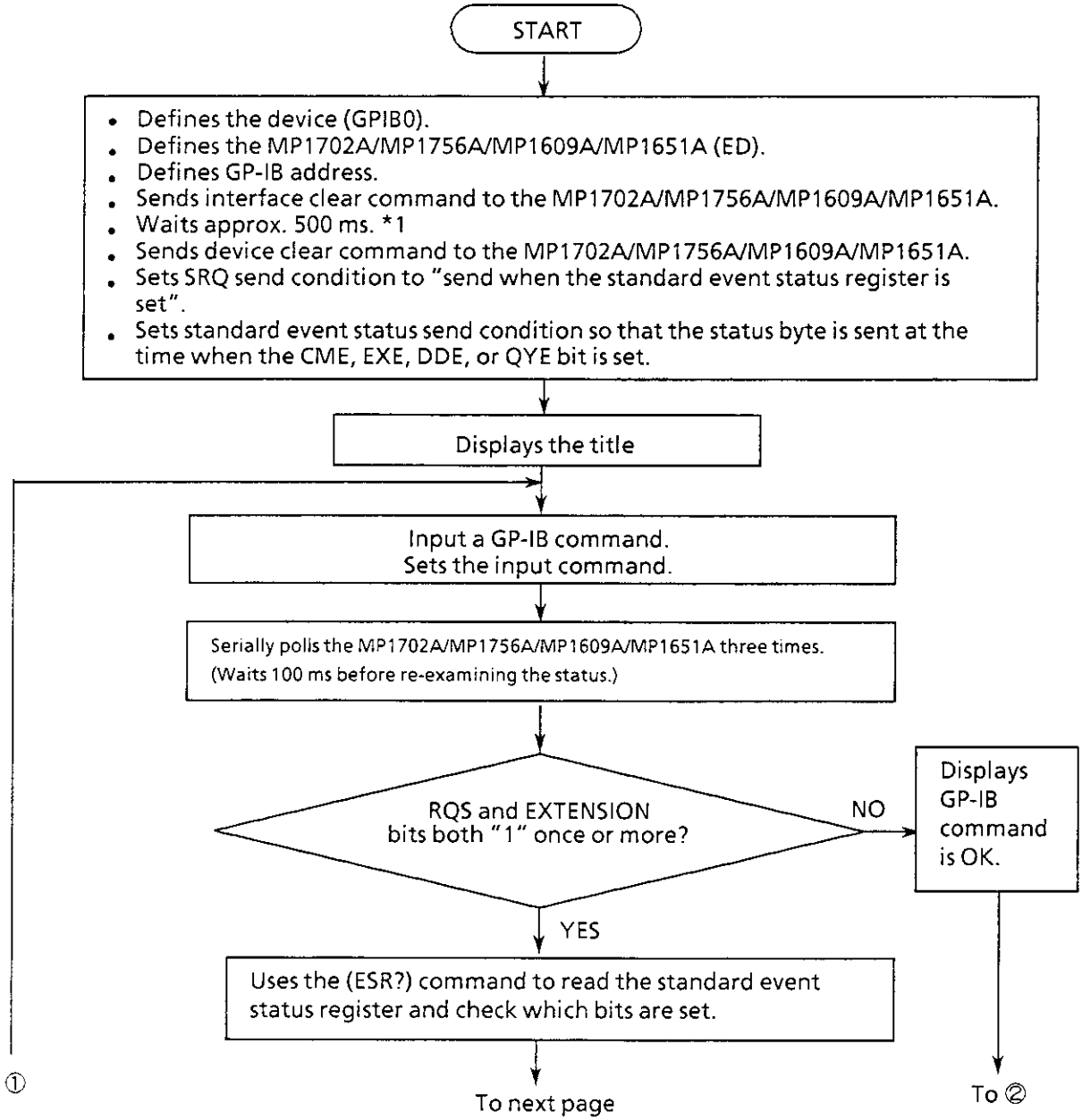
```
      tim = 0.5
      GOSUB waidly
      '
      CALL IBCLR(ED%)                        ' Device clear (ED)
      '
      RETURN
           '
wrtcmd:  ' ----------  Write command  ----------
           '
      wrt$=wrt$+chr$(13)+chr$(10)
      CALL IBWRT(ED%, wrt$)                  ' Write command
      IF IBSTA% < 0 THEN GOTO trap           ' Trap
      '
      RETURN
           '
readcmd: ' ----------  Read command  ----------
           '
      rd$= SPACE$(12)
      CALL IBRD(ED%, rd$)                    ' Read command
      IF IBSTA% < 0 THEN GOTO trap           ' Trap
      '
      RETURN
           '
waidly:  ' ----------  Wait delay  ----------
           '
      stm = TIMER
      etm = TIMER
      '
      WHILE etm - stm < tim
          etm = TIMER
          IF etm < stm THEN etm = etm + 86400
      WEND
      '
      RETURN
           '
trap:    ' ----------  System trap  ----------
           '
      PRINT "IBERR%:" + STR$(IBERR%)
      '
      STOP
      '
      '
      END
```

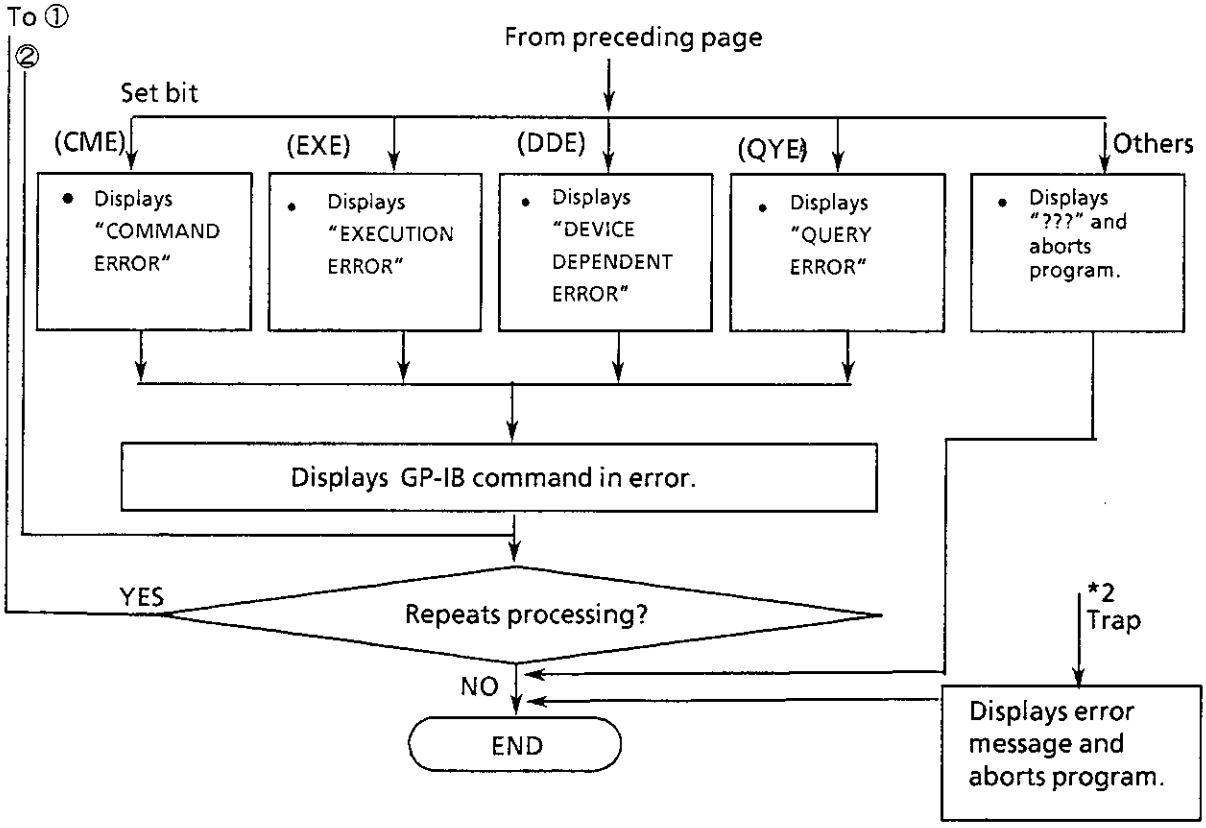
(13) Standard event status byte check (COMMAND ERROR bit check by serial polling)

This program checks the CME, EXE, DDE, and QYE bits of the standard event status byte, and displays whether the input GP-IB command is correct or not.

When the command is incorrect, the cause of the error is displayed.

The GP-IB status byte is monitored by serial polling and by the status of the standard event status byte is checked by (ESR?) data request command.





Note:

See the beginning of paragraph 15.5 for *1 and *2.

Program listing

```
' ************************************************************
' *                                                          *
' *              MP1702A / MP1609A / MP1651A                 *
' *     STANDARD EVENT STATUS REGISTER CHECK  SAMPLE SOFT    *
' *                                                 ED_ESE   *
' ************************************************************
'
'------------------------------------------------------------'
'                          MAIN  ROUTINE                     '
'------------------------------------------------------------'
'
common shared IBSTA%,IBERR%,IBCNT%    ' Setup GPIB-PC functions
GOSUB gpinit                          ' Setup GPIB interface
'
wrt$ = "SRQ 32"  : GOSUB wrtcmd       ' SRQ : Command error bit
wrt$ = "ESE 60"  : GOSUB wrtcmd       ' CME,EXE,DDE,QYE bit
'
PRINT "* MP1702A/MP1609A/MP1651A STANDARD STATUS REGISTER CHECK *"
PRINT
'
DO
    INPUT " INPUT ANY GP-IB COMMAND,Please.",com$
    wrt$ =com$  : GOSUB wrtcmd
    '
    GOSUB Check
    '
    INPUT " NEXT COMMAND SET?  [ YES=0 , NO=1 ]";loop$
    PRINT
    '
LOOP UNTIL loop$="1"
'
STOP
'
'------------------------------------------------------------'
'                          SUB  ROUTINE                      '
'------------------------------------------------------------'
    '
Check:  '----------  Serial polling  ----------
    '
    byt=0
    '
    FOR I=0 TO 2
        '
        CALL IBRSP(ED%,SPR%)                ' Serial polling
        IF IBSTA% < 0 THEN GOTO trap
        '
        s=SPR% AND &H40
        q=SPR% AND &H20
        '
        tim = 0.1 : GOSUB waidly
        '
        IF s=&H40 AND q=&H20 THEN byt=96   ' ROS,Comand Error bit=1
        '
    NEXT I
    '
    IF byt=96 THEN
        GOSUB Ero
    ELSE
        PRINT " GP-IB COMMAND IS OK !"
    END IF
    '
    RETURN
```

```
Ero:    ' ---------- ERROR ----------

    wrt$ = "ESR?" : GOSUB wrtcmd    ' REQUEST standard event status?
    GOSUB readcmd                   ' READ standard event status
    '
    IF MID$(rd$,9,1)="1" THEN PRINT "* COMMAND ERROR !!"
    IF MID$(rd$,10,1)="1" THEN PRINT "* EXECUTION  ERROR !!"
    IF MID$(rd$,11,1)="1" THEN PRINT "* DEVICE DEPENDENT ERROR !!"
    IF MID$(rd$,12,1)="1" THEN PRINT "* QUERY ERROR !!"
    PRINT " INPUT COMMAND = "+com$
    '
    IF MID$(rd$,7,1)="1" OR MID$(rd$,8,1)="1" OR MID$(rd$,13,1)="1" OR MID$(rd$,
14,1)="1"  THEN GOSUB Bug
    '
    RETURN
    '
gpinit: ' ---------- Setup GPIB interface ----------
    '
    CALL IBFIND("GPIB0", GPIB0%)        ' Open device (GPIB0)
    IF GPIB0% < 0 THEN GOTO trap        ' system error
    '
    CALL IBFIND("ED", ED%)              ' Open device (ED)
    IF ED% < 0 THEN GOTO trap           ' system error
    '
    CALL IBSIC(GPIB0%)                  ' Interface clear
    IF IBSTA% < 0 THEN GOTO trap        ' system error
    '
    tim = 0.5
    GOSUB waidly
    '
    CALL IBCLR(ED%)                     ' Device clear (ED)
    '
    RETURN
    '
wrtcmd: ' ----------  Write command  ----------
    '
    wrt$=wrt$+chr$(13)+chr$(10)
    CALL IBWRT(ED%, wrt$)               ' Write command
    IF IBSTA% < 0 THEN GOTO trap        ' Trap
    '
    RETURN
    '
readcmd: ' ----------  Read command  ----------
    '
    rd$=SPACE$(16)
    CALL IBRD(ED%, rd$)                 ' Read command
    IF IBSTA% < 0 THEN GOTO trap        ' Trap
    '
    RETURN
    '
waidly: ' ----------  Wait delay  ----------
    '
    stm = TIMER
    etm = TIMER
    '
    WHILE etm - stm < tim
        etm = TIMER
        IF etm < stm THEN etm = etm + 86400
    WEND
    '
    RETURN
    '
```

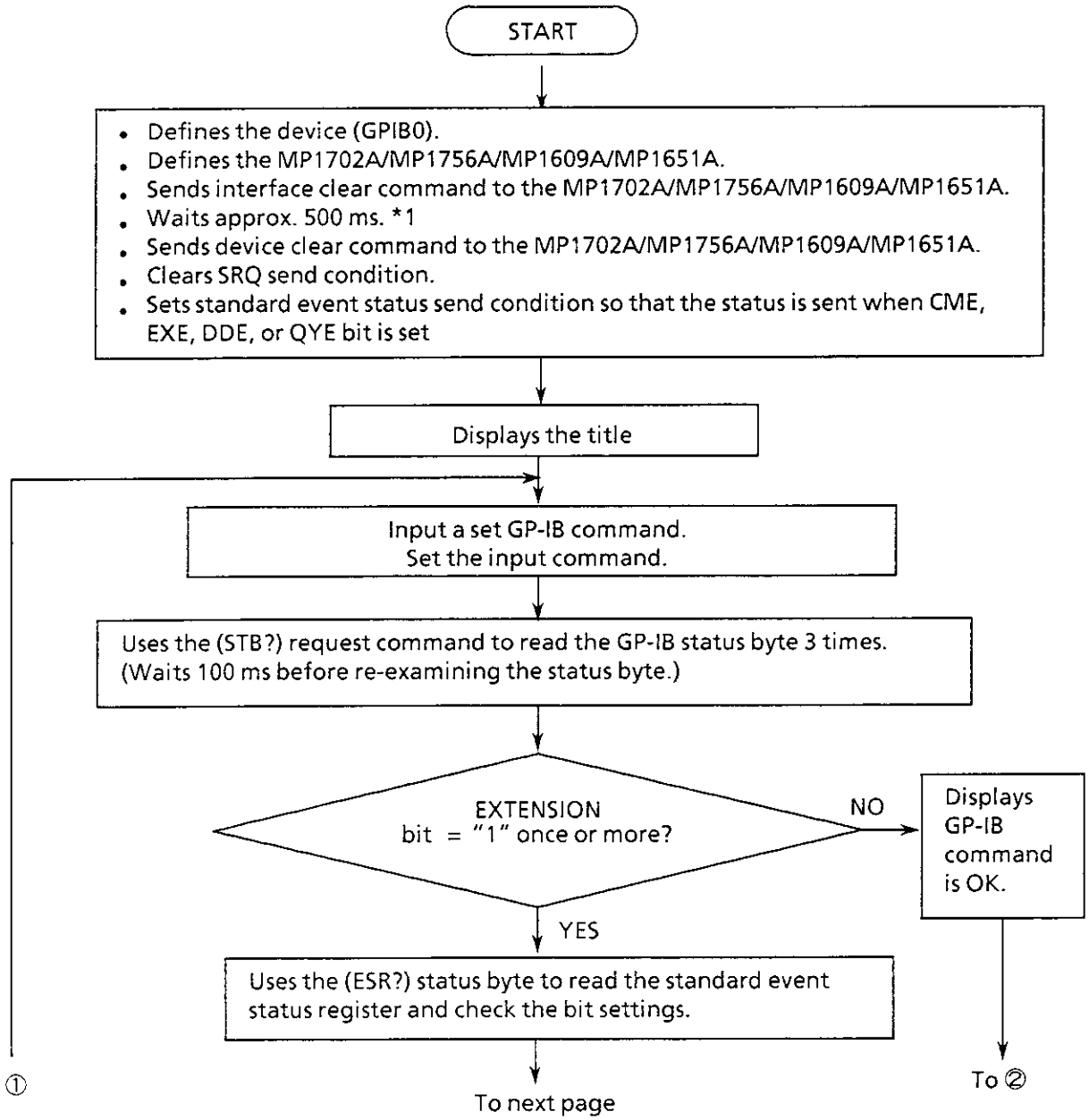
```
Bug:    ' ----------  BUG  ----------
        '
    PRINT "???"
        '
    STOP
        '
trap:   ' ----------  System trap  ----------
        '
    PRINT "IBERR%:" + STR$(IBERR%)
    STOP
        '
        '
    END
```

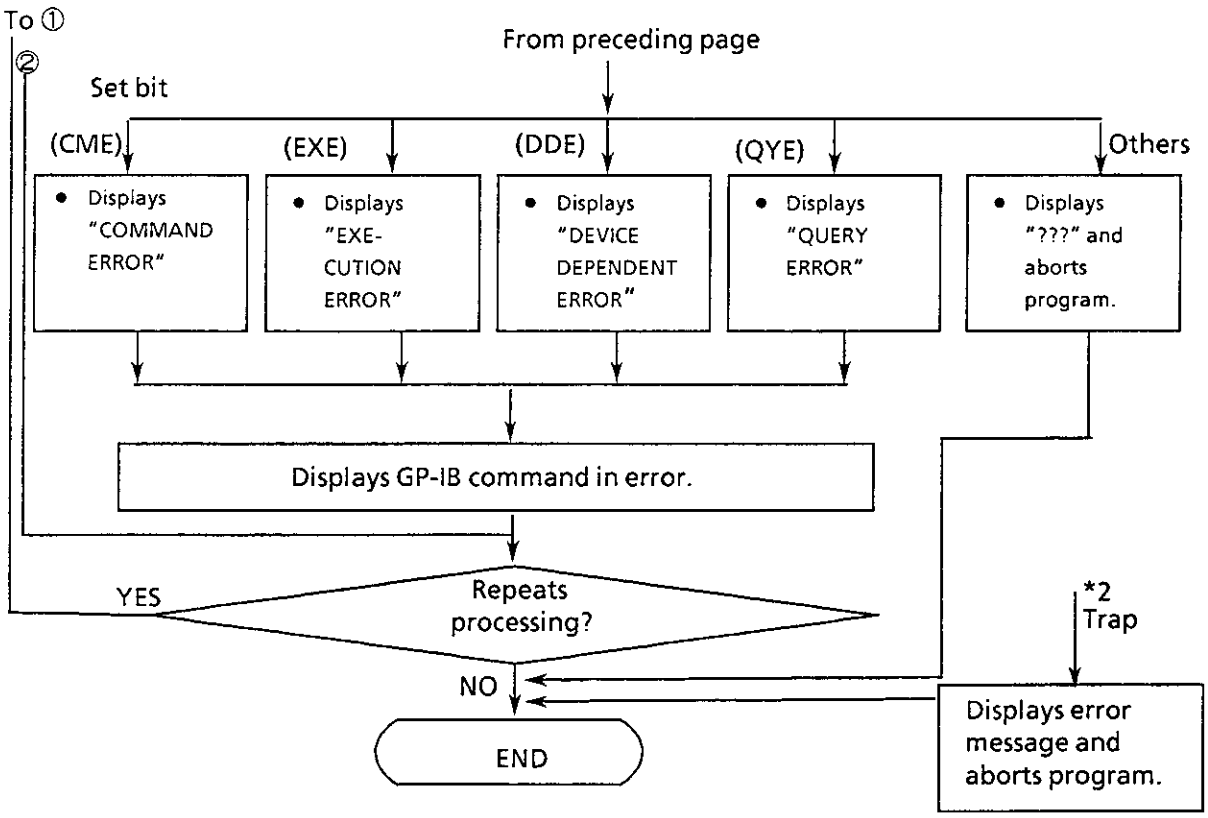
(14) Standard event status byte check (COMMAND ERROR bit check by (STB?) request command)

This program checks the CME, EXE, DDE, and QYE bits of the standard event status byte, and displays whether the input GP-IB command is correct or not.

When the command is incorrect, the cause of the error is displayed.

The GP-IB status byte and the standard event status byte are both checked by data request commands.





Note:

See the beginning of paragraph 15.5 for *1 and *2.

Program listing

```
' ************************************************************
' *                                                          *
' *              MP1702A / MP1609A / MP1651A                 *
' *     STANDARD EVENT STATUS REGISTER CHECK  SAMPLE SOFT    *
' *                                                ED_ESE2   *
' ************************************************************
'
'------------------------------------------------------------'
'                          MAIN  ROUTINE                     '
'------------------------------------------------------------'
'
common shared IBSTA%,IBERR%,IBCNT%    ' Setup GPIB-PC functions
GOSUB gpinit                          ' Setup GPIB interface
'
wrt$ = "SRQ 0 "   : GOSUB wrtcmd      ' SRQ :OFF
wrt$ = "ESE 60"   : GOSUB wrtcmd      ' CME,EXE,DDE,QYE bit
'
PRINT "* MP1702A/MP1609A/MP1651A  STANDARD STATUS REGISTER CHECK *"
PRINT
'
DO
  INPUT " INPUT ANY GP-IB COMMAND,Please.",com$
  wrt$ =com$   : GOSUB wrtcmd
  '
  GOSUB Check
  '
  INPUT " NEXT COMMAND SET?  [ YES=0 , NO=1 ] ";loop$
  PRINT
  '
LOOP UNTIL loop$="1"
'
STOP
'
'------------------------------------------------------------'
'                          SUB  ROUTINE                      '
'------------------------------------------------------------'
    '
Check:  '----------  Check Status byte  ----------
    '
  byt$=SPACE$(16)
  FOR I=0 TO 2
    '
    wrt$="STB?" : GOSUB wrtcmd    ' REQUEST Status byte register?
    GOSUB readcmd                 ' READ Status byte register
    '
    IF MID$(rd$,9,1)="1" THEN byt$=rd$    ' Command error bit = 1
    '
    tim=0.1 : GOSUB waidly
    '
  NEXT I
  '
  IF MID$(byt$,9,1)="1" THEN
     GOSUB Ero
  ELSE
     PRINT " GP-IB COMMAND IS OK !"
  END IF
  '
  RETURN
      '
```

```
Ero:     ' ----------  ERROR  ----------
         '
     wrt$ = "ESR?" : GOSUB wrtcmd    ' REQUEST Standard event status?
     GOSUB readcmd                   ' READ Standard event status
     '
     IF MID$(rd$,9,1)="1" THEN PRINT "* COMMAND ERROR !! "
     IF MID$(rd$,10,1)="1" THEN PRINT "* EXECUTION  ERROR !! "
     IF MID$(rd$,11,1)="1" THEN PRINT "* DEVICE DEPENDENT ERROR !! "
     IF MID$(rd$,12,1)="1" THEN PRINT "* QUERY ERROR !!  "
     PRINT " INPUT  COMMAND = "+com$
     '
     IF MID$(rd$,7,1)="1" OR MID$(rd$,8,1)="1" OR MID$(rd$,13,1)="1" OR MID$(rd$,
14,1)="1"  THEN GOSUB Bug
     '
     RETURN
          '
          '
gpinit:  ' ----------  Setup GPIB interface  ----------
          '
     CALL IBFIND("GPIB0", GPIB0%)        ' Open device (GPIB0)
     IF GPIB0% < 0 THEN GOTO trap        ' system error
     '
     CALL IBFIND("ED", ED%)              ' Open device(ED)
     IF ED% < 0 THEN GOTO trap           ' system error
     '
     CALL IBSIC(GPIB0%)                  ' Interface clear
     IF IBSTA% < 0 THEN GOTO trap        ' system error
     '
     tim = 0.5
     GOSUB waidly
     '
     CALL IBCLR(ED%)                     ' Device clear
     '
     RETURN
          '
wrtcmd:  ' ----------  Write command  ----------
          '
     wrt$=wrt$+chr$(13)+chr$(10)
     CALL IBWRT(ED%, wrt$)               ' Write command
     IF IBSTA% < 0 THEN GOTO trap        ' Trap
     '
     RETURN
          '
readcmd: ' ----------  Read command  ----------
          '
     rd$=SPACE$(16)
     CALL IBRD(ED%, rd$)                 ' Read command
     IF IBSTA% < 0 THEN GOTO trap        ' Trap
     '
     RETURN
          '
waidly:  ' ----------  Wait delay  ----------
          '
     stm = TIMER
     etm = TIMER
     '
     WHILE etm - stm < tim
          etm = TIMER
          IF etm < stm THEN etm = etm + 86400
     WEND
     '
     RETURN
          '
```

```
Bug:    ' ----------  BUG  ----------
        '
    PRINT "???"
    '
    STOP
        '
trap:   ' ----------  System trap  ----------
        '
    PRINT "IBERR%:" + STR$(IBERR%)
    STOP
    '
    '
    END
```