ortofon

INSTRUMENTATION MANUAL

for

ORTOFON MEASURING COMPUTER P400

Software Revision 2D

ORTOFON INSTRUMENTS A/S
Copenhagen, 4th September, 1987

0.0 List of contents

15.0 The P400 Commands:

15.1	AA	Manual/auto ch A	
15.2	AB	Manual/auto ch B	
15.3	BC	Buffer clear	
15.4	BD	Balance deviation	(NOT L-VERSION)
15.5	BP	Backpanel	
15.6	BR	Breakpoint	
15.7	CA	Arm ch A	
15.8	CB	Arm ch B	
15.9	CC	Colour code	
15.10	CD	Connect dots	
15.11	CO	Curves ON/OFF	
15.12	CP	Compressor	
15.13	CS	Curve summations	(NOT L-VERSION)
15.14	CT	Curve test	
15.15	CX	Set cursor	
15.16	DB	Debugger picture	
15.17	DC	Decimation count	
15.18	DI	Display function	
15.19	DS	Dump screen	
15.20	EA	Equalizing ch A	
15.21	EB	Equalizing ch B	
15.22	FA	Filter ch A	
15.23	FB	Filter ch B	
15.24	FC	Frequency channel	
15.25	FM	Function mode	
15.26	GA	Gain ch A	
15.27	GB	Gain ch B	
15.28	GC	Get curves	
15.29	HC	Hard copy	
15.30	IC	Index count	
15.31	IT	Internal trig	
15.32	KE	Keybox functions	
15.33	LA	Lin/log ch A	
15.34	LB	Lin/log ch. B	
15.35	LT	List test results	
15.36	LF	Internal trig	
15.37	LV	Internal trig	
15.38	MD	Delete curves from cassette	
15.39	ML	Load curves from cassette	
15.40	MO	Move option	
15.41	MS	Store curves on cassette	
15.42	OA	Output amplifier functions	
15.43	OF	Output frequency	
15.44	OM	Output mode	
15.45	OT	Output attenuator	
15.46	OW	Sweep rate	
15.47	PP	Parallel poll	
15.48	PS	Printer status	
15.49	PT	Polarity test	

1.0 INTRODUCTION

1.1 Instrumentation

P400 is intended to be a highly sophisticated instrument which in itself can manage most test problems occurring in test procedures with manufacturers of acoustical/electric transducers.

To cover test problems which cannot be managed by P400 alone, P400 can be connected to external computers and instruments, when EXTCON option is installed.

This connection is made according to the IEEE-standard, and this manual covers the use of these features.

This standard is supplied under several names but they are all alike or almost alike.

Following names and others may be used:

HP-IB
GPIB
IEEE BUS
ASCII BUS
PLUS BUS

Following organizations have approved the standard under following numbers:

IEEE 488-1978 with later revisions
ANSI MC1.1, identical to IEEE 488-1978
IEC 625-1, identical to IEEE 488-1978 except for connectors
B.S. 6146, British standard, identical to IEC 625-1

P400 uses IEEE 488-1978 connectors.

1.2 Control and Statistics

The reasons for adding external computing power to P400 may be many:

- Test sequence depending on test results
- More parameters in test sequence (temperature, humidity etc.)
- Automatic statistical use of test results
- Use of P400 with automatic systems and robottics
- Use of P400 with other equipment for synchronizing
- User defined test sequences

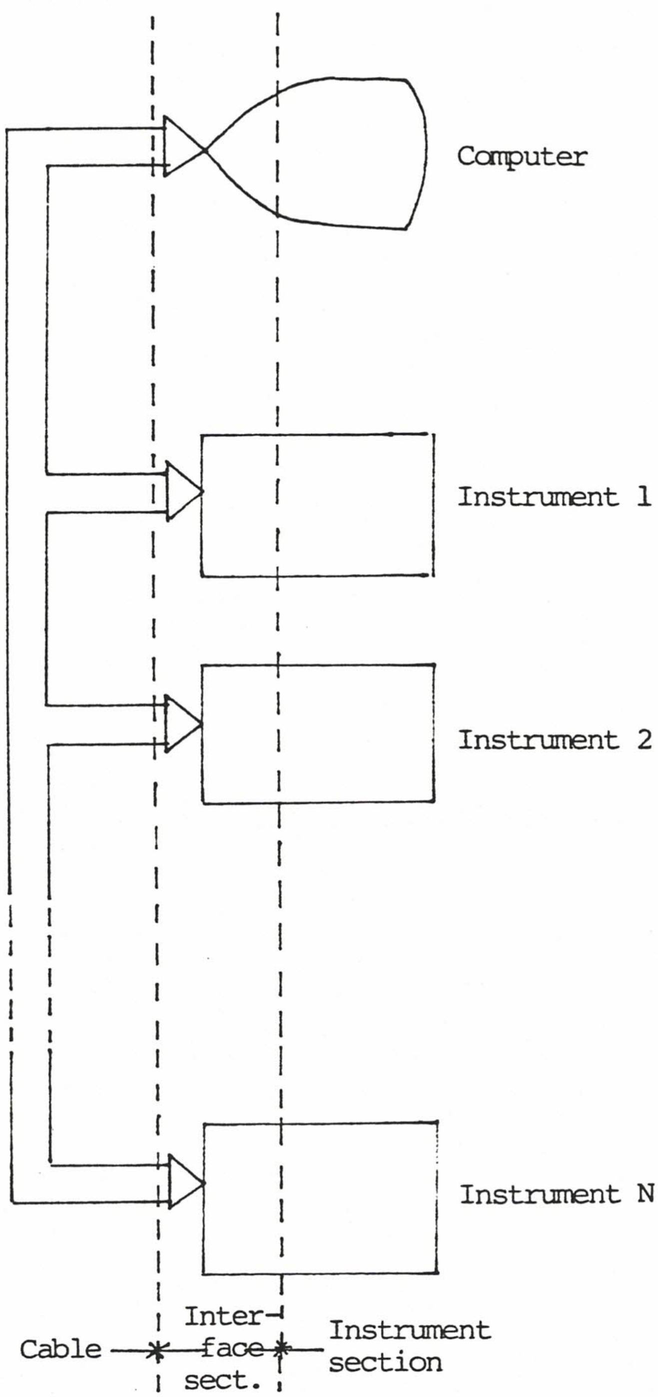
The IEEE software and hardware package provides a flexible tool to make all mentioned above functionable.

2.0 THE IEEE-STANDARD

2.1 Hardware

Let us consider a system where several instruments are connected to a computer.

Fig. 1:



The standard covers the connecting cables and the sections in the computer and the sections in the instruments which are connected to the cables.

Furthermore, the standard covers the definitions of some essential commands on the bus.

At this point, three essential terms will be defined:

1. Controller: The device which is controlling the information flow on the bus system. This is done by sending commands and control signals on the bus and by deciding which instrument is to send and which one to receive data.

Multiple Controller Systems can be implemented but are not covered here.

2. Listener: A device which is receiving data from other instruments or from the controller. Please note that the controller can be listener according to this definition and that several instruments in the system can be listeners at the same time.

3. Talker: A device which is sending data to the system. Please note that one device only can be controller at the same time.

The data flow on the bus is controlled by special signals in the cabling so that the user normally do not need to pay attention to this.

Other signals in the cabling control essential commands and requests and other signals control ordinary data and addressing information.

2.2 Information Hierarchy

When the controller issues a command on the bus system, it may be designated to be understood by all devices in the system. Commands of this kind are called general commands and fill the upper level in the information hierarchy.

Other commands may be designated only to one of the instruments on the bus. Commands of this kind are called addressed commands and fill the second level in the information hierarchy.

Further commands may be necessary in implementing a controlled system. These are the commands that are not defined by the standard but are dependent on the function of the single instrument. These commands are considered general data to the bus system and command interpretation is done in the instrument.These commands are called instrument commands and fill level 3 in the information hierarchy.

Sometimes pure data files may be transferred. These data transfers fill level 4 in the information hierarchy.

Fig. 2:

THE INFORMATION HIERARCHY
1. General commands
2. Addressed commands
3. Instrument interpretable commands
4. Data files

3.0 CONTROLLER CONSIDERATIONS

3.1 Service Request Ability

The controller must be carefully chosen. A great advantage of the controller is the ability to react and understand the service request applied upon it by P400. Service request is the standardized way for a peripheral instrument to request controller assistance.

If used with complicated test programmes, this facility is essential.

3.2 Data Transmission

The form in which data is transferred is not well standardized. The standard considers all data as binary codes without limitations and the end-of-data information is supplied through a special wire (EOI-line) in the cable and this signal must appear together with the last byte of information.

As an opposite to this simple concept, controllers have a great variety in the rules to end a data transmission. For P400 it is essential that the EOI line is used (some controllers consider a CR code or an LF code) as an end of transmission indication.

Some controllers use double indication of end-of-transmission, i.e. together with the EOI signal a special data code must occur, mostly a CR code.

P400 can handle end-of-transmission by EOI line alone or EOI line together with a special data code. These data codes need not be the same for input or output transmissions and for oscillator or binary data. The codes can be programmed in P400.

Some controllers use a triple indication of end-of-transmission. In these cases probably a double data byte LF code followed by CR code and with the EOI signal together with the last byte. Handling of these situations by P400 will not always be possible.

3.3 Computing Power

The computing power of the controller is to be calculated from the following items:

- The complexity of test programmes
- The number of peripherals to the system
- The amount of data flow on system

Following guidelines will be given here, but please note that they are guidelines and are not to be regarded as anything else.

8-bit Machine with 64k-RAM: 1 P400 with a complex programme. If many curves are to be transferred, add a hard-disc to controller.

16-bit Machine with Plenty of RAM and Hard-Disc: 4 P400 with complex programmes. If heavy calculations are performed, add an arithmetic co-processor.

Mini with Multiple Bus Features: 4 to 6 P400 on a single bus.

Attention should be payed to the fact that programming complex test programmes in a multi-P400 environment very well may require a well-trained programming crew and a considerable investment in time consumption.

4.0 CONNECTING

4.1 The Address Switches

The rear panel of the P400 has a switch-battery which contains 8 different switches. Prior to connecting P400 to the controller, the user must make sure that these switches are set correctly.

5 of the 8 switches are marked A1, A2, ..., A5. These switches are used to select the address, used in addressed commands and for transmitting commands and information (cf chapter 2.2 Information Hierarchy). Addresses are calculated in a binary way according to following tables:

A1 counts	1
A2 counts	2
A3 counts	4
A4 counts	8
A5 counts	16

If you wish to allocate P400 to address No. 11, switch on (up) A4 (counts 8), A2 (counts 2), and A1 (counts 1).

P400 has no limitation in address allocation but the standard uses address No. 31 for special purposes. Further restrictions can be issued from controller. Refer to controllers manual for information. No peripheral or controller connected to the bus can have the same address.

Switches Nos. 6 and 7 are normally used to configure peripheral to be a talk-only device or a listen-only device but P400 has no functions where this can be used and the switches are not used.

Switch No. 8 is used to set P400 to perform service requests to controller or not. It is recommended to use P400 in Service Request Mode, even if controller cannot handle service requests. Excluding this function makes it impossible to read in SPOLL-register if an error has occurred and the user must read output buffer instead (refer to chapter 8, 10, and 11).

Only if, for some reason, you cannot apply service requests to the bus, you may have to exclude.

4.2 Cabling

It is recommended only to use the cabling defined in the standard. If not, noise problems may occur. In industrial environment this can cause problems. For the cable lengths, the standard issues the following restrictions: maximum length is 2 m per device on the bus, in total maximum 20 m. If you connect 1 P400 to 1 controller, the maximum cable length is 4 m.

For the same reason P400 has no internal ground connections as ground connections are always in the controller.

If P400 is not connected to controller, the start-up switch settings on self-test picture may be incorrect.

5.0 GENERAL COMMANDS

5.1 The Nature of General Commands

According to the definition in chapter 2 the general commands are commands designated to be understood by all devices in the bus system. Most important in the general command group are the following commands:

- Parallel Poll
- Go to local
- Device clear
- Trigger

Refer to chapter 14 for further information on these commands.

5.2 Test Programme for DEVICE CLEAR

1: Connect the controller and the P400 as described in chapter 4
2: Power on controller
3: Power on P400
4: Wait while P400 performs self-test
5: When self-test is ended, issue general command Device Clear.
6: Observe that P400 enters normal mode as if pressing START button
7: Issue Device Clear command again
8: Observe that P400 starts up self-test procedure again
9: Start again in item 4

If the Device Clear command works in this context, there is a great probability that the connection is correct. If it does not work,there is a fundamental error in communication, probably caused by bad cabling or by wrong controller initialization.

For a full description of power-up procedures, refer to chapter 16.

6.0 ADDRESSED COMMANDS

6.1 Nature of Addressed Commands

According to the definition in chapter 2 the addressed commands are designated to be understood by one peripheral only in the system.

Most important in the addressed command group are the following commands:

- Go to remote
- Local lock-out
- Selected clear
- Selected trigger

Refer to chapter 14 for further information on these commands.

6.2 Test Programme for Correct Address

Use the set-up from chapter 5.2 with P400 in normal mode after item 6.

1: Issue addressed command GO TO REMOTE with the address selected on P400 back panel

2: Observe that the LED in LOCAL push button is turned OFF

3: Issue general command GO TO LOCAL

4: Observe that the LED in LOCAL push button is turned ON again

5: Now issue addressed command GO TO REMOTE with wrong addresses

6: Observe that nothing happens to the LED in the LOCAL push button

If this procedure works as described, P400 and controller are correctly connected. If not, there is a fundamental error in communication, probably caused by bad cabling or by wrong controller initialization.

7.0 PASSING COMMANDS TO P400

7.1 Free Input Form

Level 3 in the information hierarchy is the P400 interpretable commands. To the bus system and the controller, these transfers are a part only of the many data transfers possible and there is no physical difference between P400 interpretable commands and ordinary data files.

To accept input command strings P400 must be in remote mode. (LED in LOCAL push button turned OFF). If P400 has not requested input data files due to input command strings, all incoming data will be interpreted as commands.

Controllers may terminate transmissions in several ways. P400 is designed to accept a great freedom in input command terminations. Following terminations are allowed together with EOI signal:

1 No special character
2 cr (ASCII $0D)
3 lj (ASCII $0A)
4 , (ASCII $2C)
5 ; (ASCII $3B)
6 Byte programmed in incode

Furthermore the most common combinations of the above are allowed. Only in very special cases it is necessary to programme incode in the first programming sequences.

7.2 Test Programme for Input Commands

Perform start-up sequence as described in chapter 5 and 6. Leave P400 in remote mode.

1: Send the following command string to P400: "FM3,DBE"
2: Observe that P400 is entering Frequency response mode and that debugger picture is shown on the screen.

If no green error line is shown on screen, everything is working well.

If a transmission mismatch error is shown, it may be necessary to programme incode character by a TT1-xx command.

Other errors are probably caused by some error in command string transmitted or by wrong controller initialization.

Refer to chapter 15 for a full description of the P400 commands.

8.0 QUEUED DATA

8.1 Circular Buffers (FIFO buffers)

All command inputs to P400 and all output data from the P400 is communicated through buffers.

The buffers are organized in a FIFO (First In First Out) manner and this means that the information put into the buffer first, is also read out first. (Another way to organize a queue is the stack where the last information put into the buffer is read out first.)

8.2 Input Buffers

For input command strings this means that when the controller initiates a transfer to P400 containing P400 commands, all commands are kept in P400 memory until transfer is terminated. At this point P400 starts to interpret commands transferred.

This is done to reduce the time in which the P400 uses the bus and thereby increasing the time left to the controller for computing.

This organization introduces another problem in distincting command input and data input. In general, all inputs to P400 are considered commands when nothing else is requested.

Refer to chapter 11. "Error Handling" and 12. "Passing Data to P400" for further information about this problem.

8.3 Output Buffers/Pseudo Buffers

Each time P400 obtains data to deliver to controller, this data is put into the output queue system. The output data is left in P400 memory until controller initiates the output transfer. Output data is delivered to controller in a FIFO manner.

Output data may be large. A full curve from P400 screen may be 4096 bytes. P400 memory space is not big enough to contain this amount of data. To overcome this problem, output queue system contains two buffers.

The first buffer contains information related to output pointers, length information etc. This buffer is called "Output Queue Buffer". When output is curves, these pointers point to actual curve space in P400 memory thus avoiding to fill curve data in output buffers. When output data is not curves, the data is put into the second buffer. This buffer is called "Output Text Buffer" and pointers to the Output Text Buffer are in Output Queue Buffer.

For the user this double system should be transparent, except that the length of output buffer is undefined.

9.0 THE PROGRAMMING PHASE

9.1 The Debugger Picture

The P400 IEEE software is a state-oriented software product. This means that the allowed interaction to controller depends on the state of the P400. The state of P400 again depends on previous controller interaction on keyboard entries etc.

The states of P400 are determined by:

1: The input/output queue system
2: Certain flags in P400 memory describing previous events
3: The hardware connection between P400 and controller

As P400 allows very complex programming, the number of states is similar high.

In the programming phase of IEEE controller programmes, the P400 can produce a display on screen showing selected flags and variables covering the queues, the flags, and the registers of the interface adapter.

The debugger picture is turned on and off using the commands DBE and DBO(refer to chapter 15.15 for further information).

WARNING: The use of debugger picture facilities will drasticly reduce execution speed.

9.2 Picture Sections

To cover the items queues, flags and interface, the debugger picture is divided into sections called the queue section, the flag section, and the interfacing section.

9.2.1 The Queue Section

The queue section is the upper for lines of picture in green and red colours. As described in chapter 8, all the queues are circular buffers, and to organize this, all the 3 queues are left with 256 bytes of P400 memory, and to each queue is connected 2 pointers, one telling up to where the queue is filled, and the other telling up to where the queue is emptied. If the two pointers are equal, the queue is empty.

The upper red line is displaying the pointers connected to the buffers:

1 - 2	Input buffer
3 - 4	Output text buffer
5 - 6	Output queue buffer

The binary pointers are displayed in Hexadecimal code.

The second green line shows the first characters of input buffer. This is a character buffer, and any non-printable character is substituted *.

The third green line shows the first characters of the output text buffer. Similar to the input buffer any non-printable character is substituted *.

The last green line of queue section shows the output queue buffer. Each output uses 6 bytes of output queue buffer. The first 2 bytes of the 6 are the output type (refer to chapter 13 for a full description of the output types).

Bytes 3 and 4 are a pointer, either for an entry in Output text buffer if output is text and status information, or an address in actual P400 memory if the output is curves.

The last bytes are the length of the output transmission in bytes.

9.2.2 The Flag Section

White lines 5 - 8 show the flags that P400 uses to determine the allowed actions.

INPMIS: P400 has requested input from P400. All inputs are interpreted as data.
Allowed values 0 (input is not requested)
1 (input is requested)

SDCMIS: P400 is in an error situation or in a simulated error situation due to an error 3, USER REQUESTED BREAKPOINTS (for a full error description refer to chapter 11). If service request mode is selected on back panel switch, a service request is issued. The controller must perform the addressed bus command SELECTIVE DEVICE CLEAR.
Allowed values: 0 (P400 is not in error)
1 (P400 is in error)

OUTFL: P400 is performing an output transmission. Allowed value 0.

DTIFLG: Used when a status input is performed. Allowed value 0.

DBUGFL: Used when debugger picture is used. Allowed value 3.

INXCUR: Skip count used on output curve transfers. Allowed values 0-F.

KEYSEN: Used when keybox can be read from controller.
Allowed values 0 (keybox cannot be read)
1 (keybox can be read)

INCODE: Termination character for input transfers. All values allowed. Default value 0.

TXTCOD: Termination character for output text transfers. All values allowed. Default value 0.

DATCOD: Termination character for curve and status output transfers. All values allowed. Default value 0.

TIMCNT: Time limit for output transfers. All values allowed. Default value 10 mS = 0 on display.

DELAYV: Internal variable showing delay before measuring allowed values 0-1.

EXEFL: Flag showing whether P400 is working or idle:
1 = Working
0 = Idle

SPPIC: Internal representation of serial poll register. Contains always last write-out to interface spoll register.

$RQFLG: P400 has requested a service request to controller

LOCK01: The number of A curves in output buffer
LOCK02: The number of A ref.curves in output buffer
LOCK03: The number of B curves in output buffer
LOCK04: The number of B ref.curves in output buffer

INLNGT: The expected length of input transfers. Refers only to programme and curve transfers. For text transfer S = 0.

9.2.3 The Interface Section

The interface to the bus is performed via the interface adapter circuit MC68488 from Motorola. Refer to Motorola documentation for a full description of these registers. These registers are mostly used in error situations and not by the ordinary user.

All bits are 8-bit registers and are displayed in Hexadecimal code.

Following may be of general interest:

```
IEEER1:   |_____|_REM_|_LOK_|_____|_____|_____|_____|_____|
                    |     |
                    |     |____ Local lock-out enabled
                    |__________ Remote enabled
```

IEEER4: The back panel address switch of P400. This shows the P400 bus address in hex. code. If service request mode is selected, add hex. value $80.

IEEER5: The serial poll register (see chapter 10.2).

10.0 KEEPING P400 ON TRACK

10.1 Serial Poll

The controller frequently needs to determine the state of P400. The P400 keeps selected information in a special register. When output data transfers from the selected information, P400 is not performed. This register can always be read from the controller. When reading while output data transfers are performed, data is lost and an error is introduced.

The process of reading this register is called spolling and the register is called the spoll register.

The reading is initiated by the addressed bus command for a serial poll.Refer to chapter 14.1.9 for further information.

10.2 Spoll Register Description

The spoll register is an 8-bit register defined in the following way:

|	bit 7	(128/-127)	Busy/Ready
|	bit 6	(64)	SRQ is requested
MSB	bit 5	(32)	P400 not idle
|	bit 4	(16)	Output buffer not empty
|	bit 3	(8)	Input command buffer not empty
LSB	bit 2)	(4)	
|	bit 1)	(2)	Error codes 0-7
|	bit 0)	(1)	

Busy/Ready: Busy-bit is set during curve recording, otherwise cleared.

SRQ is requested: When P400 wants attention from controller, this bit is set (refer to chapter 4.1 for further information). If the controller can handle a service request from the P400, the setting of this bit will initiate a service request. The bit is cleared when controller performs a serial poll to P400.

P400 Not Idle: P400 is performing tasks as instructed in input command buffer. When P400 enters idle state, this bit is cleared.

Output Buffer not empty: When this bit is set, output buffer contains information, and a data transfer from P400 can be initiated from controller. P400 will perform the output transfer when execution of current command in input command buffer is terminated. When output buffer is empty, this bit is cleared.

Input Command Buffer not empty: While input command buffer still contains commands to be executed, this bit is set, otherwise cleared.

The "INPUT BUFFER NOT EMPTY" flag is not set until after a command transfer to P400 is finished. It may therefore be necessary to insert a BRx command in the command stream and wait for a service request.

Alternatively, insert a wait in the spoll loop.

Error Codes 0-7: Refer to chapter 11 for further information. Values are only valid when SRQ is requested - bit is set.

10.3 Parallel Poll and Service Requests

The standard describes a method called parallel poll to ask all peripherals on the bus for their states.

Naturally, the number of information must decrease. The answer obtainable is telling whether a peripheral bus tried a service request or not. If your controller is not answering to a service request, this feature may be useful because you can save some computing time in the controller when it is not necessary to serial poll all the peripherals.

The parallel poll information is cleared when the appropriate peripheral is serial polled.

10.4 Parallel Poll Configure/Unconfigure

Each peripheral has an 8-bit ppoll register. When the parallel poll is performed from controller, all these registers are or'ed to give the result to controller. The content of the ppoll register is either established in using the addressed bus commands "Parallel poll configure" or "Parallel poll unconfigure" if the controller is able to handle this, or by the P400 command PP. Refer to chapter 15 for further information.

11.0 ERROR HANDLING

11.1 Service Request Flags

When P400 requests service from the controller, the P400 stops execution of commands in input command buffer. This can occur when an unwanted error occurs in P400 or in a controlled manner as a response to correct programming.

11.1.1 Errors Continued on Device Clear

Some types of service requests require a Selective Device Clear from controller to continue programme execution. Except for bus commands, P400 accepts only read-out requests from output buffer.

11.1.2 Errors Continued on Input Data

When P400 needs data input, it opposes a service request to controller. Programme execution is continued when the input needs are fulfilled. Except for bus commands, P400 accepts read-out requests from output buffer and all inputs are treated as data.

11.2 Error Types

Although some service requests are controlled, we will name all requests as errors in this manual to retain standard terminology. When this is the case, P400 has 8 error types which are described below:

11.2.1 Programming Errors (Type 0)

Errors to occur when illegal or undefined character strings are placed into the input buffer or when an input buffer overflow occurs(on input strings longer than 229 characters.

P400 will make following actions:

- SRQ is requested
- Input buffer is cleared
- Output buffer is left for output
- P400 is cleared to state before the programming errors
- P400 continues on Selective Device Clear (SDC)

The debugger picture is not updated on "INTERNAL ERRORS" although the error text is displayed and a service request is issued.

11.2.2 Curve Lock Errors (Type 1)

A programming attempt, correct in itself, will cause curve data in output buffer lost if executed.

P400 makes following actions:

- SRQ is requested
- Buffers are left unchanged
- The command causing the error is retained to be executed when output buffer is emptied
- P400 continues on Selective Device Clear (SDC)

11.2.3 Output Buffer Overflow Errors (Type 2)

Output buffer is not properly emptied.

P400 will make following actions:

- SRQ is requested
- The buffers are left unchanged, but no guarantee can be issued that the last data put into buffer(s) is correct
- Current command is retained
- P400 continues on Selective Device Clear (SDC)

11.2.4 User Requested Breakpoints (Type 3)

The user has inserted a breakpoint command in input buffer or the user has used the keybox.

P400 will take the following action:

- SRQ is requested
- Buffers are left unchanged
- Current command is skipped
- Keybox active state is disabled
- P400 continues on Selective Device Clear (SDC)

11.2.5 Input Requested (Type 4)

P400 needs data to be transferred from controller.

P400 will take the following action:

- SRQ is requested
- Buffers are left unchanged
- Current command is skipped
- P400 continues when transfer is terminated. If Selective Device Clear (SDC) is issued, P400 enters start-up procedure.

11.2.6 Input Transmission Error (Type 5)

An input transmission is of incorrect length or terminates with an illegal character.

P400 will take following actions:

- SRQ is requested
- Buffers are left unchanged
- Current command is skipped
- P400 continues on Selective Device Clear (SDC)

11.2.7 Output Transmission Error (Type 6)

The output transmission is of incorrect length, terminates with an illegal character, or P400 times-out due to stop in transmission.

P400 will take following actions:

- SRQ is requested
- Input buffer is left unchanged
- Current command is retained
- Output transmission in error is skipped
- P400 continues on Selective Device Clear (SDC)

11.2.8 Illegal Controller Request (Type 7)
The controller has interrupted P400 during a data transfer.

P400 will take the following action:

- SRQ is requested
- Buffers are left unchanged
- Current command is retained
- P400 continues on Selective Device Clear (SDC)

12.0 PASSING DATA TO P400

12.1 EOI-Wire, Termination Character

Any data transfer on IEEE-bus is 8-bit parallel, using usual IEEE-handshake. However, to terminate a transmission a special wire is used. The EOI (End Of Information or End Or Identify)wire logically stops the current transmission. Some controllers add to the information an extra data-byte to send together with EOI-wire signal. This extra character is unnecessary information but cannot always be avoided. P400 can be programmed to accept this extra character by the use of TT-command (refer to chapter 15 for further information). The use of the extra character is a function of the input data type as described below. If transmission is stopped, time-out functions must be performed by controller.

12.2 Command Input, Text Input

If input logically is to be interpreted as character-codes, input may be command input or text input for P400 screen. In these cases a high rate of freedom in input format is allowed.

All usual termination characters (carnage return), (line feed) are always accepted. The termination character specified to P400 by TT-command is added to the list of default termination characters. No specified input data length is required.

12.3 Curve Input, Status Input

If input logically is to be interpreted as binary data, input may be curve input or status input. In these cases input termination format is restricted as specified by TT-command. Transmission must be terminated exclusively by the termination character if specified, or termination character must be avoided if this is specified. Furthermore, the transmission length must match exactly to the length, expected by P400.

13.0 PASSING DATA FROM P400

13.1 Data Output Form

All data outputs from P400 are in specified format as described below.

Data Output Format

2 bytes	Identification block
2 bytes	Block length
(block length)	Data block
Termination character if specified	

If transmission is stopped, P400 performes the time-out function after 10 ms. This value can be changed by the use of TL-command (refer to chapter 15 for further information).

13.1.1 Identification Bytes

The first of the two identification bytes describes the type of information to follow. If the information is curve data, second byte is used for additional information.

Byte 1: Information Type

|	bit 7:	Status output
|	bit 6:	Byte-out data (not yet used)
MSB	bit 5:	Error message
|	bit 4:	Cursor data
|	bit 3:	Test result data
LSB	bit 2:	Frequency counter data
|	bit 1:	Voltmeter data
|	bit 0:	Curve data

Only 1 of these bits set in one transmission

Byte 2: Only Defined if Curve Data

|	bit 7:	1 = A channel; 0 = B channel
|	bit 6:	1 = Split transfer; 0 = Index count
MSB	bit 5:	Reference curve
|	bit 4:	Measured curve
LSB	bit 3:)	(
|	bit 2:)	(Section No. split transfer
|	bit 1:)	(or
|	bit 0:)	(Index count = (0..15) ~ (1..16)

If output is error message, the data block has the form described below:

NN;XXX.... X;YY...Y

NN : Error type in ascii-code

XXX... X = Error message text

YYY... Y = Command which caused the error, or, if error is not caused by incorrect programme test, the last will execute command

The following error messages are obtainable:

<u>Error 0: Programming Errors</u>
- Out of range
- Illegal frequency
- Undefined command
- Internal error
- No keybox
- Move cursor to A or B reference

<u>Error 1: Curve Lock Error</u>
- Curve locked

<u>Error 2: Buffer Overflow</u>
- Input overflow
- Output overflow

<u>Error 3: User Requested Breakpoints</u>
- Breakpoint
- 0 Start "L" or Start "R")
- 1 Programme step "L")
- 2 Programme step "R") Keybox interrupts
- 3 Programme select "L")
- 4 Programme select "R")

<u>Error 4: Input Request</u>
- Input request

<u>Error 5: Input Transmission Error</u>
- Transmission mismatch

<u>Error 6: Output Transmission Error</u>
- Transmission mismatch
- Time-out error

<u>Error 7: Illegal Controller Request</u>
- Illegal IRQ

If output is test results, the data block has the format described below:

xxxxx;T1T2T3....Tn
xxxxx: Numerical value stating the tests which have been carried out

(T1)	1	sensitivity ch A
(T2)	2	sensitivity ch B
(T3)	4	channel balance
(T4)	8	frequency response ch A
(T5)	16	frequency response ch B
(T6)	32	rub & buzz ch A
(T7)	64	rub & buzz ch B
(T8)	128	polarity ch A
(T9)	256	polarity ch B
(T10)	512	user def. ch A (freq.resp.)
(T11)	1024	user def. ch B (freq.resp.)
(T12)	2048	user def. ch A (rub & buzz)
(T13)	4096	user def. ch B (rub & buzz)

If several tests have been carried out, xxxxx is the sum of the numerical values stated. xxxxx is intended as a control possibility during programming of the controller.

Tn states the tests which are "approved".
Example: 169;T1T4T8
169 = 1+8+32+128 i.e. T1, T4, T6, and T8 are carried out. Only T1, T4, and T8 are approved. T6 is rejected.

13.1.2 EOI-Wire, Termination Character

If necessary, P400 can be programmed to transmit an additional character together with the EOI-wire signal. As some controllers require a different termination character depending on type of information, one termination character can be specified if output is text, and another, if output is binary data.

13.3 Curve Data in P400

	2000 points	250 points
A curve	x	
A curve made to A reference	x	
B curve	x	
B curve made to B reference	x	
A reference low band		x
B reference low band		x
A reference high band		x
B reference high band		x
A reference loaded from cassette		x
B reference loaded from cassette		x

How Curves Are Stored in P400

A curve, A or B channel, is made of 2000 points of 2 bytes each.

```
                12 bit resolution
                      |
         MSB    |___________|  LSB
1 point:  0000|xxxx|xxxx|xxxx|
          |___|__________|
          zero   8 bit normal display of curve
```

One bit step on displayed curve = $\frac{60\ dB}{240\ points}$ = 0.25 dB

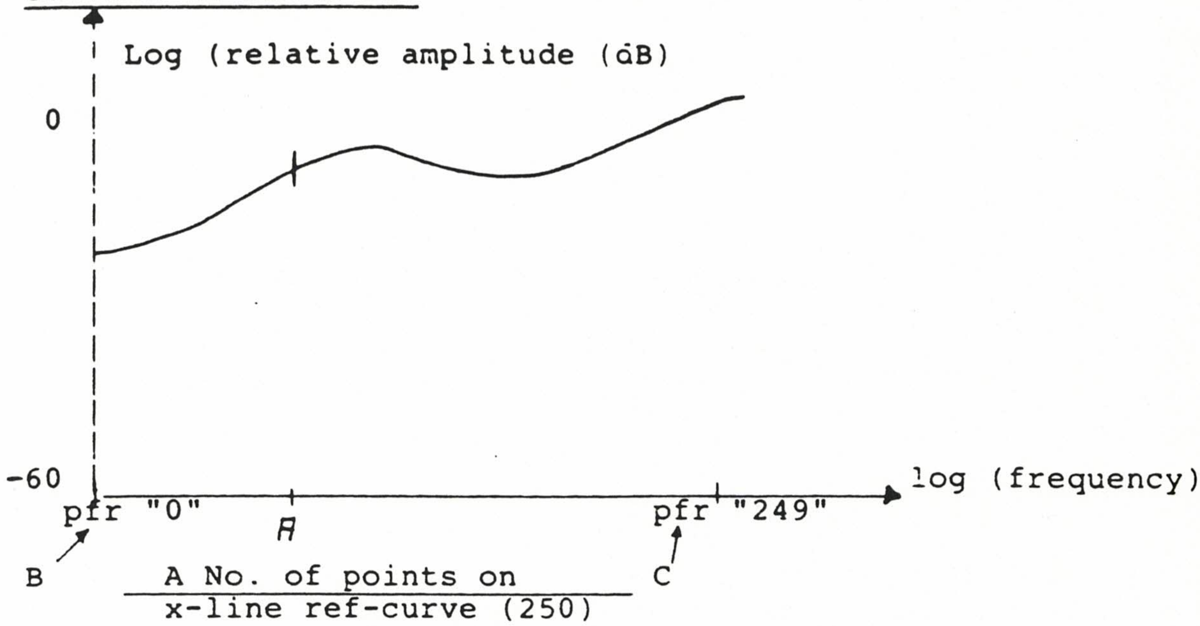
Screen vertical physical = 240 points
Screen horizontal physical = 250 points

REFERENCE CURVE STORED ON CASSETTE:
A reference curve, A reference or B reference, is made of 250 points of 2 bytes. The curve is made by selecting every 8th point from a 2000 curve.

$\frac{2000}{8}$ = 250 points reference curve

```
               12 bit resolution
              |_________________|
1 point: 0000|xxxx|xxxx|0000|
         |__|           |__|
         zero           zero
```

Convert to Point No.



If frequency is known, then formula is:

$$x = \frac{\log(f)-\log(B)}{\log\left(\frac{C}{B}\right)} \cdot (A-1)$$

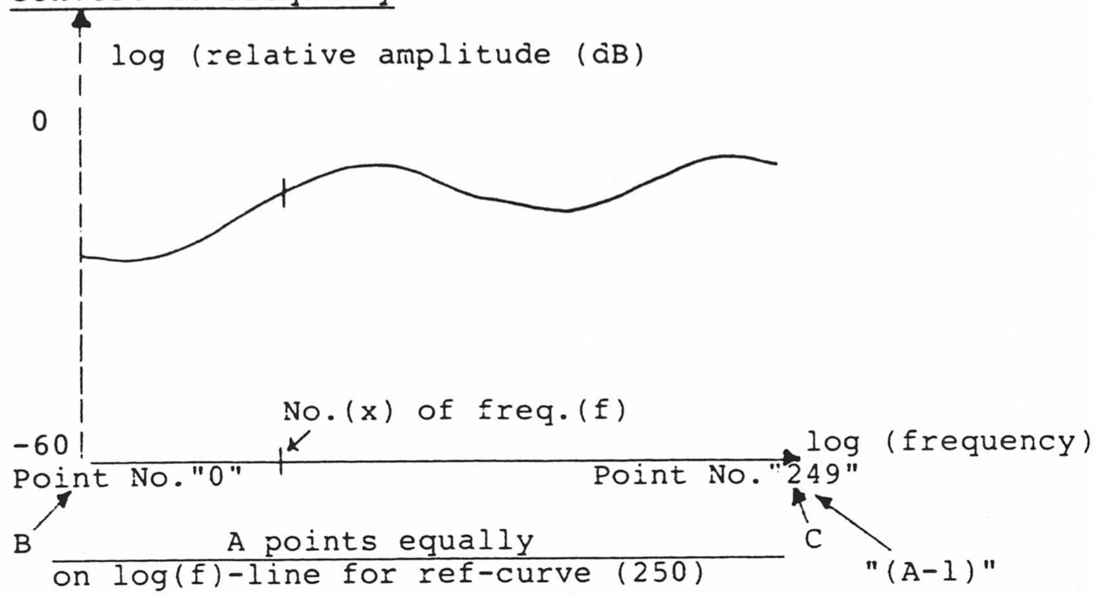
A = No. of points = 250
B = Start frequency = 100 Hz
C = End frequency = 10,000 Hz
f = Frequency known = e.g. 10 kHz

$$x = \frac{\log(10000) - \log(100)}{\log\left(\frac{10000}{100}\right)} \cdot (250-1)$$

$$x = \frac{4 - 2}{2} \cdot (249)$$

$$x = 249$$

Convert to Frequency



If point No. x is 249, then formula is:

$$f = B \cdot 10^{\left(\frac{x}{(A-1)} \cdot \log \left(\frac{C}{B} \right) \right)}$$

On a ref-curve the variables are:

B = start frequency = 100 Hz
C = end frequency = 10 kHz
A = No. of points = 250
x = point No. on axis, e.g. 249

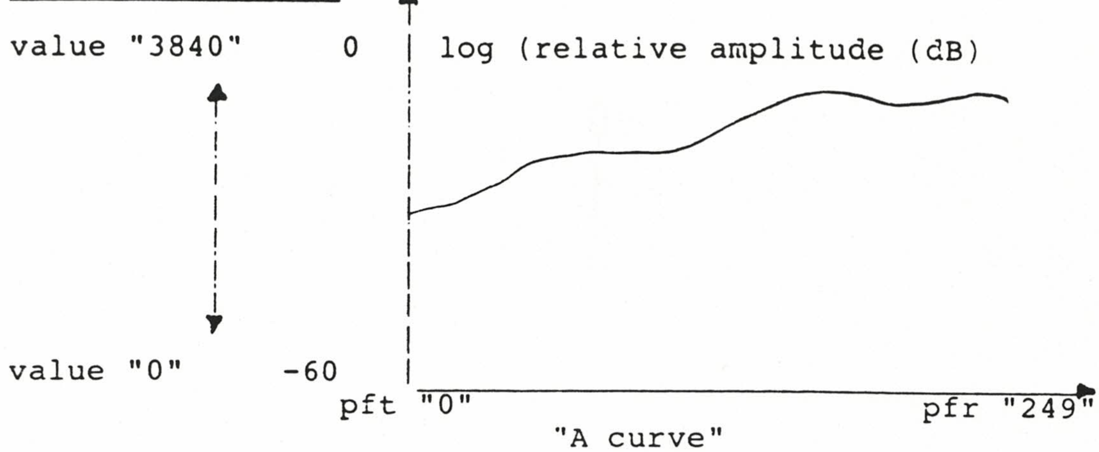
$$f = 100 \cdot 10^{\left(\frac{249}{(250-1)} \cdot \log \left(\frac{10000}{100} \right) \right)}$$

$$f = 100 \cdot 10^{(2)}$$

$$f = 100 \cdot 100$$

$$f = 10 \text{ kHz}$$

Convert to Log dB



To convert value into dB:

$$-60\ \text{dB} = 0$$
$$0\ \text{dB} = 3840$$

$$\Rightarrow \frac{60\ \text{dB}}{3840} = 0.015625\ \text{dB per step}$$

Formula:

$$x = B + \left(\frac{60}{3840} \cdot \text{value}\right)$$

$B = -60$, value $= 3840$

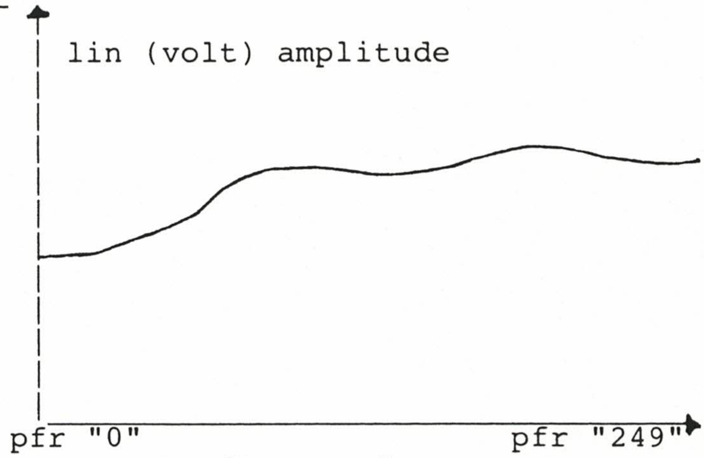
$$x\ \text{dB} = -60\ \text{dB} + \left(\frac{60\ \text{dB}}{3840} \cdot 3840\right)$$

$x = 0$ dB

Convert to Lin Volt

Value
10.00 V = "4000"
1.00 V = "4000"
100.00mV = "4000"
10.00mV = "4000"
30.00 V = "3840"
3.00 V = "3840"
300mV = "3840"
30mV = "3840"

Value "0"   0V



"A curve"

To convert value into Volt
0V = 0
10V = 4000
3V = 3840

=) $\frac{10V}{4000}$ = 0.002500 Volt per step

Formula:

$$x = \left(\frac{10V}{4000} \cdot \text{value} \right)$$

V = 2047

$$x \text{ Volt} = \frac{10}{4000} \cdot 2047$$

x = 5118 Volt

14.0 BUS COMMANDS

This chapter is a brief description of selected bus commands.

The description is logical, so how the standard physically implements these commands is not covered. For a full description of these items, the standard published by The Institute of Electrical and Electronic Engineers in New York is recommended. Furthermore, the description covers only a subset of the possible ones. Refer to controllers manual for further information.

14.1 BUS COMMAND EXAMPLES

14.1.1 Parallel Poll

Parallel poll is an unaddressed command. When performed, all peripherals on bus assert parallel poll register on data lines of bus.

14.1.2 Go To Remote (Remote Enable)

Go To Remote is an addressed command to logically connect P400 to controller.

14.1.3 Go To Local

Go To Local is a general command to logically disconnect P400 from controller. This function is identical to pressing the "local" push button on P400 frontpanel.

14.1.4 Local Lock-Out

Local Lock-Out is an addressed command. Performing this command will inhibit the use of "local" push button on P400 front panel.

14.1.5 Device Clear

Device Clear is a general command to bring the peripherals to a predefined state. P400 starts power-on procedure and initiates self-test.

14.1.6 Selected Device Clear (SDC)

Selected Clear is an addressed command to bring P400 to a predefined state. Two results can be obtained:

1: If P400 is in error state, the error condition is cleared.

2: If P400 is not in error state, P400 starts power-up procedure but skips self-test.

14.1.7 Trigger
Trigger is an unaddressed general command. On P400 a trigger initiates a measuring sequence.

If in voltmeter mode, the voltmeter readings are put into output buffer.

If in frequency counter mode, the frequency counter reading is put into output buffer.

If in time based or frequency based mode, P400 initiates new sweeps.

Refer also to 14.1.8 Selected Trigger and to P400 command IT (chapter 15). As this command is unaddressed, it can be used to synchronize several peripherals.

14.1.8 Selected Trigger
Selected Trigger is an addressed command. Except for the synchronization facilities, the function is identical to the general trigger.

14.1.9 Serial Poll
Serial Poll is an addressed command. When performed, P400 puts content of serial poll register on data lines of bus. P400 clears SRQ-bit in spoll register and enters parallel poll register.

14.2 IMPORTANT USER CONSIDERATIONS

Every bus command will interrupt normal P400 programme flow. This fact gives both advantages and trade-offs to the user.

The advantage is quick response to a bus command. The trade-offs are described below:

1) Every time normal programme flow is interrupted, the interrupt control logic in P400 must be reset. This means that a new bus command inserted to P400 while the old is executed may give unpredictable results. The time to execute a bus command is 200-600 μs.

2) The time used to execute bus commands is taken from normal programme time.

   Consider following programme sequence:

```
100 OUTPUT P400,"(commands)"
110 SPOLL P400
120 If (condition about spoll-reg) then 110
```

If spoll execution time is approximate 250 μS and controller can loop lines 110-120 in 1000 μS, P400 programme execution time will increase. If fast response from controller is required, the user should consider the P400 commands "BR'xx...x'" to generate SRQ to the controller.

3) When P400 is warm-started by Selected Device Clear, all P400 hardware including IEEE bus hardware is reset. The user must allow about 500 mS for these procedures.

14.3 TRIGGER CONSIDERATIONS

When performing measurement on transducers where mechanical or electrical transitions are involved, proper action must be taken to initial oscillations.

When time or frequency sweeps are performed, necessary delays can be programmed using P400 trigger functions.

If initial oscillations cause problems in voltmeter or frequency counter measurings, precautions using more than one measuring must be taken. When synchronizing P400 to other peripherals is not needed, the use of IT, internal trig P400 command is recommended.

14.4 MULTI-CONTROLLER CONSIDERATIONS

When using P400 in a multi-instrument or multi-controller environment, or when using P400 together with background transfer capabilities, special considerations must be taken if the active controller wishes to interfere a data transfer.

The reason for these special considerations is to be found in the fact that the EOI signal line information is lost in the Motorola 68488 interface circuit if the active controller issues an Attention request to P400.

14.4.1 Transfers FROM P400

If controller interferes in an output transfer before last byte is transferred, P400 ends transmission and enters error state containing error No. 7, Illegal Controller Request.

If controller interferes in an output transfer and the EOI-line is set, P400 considers transmission ended and the controller must check whether all listeners have accepted the message.

In general it is recommended to increase the P400 execution speed and always to terminate transfers from P400 by applying an Attention request. Most controllers do this automatically - but if they do not, the untalk command is recommended.

14.4.2 Transfers TO P400

If the controller interferes when information string is sent to P400 but before the EOI-line is set, the P400 resumes transmission after the controller has released the bus.

If the controller interferes when EOI-line is set, the EOI information is lost in the P400 which cannot detect whether the transmission is terminated correctly.

To avoid this the controller must monitor EOI-line prior to interference in transfer to P400 to see whether it is set or not.

If it is set, the controller can choose to wait till transmission is terminated, or if the controller must interfere, it must restore EOI-signal while releasing the bus and see that transmission is terminated correctly.

15.0 THE P400 COMMANDS

All P400 commands are initiated by 2 characters (only upper case characters are usable for commands), mostly followed by additional characters. The following is a description of P400 commands:

15.1 AAn-Command
Sets gain mode in P400 input section A channel.

n = 1 = manual gain
n = 2 = automatic gain

Example:
AA1 sets channel A to manual gain

15.2 ABn-Command
Sets gain mode in P400 input section B channel.
(refer to 15.1 AAn-command)

15.3 BCx-Command
Clears buffers and related information.

x = I: Clears input buffer and pointers
x = O: Clears output buffer, pointers and curve locks

Example:
BCI: Clears input buffer

15.4 BDn-Command (NOT L-VERSION)
Sets balance deviation on reference band pointed o

0 240
= 0 .... 30.0 dB step 0.125 dB

Example:
BD8: Sets balance deviation to 1 dB

15.5 BPn-Command

Sets backpanel user defined bits.

n = 0 7
  = binary code for 3 backpanel bits

Example:
BP5: Backpanel bit 1 and bit 3 are set TRUE
i.e. pin 22 and pin 23 LOW against pin 7 (13)

WARNING: Backpanel setting is configured to "active low"

Level	State
L	True
H	False

H = High level (5 V TTL)
L = Low level

RS 232 CONNECTORS/PIN NOS.

	Pin22(1)	Pin10(2)	Pin23(4)
BP 0	HIGH	HIGH	HIGH
BP 1	LOW	HIGH	HIGH
BP 2	HIGH	LOW	HIGH
BP 3	LOW	LOW	HIGH
BP 4	HIGH	HIGH	LOW
BP 5	LOW	HIGH	LOW
BP 6	HIGH	LOW	LOW
BP 7	LOW	LOW	LOW

15.6 BRx-Command

Introduces a service request to controller and logically disconnects keybox.

x = up to 16 characters for user documentation

Example:
BR25: Produces a service request to controller and puts following error message to output buffer:
03;BREAKPOINT;BR25

15.7 CAn-Command

Opens or closes A channel for measuring.

n = 1	DC measuring	and open	if closed	
n = 2	RMS measuring	-	-	
n = 3	ABS AVE,	-	-	(NOT L-VERSION)
n = 4	AC. AVE,	-	-	(NOT L-VERSION)
n = 5	POS. PEAK	-	-	(NOT L-VERSION)
n = 6	NEG. PEAK	-	-	(NOT L-VERSION)
n = 7	CLOSE IF OPEN			

In time based mode or in frequency based mode this command opens for measuring and must be performed before any trig.

In voltmeter mode this command selects measuring mode. Channel is turned off when a command is asserted which matches P400 setting.

Example:

CA2	In time based or frequency based mode: Opens channel A to RMS measuring.
CA2	In voltmeter mode and if channel A is not opened in RMS: Opens channel A to RMS measuring.
CA7	Channel A OFF
CB7	Channel B OFF

15.8 <u>CBn-Command</u>
Opens or closes channel B for measuring.
(refer to 15.7 CAn-command)

15.9 <u>CCn-Command</u>
Selects colour for succeeding text transfers to P400 screen.

n = 0,1,....,255

```
Colour Code                         Colour Code
    |                                    |
   (0)                                (0.16)
   (1)                                (1x16)
   (2)                                (2x16)
   (3)      + 8 if flash +            (3x16)  + 128 if flash
   (4)                                (4x16)
   (5)                                (5x16)
   (6)                                (6x16)
   (7)                                (7x16)
   |______________________|           |_______________________|
            text                             background
```

Colour Codes:

0 = black
1 = red
2 = green
3 = yellow
4 = blue
5 = magenta
6 = cyan
7 = white

Example:

CC31 (31 = 7 + 8 + 1x16): Prints succeeding test transfers to P400 screen in white flashing characters on a red non-flashing background.

WARNING: The error UNDEFINED COMMAND will be issued if the command "DID" has been sent to P400.

15.10 CDn-Command
Enables or disables dot connect function.

n = 1	A channel ON
n = 2	B channel ON
n = 3	A channel OFF
n = 4	B channel OFF

Example:
CD2: Connects dots on curves in channel B

WARNING: The error UNDEFINED COMMAND will be issued if the command "DID" has been sent to P400.

15.11 COn-m Command
Turns ON or OFF display of selected curves.

n = 1	A curve
n = 2	A reference curve
n = 3	B curve
n = 4	B reference curve

m = 1	ON
m = 2	OFF display

Example:
CO1-1: Turns ON display of A curve

WARNING: The error UNDEFINED COMMAND will be issued if the command "DID" has been sent to P400.

15.12 CPx-Command
Enables or disables compressor compensation of measured curves in automatic test mode.

x = E	Enables compressor compensation
x = D	Disables compressor compensation

Example:
CPE Compressor compensation is enabled

15.13 CSn-Command
Performs certain arithmetic functions (additions and subtractions) between the reference curve and the one measured.

n = 1	A curve = A curve + A ref
= 2	A curve = A curve - A ref
= 3	B curve = B curve + B ref
= 4	B curve = B curve - B ref
= 5	A curve = A curve + (20 DB rel - B curve) (NOT L-VERSION)

Example:

CS1 Adds A curve and A reference to new A curve

CS5 Compressor seeks to maintain an output level 20 dB below full scale. Any difference between compressor curve (B curve) and the level 20 dB below full scale is added or subtracted to A curve.

15.14 CTx-Command

Performs curve test. If reference curve is a band, a response test is performed, i.e. measured curve must lie within the two bands. Automatic move function is performed according to the move option setting (refer to 15.38 MO-command). If reference curve is a single curve, a rub & ruzz test is performed, i.e. measured curve must be below reference curve.

x = A Curve test in channel A
x = B Curve test in channel B

Example:

CTA A curve test is performed in channel A

The test result is transferred to output buffer (refer to chapter 13 for further information).

One command at a time is stored in output buffer.

Example:

"CTA"	=	512;T10	) Frequency
"CTB"	=	1024;T11	) response
"CTA"	=	2048;T12	) Rub & buzz
"CTB"	=	4096;T13	)

15.15 CXn-m Command

Sets cursor on curve and transfers cursor values to output buffer.

n = 0,1,....,249
m = 1,2,3,4,5,6

n = The x-position on curve
0 = the leftmost curve point
249 = the rightmost curve point

m = The curve
m = 1 A curve
m = 2 A reference (or upper limit)
m = 3 B curve
m = 4 B reference (or upper limit)
m = 5 A reference lower limit
m = 6 B reference lower limit

Example:

CX120-1 Cursor is set on A curve and cursor values are transferred to output buffer.

WARNING: Certain commands require a prior execution of CXn-m command.

WARNING: The error UNDEFINED COMMAND will be issued if the command "DID" has been sent to P400.

WARNING: See chapter 15.20 EAn-Command

15.16 DBx-Command
Enables or disables the debugger picture

x = D   Disables picture
x = E   Enables picture

Example:
DBE: Enables picture

WARNING: The error UNDEFINED COMMAND will be issued if the command "DID" has been sent to P400.

WARNING: If Debugger picture is on, the command "DID" will give error.

15.17 DCn-m Command
Selects decimation count connected to reference curve.

n = 1   Selects A reference
n = 2   Selects B reference

m = 1 to 250, selects decimation value

Example:
DC1-16   Sets decimation count in channel A to 16

15.18 DIx-Command
Enables or disables P400 screen while running remote mode.

x = D   disables P400 screen
x = E   enables P400 screen

Example: DID disables P400 picture

WARNING: If Debugger picture is ON, the command "DID" will give UNDEFINED COMMAND error.

15.19 DS-Command
Performs a print-out of curves currently displayed on the screen.

Example:
DS   Initiates print-out on Olivetti TH240 or Siemens PT88 printer connected

15.20 EAn-Command

Performs curve smoothing in channel A. When P400 is changed from local to remote mode, any curves on screen are turned off. The performing of an EA-command turns A curves on again

n = 1	No smoothing
n = 2	2 points smoothing
n = 3	4 - -
n = 4	8 - -
n = 5	16 - -
n = 6	32 - -
n = 7	64 - -
n = 8	128 - -

Example:

EA1 Removes any curve smoothing on A curve and A reference

WARNING: The error UNDEFINED COMMAND will be issued if the "DID" has been sent to P400.

WARNING: Do not execute after FM1 or FM4.

15.21 EBn-Command

Refer to 15.20 EAn-command.

15.22 FAn-Command

Sets filter functions in channel A.

If filter is standard tracking filter:

n = 0	Disables filter functions	
n = 1- 2	0 dB gain	) Figure indicates tracking
n = 3-12	20 dB gain	) controlled by P400 generator

If filter is optional broad band filter:

n = 0	Disables filter functions
n = 1- 2	Enables filter functions with 0dB gain in filter
n = 3-12	Enables filter funktions with +30dB gain in filter

Example:

FA4	Standard filter	Filter is tracking 4th harmonic from generator frequency
	Optional filter	Filter is enabled with +30dB gain

15.23 FBn-Command

Sets filter functions in channel B.
(refer to 15.22 FAn-command)

15.24 FCn-Command

Sets input source to frequency counter and trigger device.

n = 1	Frequency counter and trigger are sourced internally (i.e. from P400 generator)
n = 2	Frequency counter and trigger are sourced from channel A input
n = 3	Frequency counter and trigger are sourced from channel B input

Example:

FC1	Triggering and frequency measuring are performed internally

WARNING: Use of P400 as frequency counter may change these values.

WARNING: The "FM4" command automatically sets external frequency channel A.

15.25 FMn-Command

Sets the basic P400 function mode.

n = 1	Voltmeter mode
n = 2	Time based mode
n = 3	Frequency based mode
n = 4	Frequency counter mode

Example:

FM3	Initiates frequency based mode

15.26 GAn-Command

Sets gain in P400 input amplifier section channel A.

n = 1	10 mV	full scale
n = 2	30 mV	-
n = 3	100 mV	-
n = 4	300 mV	-
n = 5	1 V	-
n = 6	3 V	-
n = 7	10 V	-
n = 8	30 V	-
n = 9	100 V	-

Example:

GA4	Channel A is set to 300 mV full scale

15.27 GBn-Command

Sets gain in P400 input amplifier section channel B.
(refer to 15.26 GAn-command)

15.28 GCn-m Command

n = 1,2,3,4,5,6,7,8

n =	1	A curve
	2	A reference
	3	B curve
	4	B reference
	5	A reference (low band) (only m = 1x)
	6	B reference (low band) (only m = 1x)
	7	A reference (high band)(only m = 1x)
	8	B reference (high band)(only m = 1x)

m = 10,11,12

10	250 points of 2 bytes
11	1st 125 points of 250 points of 2 bytes
12	2nd 125 points of 250 points of 2 bytes

m = 200,201,202,....215,216

200	2000 points of 2 bytes
201	1st 125 points of 2000 points of 2 bytes
202	2nd 125 points of 2000 points of 2 bytes
203	3rd 125 points of 2000 points of 2 bytes
...	.... 125 points of 2000 points of 2 bytes
...	.... 125 points of 2000 points of 2 bytes
215	15th 125 points of 2000 points of 2 bytes
216	16th 125 points of 2000 points of 2 bytes

Get curve command sets the out queue buffer with the output information block (13.1.1). At the same time curve lock register is ON for the selected curve. That means that the selected curve cannot be recorded until the selected curve is read by the controller.

Example:

Step 1	Commands sent to P400 GC1-201;SC1-201
Step 2	P400 responds with: SRQ ON 01;curve lock error;SC1-201
Step 3	The controller reads the curve data
Step 4	The controller sends a SDC
Step 5	P400 responds with: SRQ ON 04;input request;SC1-201
Step 6	The controller stores the new curve data

Special considerations for commands GCx-10, GCx-11, GCx-12:

These commands are intended for use only together with reference curves. If they are used with measured curves, special functions are introduced which may cause unexpected results to the ordinary user.

15.29 HCn-Command

Sets default prints programming mode to auto tests.

n = 0 to 15

The n-value must be considered a 4-bit word, each word indicating the curve which is to be printed out.

```
n = | x | x | x | x | 8 | 4 | 2 | 1 |
    ------------------------------------
                      |   |   |   |__ A curve
                      |   |   |_______ A reference
                      |   |___________ B curve
                      |_______________ B reference
```

If A curve and B curve are to be programmed, the command to be performed is HC5 (5 = 1+4).

Example:

HC0: Sets auto select in auto mode

WARNING: The error UNDEFINED COMMAND will be issued if the "DID" has been sent to P400.

15.30 ICn Index Count

n = 1..16

Index count is used with GET curve command. It is used to step through a 2000 point or 250 point curve (chapter 13.3) with the possible step of 1 to 16.

Example:

IC4 Index count set to

GC1-200 Get curve channel A, the number of points is $\frac{2000}{4} = 500$

CURVE DATA

```
Point  0 1 2 3 4 5 6 7 8        1995 1966 1997 1998 1999
       ↑       ↑       ↑        ↑                   ↑
       0       1       2        498                 499
```

The Index count is stored in the Identification Bytes (chapter 13.0)

Byte 1: Information Type

```
  |     bit 7:   Status output
  |     bit 6:   Byte-out data (not yet used)
 MSB    bit 5:   Error message
  |     bit 4:   Cursor data
  |     bit 3:   Test result data
 LSB    bit 2:   Frequency counter data
  |     bit 1:   Voltmeter data
  |     bit 0:   Curve data
```

Only 1 of these bits set in one transmission

Byte 2: Only Defined if Curve Data

	bit 7:	1 = A channel; 0 = B channel
	bit 6:	1 = Split transfer; 0 = Index count
MSB	bit 5:	Reference curve
	bit 4:	Measured curve
LSB	bit 3:)	(
	bit 2:)	(Section No. split transfer
	bit 1:)	(or
	bit 0:)	(Index count = (0..15) (1..16)

GC2-10 or GC4-10	250 points 250 points
n = 1	250
n = 2	125
n = 3	83
n = 4	62
n = 5	50
n = 6	41
n = 7	35
n = 8	31
n = 9	27
n = 10	25
n = 11	22
n = 12	20
n = 13	19
n = 14	17
n = 15	16
n = 16	15

GC1-200 or GC3-200	2000 points 2000 points
n = 1	2000
n = 2	1000
n = 3	666
n = 4	500
n = 5	400
n = 6	333
n = 7	285
n = 8	250
n = 9	222
n = 10	200
n = 11	181
n = 12	166
n = 13	153
n = 14	142
n = 15	133
n = 16	125

15.31 ITx-Command

Initiates measuring in P400. Performs same function as a bus-trigger-command. Refer to chapter 14.1.7 and 14.1.8 for further information.

x = up to 16 characters for user documentation only

Example:
ITR6: Performs a trigger function

WARNING: This command is compatible with LV and LT commands

15.32 KEx-Command

Logically connects the keybox to P400.

x = 0 Logically disconnects the keybox
x = F Connects keybox for service requests

If keybox is connected for service request, any use of the keybox will cause a service request to controller. The key pressed can be determined by reading the error message in output buffer:

0: Lower key	("START" left or right)
1: Middle right	("PROGRAM STEP" right)
2: Middle left	("PROGRAM STEP" left)
3: Upper right	("PROGRAM SELECT" right)
4: Upper left	("PROGRAM SELECT" left)

WARNING: The keyboard interrupt is not disabled automatically after a key has been pressed, i.e. it is necessary to insert a KEO command as the first command in the keyboard interrupt handler routine.

15.33 LAn-Command

Selects display mode for amplitude measuring in channel A to be linear or logarithmic.

n = 1 Linear display is selected
n = 2 Logarithmic display is selected

Example:
LA2: Logarithmic display in channel A is selected

15.34 LBn-Command

Selects display mode amplitude measuring in channel B to be linear or logarithmic. (refer to 15.33 LAn-command)

15.35 LT-Command

Any test performed when P400 is in remote mode (i.e. controlled via the IEEE bus) will fill the test results in output buffer. If a test is performed from the keybox in manual mode, the test results can be monitored by controller after remoting P400. The LT-command transfers most recent test results to output buffer.

Output format, refer to chapter 15.32 KEx-command. This command will not normally be used.

15.36 LF-Command

Same as IT-Command, see chapter 15.31.

15.37 LV-Command
Same as IT-Command, see chapter 15.31.

15.38 MD'x'-Command
Deletes reference curves from connected EEPROM cassette.
x = up to six characters in correct EEPROM driver syntax

Example:
MD'L1xx': Ref.curve named L1xx is deleted from memory cas.

WARNING: When P400 is accessing memory cassette, no further use of P400 is allowed.

15.39 MLx'y'-Command
Loads reference curves from EEPROM cassette into P400 memory

x = A Load reference curve in channel A
x = B Load reference curve in channel B
y = Up to 6 characters in correct EEPROM driver syntax

Example:
MLA'L1xx':Reference curve named L1xx is loaded to P400 memory channel A.

WARNING: Refer to chapter 15.38 MD'x'-command

WARNING: If P400 is interrupted from the controller during cassette access, unpredictable results may occur.

15.40 MOx-Command
Enables or disables the move curve option when a frequency response test is performed.

x = E Move option is enabled
x = D Move option is disabled

Example:
MOE Enables move option

15.41 MSx'y'-Command

Stores reference curves from P400 memory on EEPROM cassette.

x = A	Store reference curve in channel A
x = B	Store reference curve in channel B
y =	Up to six characters in correct EEPROM driver syntax

Example:

MSA'H124'	Reference curve in channel A is stored on cassette with the name H124

WARNING: Refer to chapter 15.38 MD'x'-command

WARNING: If P400 is interrupted from the controller during cassette access, unpredictable results may occur.

15.42 OAXn-Command

This command controls the functions performed by the CVA2 optional output amplifier or by the VA1 output amplifier.

X = A   Mute functions in channel A
- n = 1: Mute off
- n = 2: Mute on

X = B   Mute functions in channel B
- n = 1: Mute off
- n = 2: Mute on

X = R   Amplifier range select
- n = 1: Low range
- n = 2: High range

X = S   Oscillator functions
- n = 1: Oscillator is off
- n = 2: Oscillator is on
- n = 3: Oscillator is automatic on/off

X = V   Output mode select
- n = 1: Output is voltage output
- n = 2: Output is current output

Examples:

OAS3:	Oscillator starts sweep automatic
OAR2:	Output is high range

15.43 OFXn-Command
Selects the frequencies supplied by the internal generator.

X = B   Selects sweep start frequency
X = E   Selects sweep final frequency
X = M   Selects manual (fixed) frequency

n is the frequency supplied in one of the following syntaxes
  n = xxxxx.xxx   the frequency in Hz
  n = xx.xxxK   the frequency in kHz
  n = xx.xxxk   the frequency in kHz

Examples:
OFB10000.000   Selects begin sweep to 10 kHz
OFB10.000K   Selects begin sweep to 10 kHz

WARNING: Leading zeroes must be omitted. Not all frequencies can be selected. P400 selects nearest available frequency.

15.44 OMn-Command
Selects the function mode of the internal P400 generator.

n = 1   A fixed frequency, continuous sinus is supplied
n = 2   A fixed frequency, sinus pulse is supplied
n = 3   A sweep is supplied

Example: OM3   The generator is initiated to sweep mode

15.45 OTn-Command
Selects the output amplitude from output amplifier, related to full output amplitude.

n = 1 to 841 in steps of 0.1 dB. In practice generator steps 0.375 dB

Example:
OT10   Select 1 dB below full output. Generator is set to -0.7 dB which is the nearest lowest value.

15.46 OWn-Command
Selects sweep time for generator sweeps. All sweep times are not available. If selected sweep time is within range of possibility, the nearest possible one is selected.

n = units of 100 mS

Example:
OW100   Sweep time is set to approx. 10 sec.

15.47 PPn-Command
This command selects the answer to controller if P400 has issued a service request and is parallel polled.

The command performs the same functions as the addressed bus commands Parallel Poll Configure and Parallel Poll Unconfigure.

n = 0 Similar to parallel poll unconfigure. No answer is issued when P400 is parallel polled.

n = 1 to 8 Similar to parallel poll configure. The answer issued is 2^{n-1} (i.e. n = 1 = answer = 1, n = 2 = answer = 2 etc.)

The parallel poll register is set when SRQ is requested and cleared when P400 is spolled.

Example:
PP7 The answer from P400 = 64
(64 dec = 40 HEX = 01000000 BIN)

15.48 PSx-n Command
Selects several printer functions to be performed in auto test mode or with direct print-out.

x = F Selects the quick mode
n = 0 Quick mode disabled
n = 1 Quick mode enabled

x = S Selects if status output is performed in auto test mode
n = 0 No status output is supplied
n = 1 Status output is supplied

x = T Selects the triggering event in auto test mode
n = 0 A print is never triggered
n = 1 A print is triggered if test is rejected
n = 2 A print is triggered if test is approved
n = 3 A print is always triggered

Example:
PST1: Selects "trig on reject"

WARNING: The error UNDEFINED COMMAND will be issued if the "DID" has been sent to P400.

15.49 PTx-Command
Performs a polarity test in one channel.

x = A Polarity test is performed in channel A
x = B Polarity test is performed in channel B

Example:
PTA: Performs a polarity test in channel A

Test result is transferred to output buffer (refer to chapter 13 for further information).

If output is test results, the data block has the format described below:

xxxxx;T1 T2 T3....Tn

PTA = 128;T8
PTB = 256;T9

5.50 RGn-m Command

Performs P400 internal memory curve register functions.
n = 1 ... 55 = curve register No. to be used

WARNING: A standard P400 contains memory for 7 curve registers. For every 64K optional user RAM, add 16 curve registers.

m = 1: Store A curve in register
m = 2: Store B curve in register
m = 3: Load A curve from register
m = 4: Load B curve from register
m = 5: Exchange A curve with curve in register
m = 6: Exchange B curve with curve in register

Example:
RG4-2: Stores B curve in register 4

15.51 RSx-n Register Status

x = S Save
L Load
n = Number between (1..4)

Load status from memory register or save status to memory register.

RSS-2 Save status in memory No. 2
RSL-2 Load status from memory No. 2

WARNING: Memory is dynamic memory and register is not saved on power off.

15.52 SCn-m Store Curves

n = 1,2,3,4,5,6,7,8

n =	1	A curve	(only m = 2xx)
	2	A reference	
	3	B curve	(only m = 2xx)
	4	B reference	
	5	A reference (low band)	(only m = 1x)
	6	B reference (low band)	(only m = 1x)
	7	A reference (high band)	(only m = 1x)
	8	B reference (high band)	(only m = 1x)

m =	10,11,12	
	10	250 points of 2 bytes
	11	1st 125 points of 250 points of 2 bytes
	12	2nd 125 points of 250 points of 2 bytes

```
m = 200,201,202,....215,216
    200      2000 points of 2 bytes
    201       1st 125 points of 2000 points of 2 bytes
    202       2nd 125 points of 2000 points of 2 bytes
    203       3rd 125 points of 2000 points of 2 bytes
    ...      .... 125 points of 2000 points of 2 bytes
    ...      .... 125 points of 2000 points of 2 bytes
    215      15th 125 points of 2000 points of 2 bytes
    216      16th 125 points of 2000 points of 2 bytes
```

Store curves command sets the input request flag and SRQ the controller. The data now is treated as curve data and not command data. If the controller sends too many bytes or less bytes than requested, P400 sets a SRQ flag and the output buffer in P400 is stored with error No. 5 text "TRANSMISSION MISMATCH".

Example of store curve:

Step 1	Command sent to P400 SC3-202 (B reference width is 2000 points, only 125 points are stored on the 2nd place)
Step 2	P400 responds with: SRQ ON 04;input requested
Step 3	The controller sends curve data
Step 4	P400 responds with: Displaying the curve

15.53 SD'x'-Command

Deletes a P400 status set-up from connected EEPROM cassette

x = Up to 6 characters in correct EEPROM driver syntax

Example:
SD'H4ABx': Deletes status set-up named H4ABx from cassette

15.54 SEx-Command

Performs sensitivity test in P400 according to information in reference curve status and if necessary also channel balance test.

x = A	Performs sensitivity test in channel A
x = B	Performs sensitivity test in channel B
x = C	Performs sensitivity test in both channels and channel balance if defined

Example:

SEA: Sensitivity test is performed in channel A and test result is transferred to output buffer.(For test result format, refer to chapter 13, Passing data from P400.

If output is test results, the data block has the format described below:

xxxxx;T1 T2 T3....Tn

SEA = 1;T1
SEB = 2;T2
SEC = 7;T1T2T3

15.55 SF-Command

Sets sensitivity frequency on reference band pointed by cursor at frequency pointed by cursor. No parameters required.

Example: SF: Sets the sensitivity frequency

15.56 SL'x'-Command

Loads a complete set-up from memory cassette
x = up to 6 characters in correct EEPROM driver syntax

Example:
SL'M4AB4': A status set-up named M4AB4 is loaded from cas.

WARNING: If P400 is interrupted from the controller during cassette access with the commands "Selected Clear", "Device Clear", unpredictable results may occur.

15.57 SS'x'-Command

Stores complete P400 programme setting an EEPROM cassette under supplied name

x = up to 6 characters in correct EEPROM driver syntax

Example:
SS'SMITH': Transfers a complete set-up to cassette as a file named SMITH.

WARNING: If P400 is interrupted from the controller during cassette access with the commands "Selected Clear", "Device Clear", unpredictable results may occur.

15.58 STx-Command

Handles status transfers between P400 and controller. Every status transfer is 64 bytes long and contains a full set-up for P400.

x = I: Produces an input request to controller
x = O: Puts the full set-up in output buffer

Example:
STO: Collects a full programme set-up in output buffer

15.59 TAn-Command

Selects time constant in channel A if generator is not in auto mode.

n = 1	Time constant is set to 1 Hz
n = 2	Time constant is set to 20 Hz
n = 3	Time constant is set to 150 Hz

Example:

TA2 Time constant is set to 20 Hz

WARNING: If generator is in AUTO, an error message will be issued.

15.60 TBn-Command

Selects time constant in channel B. (Refer to chapter 15.59 TAn-command for further information).

15.61 TDn-Command

Selects trigger delay (delay before P400 starts measuring).

n = 1 to 60000 in units of 10 mS

Example:

TD50: A trigger delay of 500 mS is selected

15.62 TFn-Command

Selects trigger frequency if trigger mode is set to frequency trig.

n = 1	1 Hz	trigger	frequency
2	2 Hz	-	-
3	5 Hz	-	-
4	10 Hz	-	-
5	20 Hz	-	-
6	50 Hz	-	-
7	100 Hz	-	-
8	200 Hz	-	-
9	500 Hz	-	-
10	1 kHz	-	-
11	2 kHz	-	-
12	5 kHz	-	-
13	10 kHz	-	-
14	20 kHz	-	-
15	50 kHz	-	-
16	100 kHz	-	-

Example:

TF10: Trigger frequency is selected to 1 kHz

15.63 TLn-Command

Sets time limit for output transfers.

n = 0 Sets time limit to default value 10 ms
n = 1..250 Sets time limit to n ms

Example:
TL80: Time limit is set to 80 ms

15.64 TMn-Command

Selects the mode to trig P400 to start measuring.

n = 1 Frequency triggering is selected
n = 2 Manual triggering is selected

Example:
TM1: Frequency triggering is selected

Frequency trigger is only usable for external frequency.

15.65 TTn(-m) Command

Manipulates all transfer termination codes for input and output transfer.

Form 1:
TT0: Clears all termination codes. Transmission must end with EOI signal on last byte of data.

Form 2: TTn-m
n = 1: Sets input termination character for input test and binary transfer

n = 2: Sets termination character for output text transfers

n = 3: Sets termination character for output binary transfers

m: The termination character

Example:
TT2-10: Output texts are terminated by EOI and line-feed ($A)

WARNING: The termination character must not be 00 HEX

15.66 WTn-m Command

Selects print position on P400 screen and generates an input request.

n = 4 to 20 Indicates line No.
m = 16 to 68 Indicates char.No.

P400 screen cannot be written further than character position 69.

Example:
WT16-16: Initiates writing on screen line 16, char. 16.

WARNING: The error UNDEFINED COMMAND will be given if "DID" command is sent to P400.

15.67 <u>XBn-Command</u>
Selects begin frequency of P400 display.

n = 1	1 Hz	is	selected
2	2 Hz	-	-
3	5 Hz	-	-
4	10 Hz	-	-
5	20 Hz	-	-
6	50 Hz	-	-
7	100 Hz	-	-
8	200 Hz	-	-
9	500 Hz	-	-
10	1 kHz	-	-

Example:
XB5: Display starts at 20 Hz

15.68 <u>XEn-Command</u>
Selects stop frequency of P400 display

n = 1	100 Hz	is	selected
2	200 Hz	-	-
3	500 Hz	-	-
4	1 kHz	-	-
5	2 kHz	-	-
6	5 kHz	-	-
7	10 kHz	-	-
8	20 kHz	-	-
9	50 kHz	-	-
10	100 kHz	-	-

Example:
XE8: Display stops at 20 kHz

15.69 <u>XTn-Command</u>
Selects time-base time on P400 display

n = 1 to 60000 in units of 100 mS

Example:
XT100: A time of 10 sec. is selected

16.0 THE SELF-TEST PROCEDURE

16.1 Spolling During Self-Test

As P400 automatically enters self-test procedure when power-up is performed, it is important that the active controller can query whether the self-test is ended or not.

To fulfil this need, simple serial polling is allowed during self-test. When self-test is terminated and if P400 is in service request mode according to back panel switch setting, the P400 issues a service request to the controller. The following serial poll answers are obtainable:

- 0: Self-test is proceeding
- 1: Self-test is passed but service request mode is not set
- 2: Self-test is in error and service request mode is not set
- 65: Self-test is passed and service request is issued
- 66: Self-test is in error and service request is issued

16.2 Device Clear After Self-Test

When self-test is terminated, you can enter normal mode by pressing START push button on P400 front panel.

Furthermore, P400 is entering normal mode if the controller issues a device clear general command or a device clear command. Only when P400 enters normal mode, the IEEE software in P400 is initialized and normal operations can occur.

Attention is drawn to the following facts:

1. If you enter normal mode by pressing START push button, the service request issued by self-test programme may still be pending in the bus system and it may be necessary to clear this service request by performing a serial poll command to the bus system.

2. It is not possible to change P400 mode from AUTO-mode using keybox and EEPROM cassette to remote mode controlled by external controller. If the user wishes to enter normal mode when keybox and cassette is connected, he should avoid file names on cassette reserved for use in AUTO-programmes.

CASSETTE STATUS FORMAT

Byte No.:	Names:
00-01	MODE: MODE 2 + MODE 1
02-03	MODE: MODE 2 + MODE 3
04-05	TRGGER: TRGKAN + TRGMOD
06-07	TRGFRQ:
08-09	DELAY
10-11	XSTART
12-13	XSLUT
14	$0
15	OPSKAN
16	$1
17	OPSKAN + 1
18-19	LINLGY: LINLGA + LINLGB
20-21	AUTMAN: AUTMNA + AUTMNB
22-23	XTID
24-25	GAINA
26-27	GAINB
28-29	RMSTID: RMSA + RMSB
30-31	SKRHSA
32-33	SKRHSB
34-35	DSPKAN: DSPKA + DSPKB
36-49	GENST
50-53	FILTR 1 + FILTR 2
54	SDELAY
55-56	CURSEL
57	COMPRS
58	BACKPN
59-63	5 x $FF

STATUS BYTES

Byte No. 00-01/02-03
DCL 1 MODE,
  2 MODE2,
  2 MODE1
DCL   MODE3

MODE2:	0 :	NORMAL MODE (LOCAL)
	1 :	IEC MODE (REMOTE)
MODE1:	1 :	VOLTMETER MODE
	2 :	TIME BASED MODE
	3 :	FREQUENCY BASED MODE
	4 :	FREQUENCY COUNT MODE
MODE3:	0 :	NORMAL (WITHOUT KEYBOX)
	11 :	AUTOMATIC LOUDSPEAKERS
	12 :	AUTOMATIC HEADPHONES
	(13 :	AUTOMATIC MICROPHONES)not used

Byte No. 04-05:
DCL 1 TRGGER,
  2 TRGKAN,
  2 TRGMOD

TRGKAN:	0 :	INTERNAL FREQUENCY MEASUREMENT
	1 :	EXTERNAL FREQUENCY MEASUREMENT CH A
	2 :	EXTERNAL FREQUENCY MEASUREMENT CH B
TRGMOD:	25 :	MANUAL TRIGGERING
	24 :	FREQUENCY TRIGGERING

Byte No. 06-07:
DCL TRGFRQ BIN(2)   TRIGGER FREQUENCY IN FREQ.COUNTER FORMAT

TRGFRQ: $9D1E :   100 Hz (e.g.)

Byte No. 08-09:
DCL DELAY BIN(2)   DELAY AFTER TRIG
DCL SDELAY

DELAY	5 :	DELAY 50 ms (e.g.) VALUES FROM 0-60000 IN STEPS OF 10 ms
SDELAY	0 :	DELAY OFF
	1 :	DELAY ON

Byte No. 10-11:
DCL XSTART BIN(2)   START FREQUENCY IN FREQ. COUNTER FORMAT

XSTART $9D1E :   100 Hz START FREQUENCY (e.g.)

```
Byte No. 12-13:
DCL XSLUT BIN(2)      END FREQUENCY IN FREQ. COUNTER FORMAT

XSLUT      $9D1E :    100 Hz END FREQUENCY (e.g.)
```

```
Byte No. 15+17:
DCL 1 OPSKAN,         DETECTOR AND ARM INFORMATION
    2 OPTAGA,
    2 OPTAGB

OPTAGA         1 :    DC          CHANNEL A
               2 :    RMS         -
               3 :    ABS AVE     -
               4 :    AC  AVE     -
               5 :    POS PEAK    -
               6 :    NEG PEAK    -
           > $10 :    CHANNEL ARMED

OPTAGB         1 :    DC          CHANNEL B
               2 :    RMS         -
               3 :    ABS AVE     -
               4 :    AC  AVE     -
               5 :    POS PEAK    -
               6 :    NEG PEAK    -
         >   $10 :    CHANNEL ARMED
```

```
Byte No. 18-19:
DCL 1 LINLGY,         LINEAR OR LOGARITHMIC SCALE
    2 LINLGA,         CHANNEL A OR CHANNEL B
    2 LINLGB

LINLGA         0 :    LINEAR      CHANNEL A
               1 :    LOGARITHMIC          -

LINLGB         0 :    LINEAR      CHANNEL B
               1 :    LOGARITHMIC          -
```

```
Byte No. 20-21:
DCL 1 AUTMAN,         AUTOMATIC OR MANUAL GAIN REGULATION IN
    2 AUTMNA,         VOLTMETER MODE
    2 AUTMNB

AUTMNA         0 :    MANUAL GAIN REGULATION   CHANNEL A
               1 :    AUTO   GAIN REGULATION   -

AUTMNB         0 :    MANUAL GAIN REGULATION   CHANNEL B
               1 :    AUTO   GAIN REGULATION   -
```

```
Byte No. 22-23:
DCL XTID BIN(2)       TOTAL MEASURING TIME IN 100 ms STEPS

XTID          30 :    3 sec. MEASURING TIME (e.g.)
                      VALUES FROM 1-60000 = 100 ms -100 MIN.
```

Byte No. 24-25:
DCL GAINA BIN(2)   GAIN CHANNEL A

GAINA	0 :	10 mV
	1 :	30 mV
	2 :	100 mV
	3 :	300 mV
	4 :	1 V
	5 :	3 V
	6 :	10 V
	7 :	30 V
	8 :	100 V

Byte No. 26-27:
DCL GAINB BIN(2)   GAIN CHANNEL B

GAINB	0 :	10 mV	-100 dB
	1 :	30 mV	- 90 dB
	2 :	100 mV	- 80 dB
	3 :	300 mV	- 70 dB
	4 :	1 V	- 60 dB
	5 :	3 V	- 50 dB
	6 :	10 V	- 30 dB
	7 :	30 V	- 20 dB
	8 :	100 V	- 10 dB

Byte No. 28-29:
DCL 1 RMSTID,   TIME CONSTANTS RMS DETECTOR
2 RMSA,
2 RMSB

RMSA	0 :	1 Hz TIME CONSTANT
	1 :	20 Hz - -
	2 :	150 Hz - -
	3 :	VARIABLE - CONTROLLED BY INT. GENERATOR
RMSB	0 :	1 Hz TIME CONSTANT
	1 :	20 Hz - -
	2 :	150 Hz - -
	3 :	VARIABLE - CONTROLLED BY INT. GENERATOR

Byte No. 30-31:
DCL SKRHSA BIN(2)   NUMBER OF POINTS FOR SMOOTHING IN CHANNEL A FOR A AND A REFERENCE

SKRHSA   1 :   NO SMOOTHING (e.g.)
VALUES CAN BE 1,2,4,8,16,32,64, AND 128

Byte No. 32-33: DCL SKRHSB BIN(2)		NUMBER OF POINTS FOR SMOOTHING IN CHANNEL B FOR B AND B REFERENCE
SKRHSB	1 :	NO SMOOTHING (e.g.) VALUES CAN BE 1,2,4,8,16,32,64, AND 128

Byte No. 34-35: DCL 1 DSPKAN, 2 DSPKA, FUNCTION 2 DSPKB		CURVES DISPLAYED ON CRT AND WHETHER DOT CONNECTION MIGHT BE IN
DSPKA	1 :	CURVE A ON
	2 :	A REFERENCE ON
	3 :	A AND A REFERENCE ON
	=$10 :	DOT CONNECTION FOR A AND A REFERENCE
DSPKB	1 :	CURVE B ON
	2 :	B REFERENCE ON
	3 :	B AND B REFERENCE ON
	=$10 :	DOT CONNECTION FOR B AND B REFERENCE

Byte No. 55-56: DCL CURSOR BIN(2)		CURSOR ON/OFF
CURSOR	$000C :	CURSOR ON
	$0060 :	CURSOR OFF

STATUS GENERATOR + FILTER

HARDW GENRO		0 :	START FREQUENCY EXP.
FEEC0		1 :	START FREQUENCY MANT.
		2 :	END FREQUENCY EXP.
		3 :	END FREQUENCY MANT.
		4 :	SWEEP RATE
		5 :	GENERATOR CONTROL
		6 :	ATTENUATOR
		7 :	FILTER CONTROL
SOFTW GENRAT		0 :	START FREQUENCY EXP.
40984		1 :	START FREQUENCY MANT.
		2 :	END FREQUENCY EXP.
		3 :	END FREQUENCY MANT.
		4 :	SWEEP RATE
		5 :	GENERATOR CONTROL
		6 :	ATTENUATOR
		7 :	FILTER CONTROL
		8 :	FILTER STATUS
		9 :	GSYNC
	GEMANF	10 :	MAN FREQUENCY EXP.
		11 :	MAN FREQUENCY MANT.
		12 :	SWTST
		13 :	SINE PULSE

GENERATOR 2 - COMMUNICATIONS SPECIFICATION

										Always read as double byte (Freq.exp., Freq.mant)	
FEEC0	R/W	0/X	0/X	0/X	0/X	FE3	FE2	FE1	FE0	Freq.exp.	If generator is running
1	R	"1"	FM6	FM5	FM4	FM3	FM2	FM1	FM0	Freq.mant	If generator is running
1	W	X	FM6	FM5	FM4	FM3	FM2	FM1	FM0	Freq.mant.	If generator is running
0	R/W	0/X	0/X	0/X	0/X	WD11	WD10	WD9	WD8	Seq.12-bit data	If generator is stopped
1	R/W	WD7	WD6	WD5	WD4	WD3	WD2	WD1	WD0	To/from waveform RAM	If generator is stopped
2	R	0	0	0	0	FE3	FE2	FE1	FE0	Frequency exponent	
2	W	X	X	X	X	SFE3	SFE2	SFE1	SFE0	Stop frequency exponent	
3	R	X	X	X	X	X	X	X	X	Stop generator, clear waveform RAM address	
3	W	X	SFM6	SFM5	SFM4	SFM3	SFM2	SFM1	SFM0	Stop frequency mant.	
4	R	X	X	X	X	X	X	X	X	Start generator	
4	W	SRE2	SRE1	SRE0	SRM4	SRM3	SRM2	SRM1	SRM0	Sweep rate exp./mant.	
5	R	SP	0	0	0	0	0	0	0	07=1 sweep proceding	
5	W	SWE	SWU/$\overline{D}$	P	$\overline{DGF}$	$\overline{EXT}$	P	P	V/$\overline{CO}$	Generator control word, see below	
6	R	X	X	X	X	X	X	X	X	No function	
6	W	ATT7	ATT6	ATT5	ATT4	ATT3	ATT2	ATT1	ATT0	Output attenuator	
7	R	X	X	X	X	X	X	X	X	Start sweep	
7	W	FCW7	FCW6	FCW5	FCW4	FCW3	FCW2	FCW1	FCW0	Filter characteristics control	

When generator is stopped, there is access only to the waveform RAM. Writing to waveform RAM spoils frequency register contents.

GENERATOR CONTROL:

SWE	Sweep enable/$\overline{\text{constant freq.}}$	$\overline{EXT}$	$\overline{\text{External input to amplifier}}$
SWU/$\overline{D}$	Sweep up/$\overline{\text{down}}$	P1-3	P.A. contr.
$\overline{DGF}$	$\overline{\text{Delete generator filter}}$	V/$\overline{CO}$	Voltage/$\overline{\text{current output}}$

FLTST2(1) = $00	OFF A/B
= 01	A ON - B OFF
= 03	A ON - B ON
= 02	A OFF - B ON

FLTST2(2) = $FE	5th harmonic
$FD	4th -
$FC	3rd -
$FB	2nd -
$FA	1st -
$FF	6th -
$ 0	7th -
$ 1	8th -
$ 2	9th -
$ 3	10th -
$ 4	11th -
$ 5	12th -

FLTST2(1) = %1XXXXXXX	Generator started
%0XXXXXXX	Generator stopped