Operating Manual

Amplifier system

MGC*plus*

Operation with computer or terminal

B 31.MGCPR.40 en

HBM

Contents

Safety instructions

Use in accordance with the regulations

The amplifier system is to be used exclusively for measurement tasks and directly related control tasks. Use for any purpose other than the above shall be deemed to be not in accordance with the regulations.

To ensure safe operation, the device may only be operated in accordance with the information given in the Operating Manual. It is also essential to comply with the legal and safety requirements for the application concerned during use. The same applies to the use of accessories.

General dangers due to non-observance of the safety instructions

The amplifier system is a state-of–the-art device and is fail-safe. The device may give rise to further dangers if it is inappropriately installed and operated by untrained personnel.

Any person instructed to carry out installation, commissioning, maintenance or repair of the device must have read and understood the Operating Manual and in particular the technical safety instructions.

Remaining dangers

The scope of supply and list of components provided with the amplifier system cover only part of the scope of measurement technique. In addition, equipment planners, installers and operators should plan, implement and respond to the safety engineering considerations of measurement technique in such a way as to minimise remaining dangers. Prevailing regulations must be complied with at all times. There must be reference to the remaining dangers connected with measurement technique.

After setting up and password protected activities, care must be taken to ensure that any control units that may be connected remain in a safe and secure state until the switching behaviour of the amplifier system has been checked.

Any risk of remaining dangers when working with the amplifier system is pointed out in this introduction by means of the following symbols:

Symbol:  **DANGER**

Meaning: **Maximum danger level**

Warns of a **decidedly** dangerous situation in which failure to comply with safety requirements **will** lead to death or serious physical injury.

Symbol:  **WARNING**

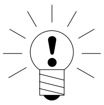
Meaning: **Possibly dangerous situation**

Warns of a **potentially** dangerous situation in which failure to comply with safety requirements **can** lead to death or serious physical injury.

Symbol:  **CAUTION**

Meaning: **Dangerous situation**

Warns of a possibly dangerous situation in which failure to comply with safety requirements **could** cause damage to property or result in some kind of minor physical injury.

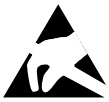
Symbol:  **NOTE**

Means that important information about the product or its handling is being given.

Symbol:

Meaning: **CE mark**

The CE mark enables the manufacturer to guarantee that the product complies with the requirements of the relevant EC guidelines (see Declaration of conformity at the end of this Operating Manual).

Symbol: 

Meaning: **Component may be damaged by electrostatic discharge**

This means that the PCMCIA hard disk (optional) must be protected against static discharges by securing it in place with the fasteners provided (CP32B only).

Working safely

Error messages must only be acknowledged when the cause of the error has been removed and no further danger exists.

Conversions and modifications

The amplifier system must not be modified from the design or safety engineering point of view except with our express agreement. Any modification shall exclude all liability on our part for any resulting damage.

In particular, any repair or soldering work on motherboards is prohibited (this includes changing components other than EPROMs). When exchanging complete modules, use only original parts from HBM.

Qualified personnel

means persons entrusted with the installation, assembly, commissioning and operation of the product who possess the appropriate qualifications for their function.

This instrument must only to be installed and used by qualified personnel, strictly in accordance with the technical data and the safety requirements and regulations listed below. It is also essential to comply with the legal and safety requirements for the application concerned during use. The same applies to the use of accessories.

Safety requirements

Before commissioning, find out whether the mains voltage and current type specified on the identification plate match the mains voltage and current type at the place of use, and whether the circuit being used is adequately protected.

Insert the mains plug only into a socket with a protection switch (Protection Class I). When connecting electrical devices to low voltage: connect to separated extra–low voltage (SELV) only (safety transformer in accordance with DINVDE 0551/EN60742).

Use only the mains cable supplied, which has been provided with a ferrite core.

Before you open the device, make sure it is switched off; remove the mains plug from the socket.

Never pull the mains plug from the socket by the mains cable.

Do not operate the device, if the mains cable is damaged.

If an amplifier channel is to be withdrawn, close off the plug-in unit with a blank plate.

Built-in devices must only ever be operated whilst they are within the housing provided for them.

The device complies with the safety requirements of DIN EN 61010-Part 1 (VDE 0411-Part 1); Protection Class I.

To ensure adequate immunity from interference, use only *Greenline* shielded ducting (see HBM offprint ”*Greenline* shielding design, EMC-compliant measuring cable; G36.35.0”)

The port connections ($\leq$50V) must have an insulating strength of at least 350V(AC).

A Introduction

1 Overview of documentation

The total product documentation for the MGC*plus* amplifier system covers the following publications:

The ***Operating Manual***

explains how to operate the device manually using the AB22A/AB32 display and control panel

Operation with computer or terminal

shows you how to program and measure with a terminal or computer

The manual for the **MGC*plus* Assistant *program***

describes how to assign parameters and control the MGC*plus* system with the aid of a PC and explains how to acquire measurement data

This binder contains the publication *Operation with computer or terminal*.

The Operating Manual is enclosed in a second binder.
Each binder is supplied in a slip–case.

1.1 How to work with this manual

This manual is designed to help you operate the MGC*plus* device quickly with the aid of a computer. It is intended just as much for inexperienced newcomers to the world of PCs as for advanced computer users.

There are several **guidelines** available to you:

- The Table of Contents at the beginning of the manual gives you an initial overview.
- The chapters are separated by handy index tabs (dividers). Each chapter starts with a list of its contents.
- The header tells you which chapter or sub–section you are currently reading.

  Example:

  Commands → *Communication* H-13

  The footer includes the name of the manual (Operation with computer or terminal, Operating Manual, etc.).
- The page numbering is linked to capital letters which correspond to the chapter headings. Each chapter begins with page 1.
- When searching for particular commands, it is best to refer to the command overviews in chapters J and K.
- A list containing all the parameters can be found in chapter I “Global tables”.

Note:

We recommend that inexperienced computer users should also read chapters B to E. Experienced computer users can skip these chapters and continue reading chapter F.

1.2 Operation with computer or terminal

This manual explains how to operate your HBM device with the aid of a terminal or computer.

- If you use a terminal it is possible to interact with the MGC*plus* device.
- You can make all the device settings by sending an appropriate command sequence from the computer. A computer and appropriate programs can be used to automate recurring measurement runs.

In order to match the signals emitted or required by the various items of equipment (computer, plotter, printer, measuring instruments etc.), devices have to be interconnected over interfaces. To allow devices from different manufacturers to be used, interfaces are subject to norms or standards.

Depending on the interface card, you can operate the MGC*plus* system over the following standard interfaces:

the RS232C (V.24)
the RS485 and
the printer (Centronics)[1] and
the IEEE 488–78 interface[2] (also IEC bus interface).

The contents of this manual relate to firmware version P2.10 of the CP12 interface card or P1.12 of the CP32 communications processor.

1) only with CP32 communications processor
2) only with CP32 communications processor and PCMCIA–GPIB board

1.2.1 Overview of the chapters and appendix

Chapter A
Introduction
General notes on using this manual. A documentation overview shows you which documents relate entirely to the MGC*plus* system.

Chapter B
Compares and contrasts the RS232C, RS485, printer and IEEE 488–78 interfaces
Explains which interfaces are built into the MGC*plus* device and tells you the main differences.

Chapter C
RS232C and RS485 interface
Describes both interfaces and how they are assigned.

Chapter D
Printer interface (Centronics)
Describes the interface and how it is assigned.

Chapter E
IEEE 488–78 interface
Describes the interface and how its address is set up.

Chapter F
Communication with the MGC*plus* device
Explains how you can activate the various interfaces and how you connect the MGC*plus* device to your computer.

Chapter G
Program creation
For creating your own programs, this chapter suggests some routines for programming the interfaces of IBM-PC's and compatibles.

Chapter H
HBM Interpreter commands
Important conventions on the structure and notation of commands are introduced. The chapter ends with the complete set of HBM Interpreter commands.

Chapter I
Global tables
Summary of the value tables.

Chapter J
Overview of commands in alphabetical order
To help you find your way around more quickly, there is an overview of commands in alphabetical order.

Chapter K
Overview of commands by function
You can also refer to an overview of commands arranged by function.

B Comparison of interfaces

Up to three different interfaces can be built into the MGC*plus*, depending on the interface card.

- an RS232C,
- an RS485
- a printer and
- an IEEE 488-78 (optional)
- Ethernet (optional[2])

However, the device can still only be operated via one interface at a time, each of them having specific advantages, but also disadvantages.

This should make it easier for you to decide on the right interface for your application. You can find out how the interfaces work in the respective chapters.

General features of interfaces:

RS232C interface (Chapter C)

- Serial data transfer
  - Transmission speed relatively ”low”.
  - In the most basic case, requires a 3–wire cable for transmission in both directions (duplex or bi–directional).
  - Only one device can be connected.

RS485 interface (Chapter C)

- Serial data transfer
  - Transmission speed same as the RS-232-C.
  - Needs a 4–wire bus.
  - Several devices can be connected to a serial port on your computer.
  - Possible distance between MGCplus and computer up to 1000 m.

Printer port[1] (Chapter D)

- Parallel data transfer
  - Transmission speed ”medium”.
  - Requires a special 11–wire cable (Interlink) for transmission in both directions (included as part of standard supply).
  - Only one device can be connected.

[1] only with CP32 communications processor
[2] In preparation

IEEE 488-78 interface[4] (optional) (Chapter E)

- Parallel data transfer
  - Transmission speed relatively ”high”.
  - Several devices can work in one link–up.

Data transfer is faster in the case of parallel interfaces than serial interfaces. Furthermore, a parallel interface is advantageous for communication with several devices, as a serial bus quickly reaches the limits of its capacity here.

Link		Max. distance	Interface
from	to		
Terminal / computer	Measuring device	20 m	RS232C[1]
Terminal / computer	Measuring device	endless	RS232C/modem
Computer	one or more measu-ring devices	1200 m	RS 485
Computer	Measuring device	2 m	Printer [3]
Computer	one or more measu-ring devices	2...20 m	IEEE 488-78[2] [4] (Maximum distance between individual measuring devices 2 m)

Tab. 1: Transmission path length in the case of different interfaces

The interfaces are described in detail in the following two chapters.

[1] By interconnecting modems, you can increase the distance as you wish.
[2] Extenders can be used to increase distances.
[3] only with CP32 communications processor
[4] only with CP32 communications processor and PCMCIA-GPIB board

C RS232/RS485/422 serial interfaces

1 RS232C and RS485/422 interfaces

Both of these interfaces are known as 'serial interfaces', as data is transferred between them bit by bit. They differ from each other in the voltage level used and in their construction; the RS232C-interface is for point–to–point connections, the RS485/422 is suitable for bus operation.

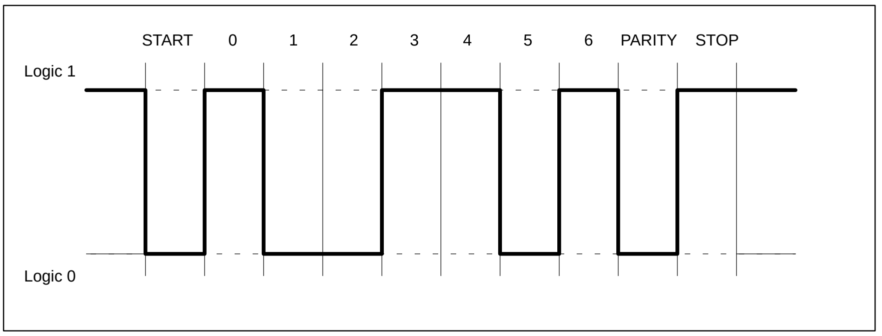


Fig. C 1: Line level of the character Y in the case of negative logic

A START bit is set before each character (data byte). Then the data bits and a STOP bit follow. As data is transferred sequentially, the transmission speed must match the reception speed.
The number of bits per second is called the baud rate. The exact baud rate of the receiver is synchronised with the START bit for each byte transferred. Then the data bits, which all have the same length, follow. Once a STOP bit is received, the receiver goes into waiting state until it is reactivated by the next START bit.

Data transfer is controlled with the software handshake X-ON (DC1) and X-OFF (DC3).

If the device is ready to transfer data, it sends the control character X-ON (DC1) via the data line. If it cannot accept any data, e.g. if the buffer is occupied, the control character X-OFF (DC3) is sent.

Characteristics of serial interfaces

Word length	8 bits
Stop bit	1
Parity	Even[2)]
Baud rate	300[2)], 600[3)], 1200[2)], 2400[3)], 4800[3)], 9600* [2)], 19 200[2)], 38400[4)], 57600[4)]
Software handshake	X-ON, X-OFF

* Factory setting
[2)] to be set up with DIP switches(CP12)
[3)] to be set up with BDR command only (CP12)
[2,3,4)] to be set up with AB22A only or the command BDR (CP32)

1.1 RS485/422 bus

The bus shown here represents an extension to the RS422 interface, where, in contrast to the latter, several transmitters are allowed. Each MGC*plus* has a receiver and a transmitter and the latter can be switched on and off using SELECT commands.

You can connect up to 32 devices to this 4–wire bus. In contrast to RS232C interfaces, only **one** computer interface is necessary to connect several devices. You only need a serial converter if the computer has no RS485/422 interface (see Chapter E).

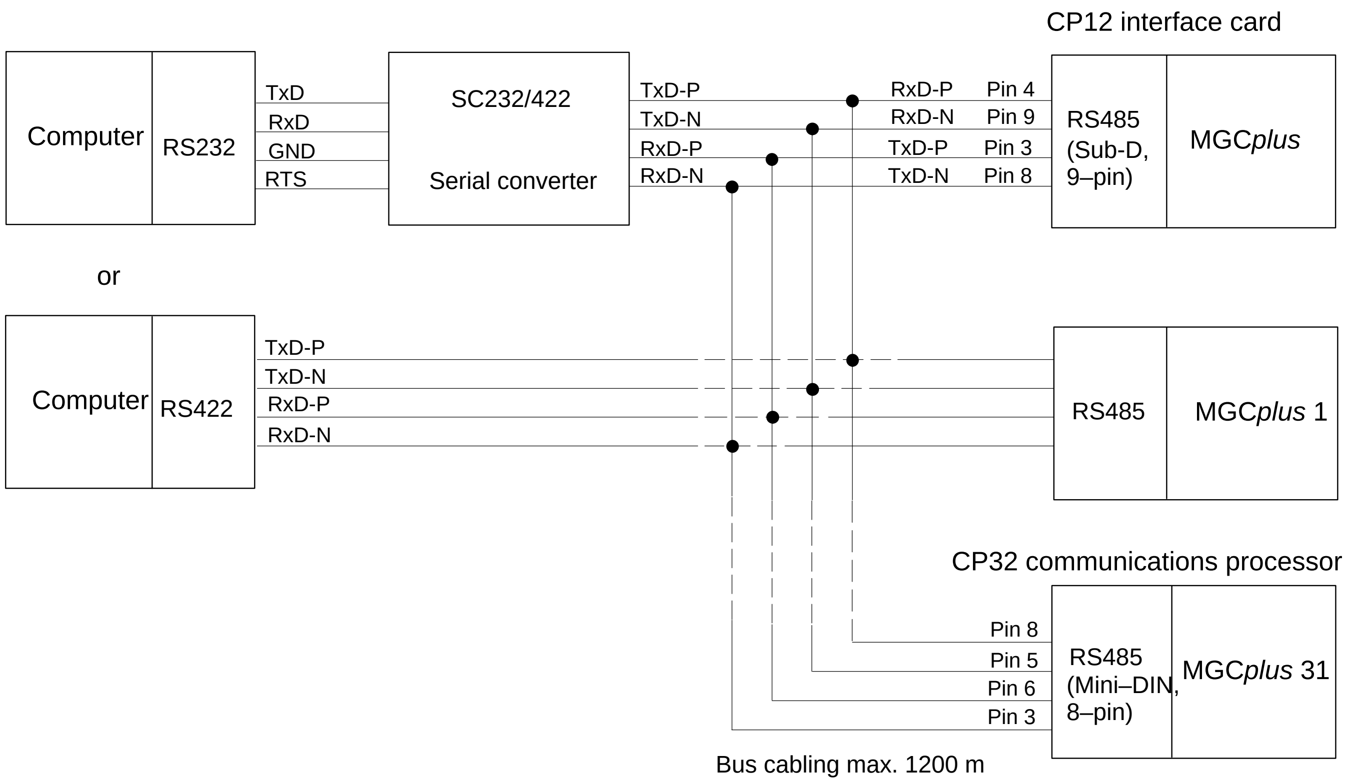


Fig. C 2: The RS485/422 bus with and without a serial converter

D Parallel printer port (Centronics)

1 Printer port

This interface is normally used for connecting a printer to the computer system. Data is transferred parallel – i.e. in bytes (8 bits =1 byte) – and asynchronously in the handshaking procedure. It is only suitable for cable lengths of up to 2m, because of the voltage level used.

This interface was developed by the Centronics company and is to be regarded as 'quasi-standard'. As it is not subject to any Standard, both the Amphenol 36–pin connector and the 25–pin sub–D connector are permissible.
In the meantime, HBM has decided on the 25–pin sub-D connector which is more commonly available.
A special 11–pin cable[1)] (Interlink) is necessary for data transfer.

1) Included in standard supply of CP32 communications processor.

1.1 Handshaking procedure

All data transport on the data bus is fed via three lines (three–line handshaking procedure).

STROBEX
The MGC*plus* applies the data to the bus and communicates that is valid or available with the STROBEX signal.

ACKNX
The computer acknowledges receipt of the data with the ACKNX signal. The MGC*plus* can now send the next data item.

BUSY
If the computer is still busy processing the data received previously, this is indicated to the MGCplus with the BUSY signal. If the computer has finished processing, it signals that it is ready to receive with the ACKNX signal.

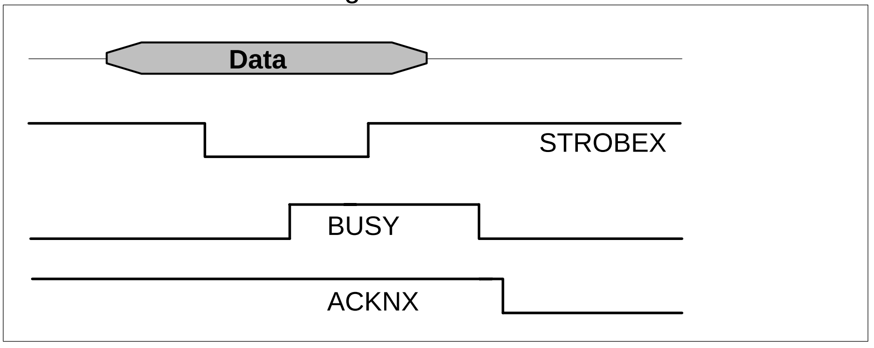


Fig. D 1: Handshaking diagram

E IEEE 488-78 parallel interface

1 IEEE 488-78 bus

The IEEE 488-78 interface was developed as a bus system for connecting computers and measuring devices. It is standardised and many manufacturers adhere to the same features.

In practice, two types of plug connector have won through:
the 24–pin Amphenol connector in accordance with the US Standard and the 25–pin miniature D connector in accordance with the international Standard IEC 625.
In electrical terms, both connectors are equal, but the 24–pin version has the disadvantage of being easily confused with an RS232C connector. This is why HBM uses the 24–pin Amphenol connector on the PI12 interface card.
In the case of the PCMCIA-GPIB board (can be used with the CP32 communications processor), the 24–pin Amphenol connector is integrated into the connection cable.
This bus allows you to interconnect 15 devices in parallel to one system. Data is transferred parallel and asynchronously in the handshaking procedure.

The bus consists of 16 lines which are divided up into three groups:

8 data lines
3 handshaking lines
5 control lines

The bus interconnects the devices electrically. Commands and data from the device are transferred using this. The commands are split into two groups:

- **Device commands:**
Are transferred from the controller to one or more devices where they are analysed and processed.

- **Interface commands:**
Are used for controlling the interface system.
If several devices are interconnected in one system, we recommend connecting them in a chain and not a star formation.

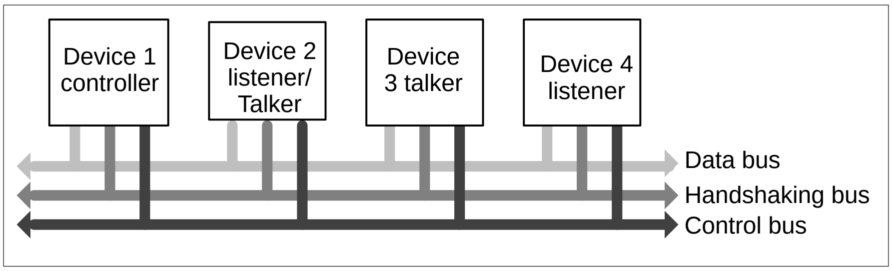


Fig. E 1: The IEEE 488-78 bus

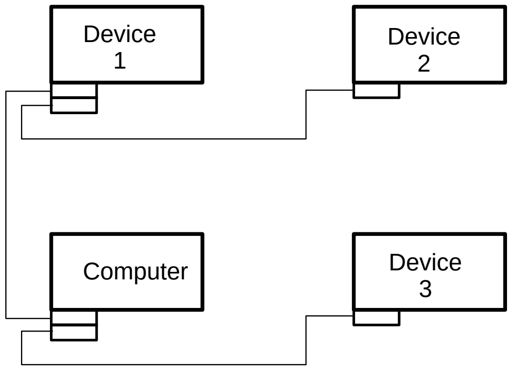


Fig. E 2: Correct wiring of devices with one another

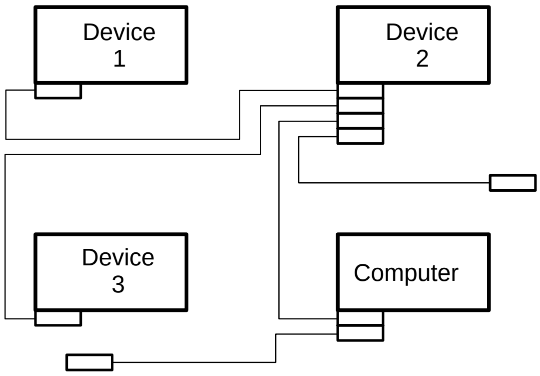


Fig. E 3: Incorrect wiring of devices with one another

1.1 Data lines

For transferring data in 8 bit lengths, the IEEE 488-78 bus uses eight data lines, DIO1 to DIO8. Interface commands are also transferred via these lines. In this case, however, at least one control line is used in addition.

1.2 Handshaking lines

All data transport on the data bus is fed via three lines (three–line handshaking procedure).

DAV
Data **V**alid signals the validity and availability of data on the databus. A signal is output by the talker to advise the listeners that the data can be accepted.

NRFD
Not **R**eady **F**or **D**ata is set by the listeners (HBM device) and advises the controller that devices are not yet ready to receive data.

NDAC
No **D**ata **Ac**cepted indicates the status of the receiver (’listener’). They output a signal to indicate to the talker that the valid data byte has not yet been accepted. The talker cannot send another data byte until all of the listeners have accepted the byte.

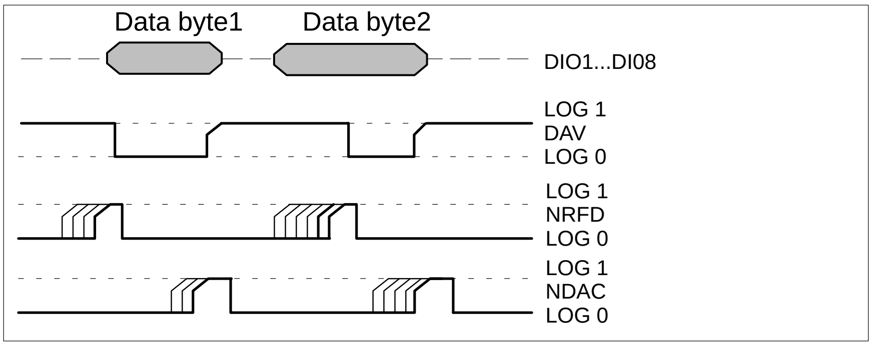


Fig. E 4: Handshaking diagram

1.3 Control lines

Control lines are necessary for the smooth flow of information via the interface:

IFC
Interface Clear puts the interface system into a precisely–defined initial status. It is sent exclusively by the controller.

SRQ
Service Request can be set by any device that is connected if it has a message for the controller. The controller can then interrupt and query the cause.

ATN
Attention can only be set by the controller and tells the devices connected whether data or interface messages are being transferred.

REN
Remote Enable switches off the front panel operation of all HBM devices that are connected.

This line must be active for remote control operation.

EOI
End Or Identify has two functions.

- It signals the end of a data transfer to the listener.
- The controller can initiate a parallel poll in conjunction with ATN.

2 Controller, talker and listener functions

Each device connected to the bus must have at least one of the following functions:

Controller
Usually a computer which activates the devices connected.

Listener
A device capable of receiving data (e.g. a printer). Several devices can be addressed simultaneously as listeners.

Talker
A device capable of transmitting data (e.g. a voltmeter). Only one talker may be active at the bus.

You can interconnect one talker (e.g. a voltmeter) and one listener (e.g. a printer) as a minimum set–up. With this, data flow is possible in one direction.
If a computer which operates as a controller is also connected to this minimum setup, measurement runs, for example, can be programmed and controlled from the computer. It is also possible to connect a number of computers which function as controllers in such a device connection setup.
However, only one computer can work as the controller. It can then hand its controller function over to another computer at any time.
The devices which are connected can also be addressed with interface commands and programmed accordingly.

3 Addressing

All devices working together in a link–up must have their own separate address so that the controller can clearly distinguish between them.

As a rule, this is set on the device with a switch using numbers from 0 to 30. Each device can be addressed using its individual address.

4 Service request (SRQ)

In a link–up of several measuring devices it can happen that a measurement sequence needs more time than usual. The controller waits until the end of this process. However, all other devices connected are blocked during this period. Nevertheless, in order to give other important processes the possibility of signalling, there exists the so–called 'Service Request'.

In functional terms:
A device requiring immediate processing (e.g. in the case of overtravel) sends an SRQ, upon which the controller interrupts the action currently running and determines with a serial or parallel poll which device made the SRQ. This device is then requested to release its data or give the reason for the service request.
Theoretically, in an IEEE bussystem, a maximum of 14 devices can make a service request simultaneously. The controller must then find out which device is making the request by means of a serial poll.
In the case of a parallel poll, eight devices can be polled at the same time. In the case of a serial poll, as opposed to a parallel poll, the reason for the SRQ can also be transferred.

4.1 Serial Poll (SPOLL)

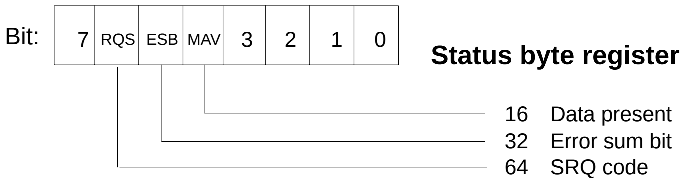
If a service request arises during a measurement, the computer can detect which device caused the interruption by starting a serial poll. The computer must address all the devices and query the status value. The computer can also detect the cause of the interruption in this way. A serial poll takes longer than a parallel poll.

A device must have the SR interface function in order to be able to respond to a serial poll. It must also be able to work as a talker and be able to decode the two interface commands SPE and SPD.

If the controller has detected an SRQ, it must start up a serial poll routine. To do this, the devices are put into serial poll status with the command SPE. Then the devices are addressed in turn to be able to set the status byte.

If the device addressed as talker has activated the SRQ line, it must activate the data line DIO7 and release the SRQ line. The other lines, DIO1 to DIO6 and DIO8, can be used for transferring status bits. After polling the device(s), the command SPD is sent by the controller to re–establish normal status.

The status bits transferred during a serial poll have the following significance:



RQS Request Service ESB Event Status Bit MAV Message Available

4.2 Parallel poll (PPOLL)

The computer can also detect which device caused an interruption in the case of a parallel poll. However it cannot detect the cause.

The PPOLL interface function is carried out in the MGC*plus*. To do this, one of the eight DIO lines must be allocated to the devices connected. This puts the controller in a position to poll eight devices simultaneously.
In the case of a parallel poll, the controller receives no indication of the cause of the SRQ. If the controller transmits the identification command (EOI + ATN = IDY = True), the connected devices must activate the data lines allocated to them. In the case of HBM devices, activation takes place as a result of the interface command PPM.

5 Interface commands

These commands do not belong to the device commands described later. They only relate to the IEEE interface. The most important of these interface commands are listed in the following table.

These commands are to be output in ASCII code via the interface. If this does not happen automatically in the computer, an appropriate translation routine is necessary.

Universal commands	**These are effective on all connected devices, if the ATN line is active.**
DCL - Device clear	Puts all devices into a precisely defined on–state.
SPE - Serial Poll Enable	If a serial poll is initiated, the device addressed as talker sends its status byte.
SPD - Serial Poll Disable	Deactivates serial poll status.
Addressed commands:	**These are only effective in the case of devices just addressed by the controller.**
GTL - GoTo Local	Puts the addressed device back into manual status. Switches off remote control operation.
SDC - Selected Device Clear	Puts one or more devices into the defined initial status.

Addressing commands:	
TAG - talker Address group LAG - Listener Address group	Addresses the device as a talker. Addresses the device as a listener.
De–addressing commands:	
UNL - Unlisten UNT - Untalker	Deletes all addresses as listeners. Deletes the address of the talker last active.

6 IEEE 488interface function in the case of HBM devices

The HBM Interpreter of your device is equipped with the following interface functions:

Interface function	Abbrevia-tion	Significance
Source handshake	SH1	All functions are included.
Accepter handshake	AH1	All functions are included.
Talker	T1	All functions are included.
Listener	L1	All functions are included.
Service request (SRQ)	SR1	All functions are included.
Remote/Local	RL2	All functions are included, with the exception of the ”Remote control with lock” setting.

Connector:
Cannon 24–pin connector (IEEE 488-78 standard) on the PI12 interface card or the connecting cable of the PCMCIA-GPIB board in the case of the CP32 communications processor.

Parallel poll	PP2	The configuration of the parallel poll response must take place via the HBM command PPM. After this a normal parallel poll can be executed. It is not possible to carry out a configuration using interface commands such as PPC and PPU etc.
Device Clear	DC1	All functions are included.
Device Trigger	DT1	All functions are included.
Controller	CO	No function provided.

F Communication with the MGCplus

MGC*plus* operation with computer or terminal

1 HBM Interpreter

The HBM Interpreter "translates" commands and associated character strings received by an interface into a code which the MGC*plus* can understand.

You can call the Interpreter over the following interfaces:

RS232C
RS485
Printer and
IEEE 488–78 (optional)
Ethernet[1)] (optional)

As soon as it has been called by an interface, access over the other interfaces is disabled.
The commands and generated data are identical for all interfaces.
All HBM Interpreter commands are listed in chapter H, "HBM Interpreter commands".

[1)] In preparation

2 Activating the RS232C interface

The HBM Interpreter is activated by the following control characters:

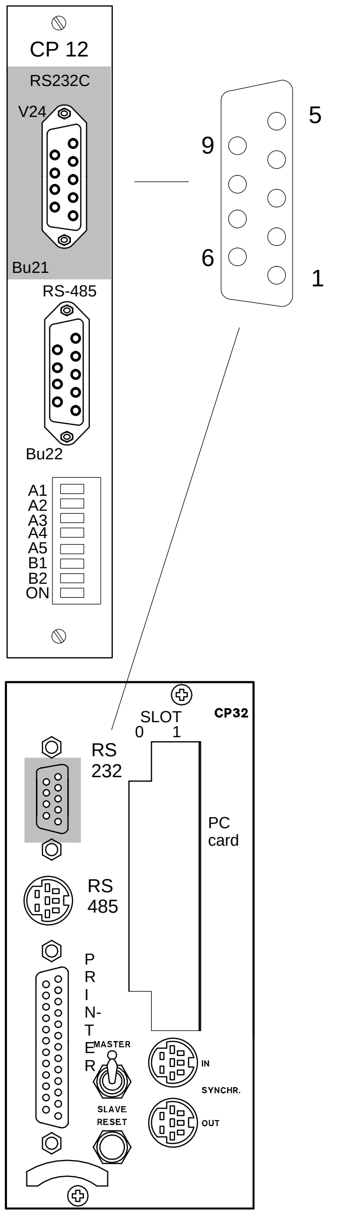
- CTRL B (STX) – computer operation without echo
- CTRL R (DC2) – computer operation without echo

When one of these control characters is input, the device goes into a remote control operation status known as ”Computer Control” during which nothing except its display functions can still be operated by the AB22A/AB32 operating panel.

Computer operation without echo means:
Generated data will be returned to the MGC*plus* but command characters will not.
With an RS232C interface, all generated information is output as soon as it has been fully loaded into the output buffer.
You can use the following commands to deactivate the remote control operation status:

DCL **or**
CTRL A (SOH) or
with the commands RES and RST

2.1 Interface assignment



The RS232C (V.24) and RS485 interface jacks are located on the CP12 interface card and the CP32 communications processor. The interfaces may only be used alternately.
The RS485 interface is provided for bus mode.

RS232C (V.24) interface assignment

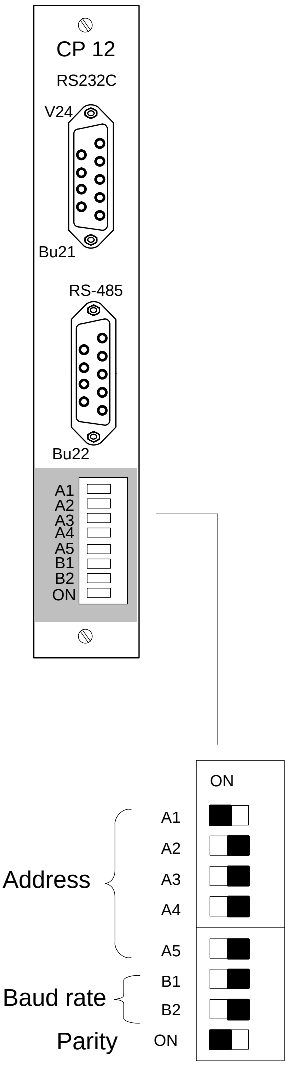
Remotes	Assignment
1	Not used
2	RD
3	TD
4	Not used
5	RTS (internal 3kOhm at +10V)[1)]
6	DTR (internal 3kOhm at +10V)[1)]
7	Ground
8	External print enabling [2)] (active low, internal 10kOhm at +5V)
9	+5V
Shield	Connection to housing by connector gland

1) No function on the MGC*plus* device in the case of a CP32 communications processor at +5V
2) Not used in the case of the CP32 communications processor

Assignment **SLOT 0**: Hard disk

Assignment **SLOT 1**: IEEE488 interface

2.2 Setting up the interface



You can use the eight DIP switches on the CP12 interface card to set up:

the baud rate and
parity
.

Baud rate

Baud rate	B1	B2
300	ON	ON
1200	OFF	ON
9600[1)]	OFF	OFF
19200	ON	OFF

Parity

Parity	Switch position
EVEN[1)]	ON
NONE	OFF

[1)] Factory setting

The device address has no significance for this interface (only one device is connected).

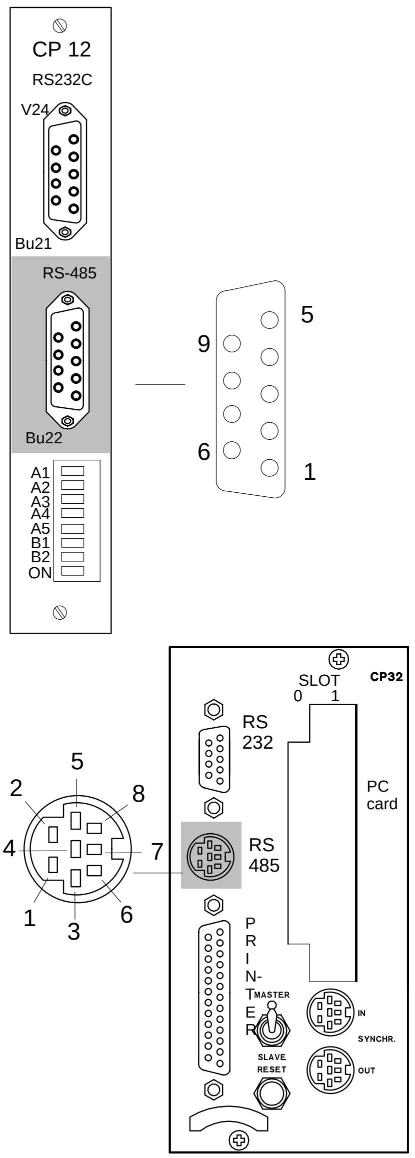
If you use a CP32 communications processor, this and other settings can be entered using the AB22A or the BDR command.

3 Activating the RS485 interface

The HBM Interpreter is activated or deactivated by the same control characters as those used with the RS232C interface. Similarly, the information demanded is output as soon as it has been fully loaded into the output buffer.

HBM devices are set up with address ”1” in the factory. If several devices are operated on this bus, they must be set up with different addresses.
Particular devices are activated by using the Select command (Sxx).

3.1 Interface assignment



MGC*plus* operation with computer or terminal

RS485 interface assignment

Remotes		Assignment
CP12	CP32	
1	–	+5V
2	4	Ground
3	6	TxD–P send data (+)
4	8	RxD–P receive data (+)
5	–	Ground
6	1,2,7	Not used
7	–	Ground
8	3	TxD–N send data (–)
9	5	RxD–N receive data (–)
Shield	Shield	Connection to housing by connector gland

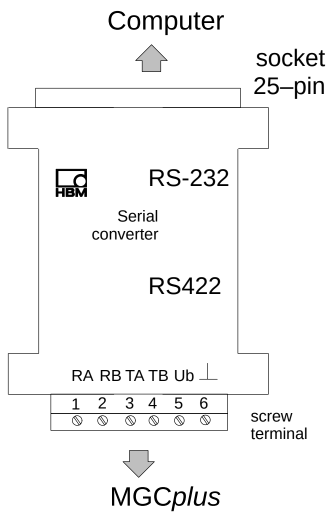
You can connect the MGC*plus* to a computer over the RS485 interface, and use it to address up to 32 devices. For this you will need a cable with a 9–pin Sub-D or 8–pin MiniDIN connector, unterminated. The unterminated cables can also be connected to the screw terminals on a Serial Converter SC232/422 (see HBM product catalogue), if the computer has no RS485/422 interface.

- Computer with 25–pin RS232 socket:
  connect converter directly to the computer.

- Computer with 9–pin RS232 socket:
  Fit the Kab 413 adapter plug onto the converter and connect it to the computer.

Assignment **SLOT 0**: Hard disk

Assignment **SLOT 1**: IEEE488 interface



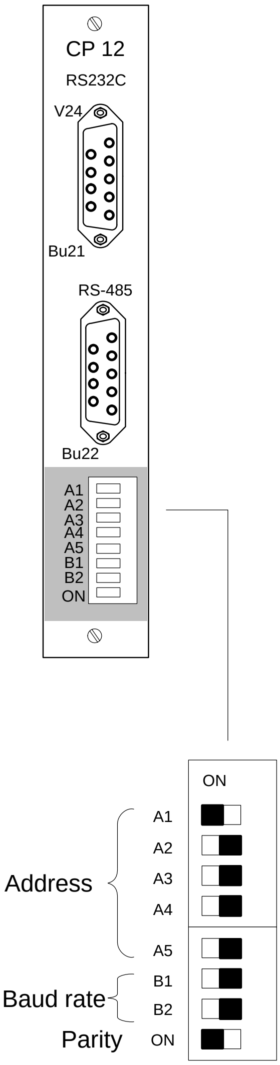
Converter assignment

Screw terminal (converter)	**Wire colour** (unterminated)	**MGC*plus***	
		CP12	CP32
1	wh	8	3
2	bk	3	6
3	bu	9	5
4	rd	4	8
5	+ *	–	–
6	– *	–	–

* Supply voltage (9V DC) for the serial converter.

A plug–in power pack is included among the components supplied with the converter.

3.2 Setting up the interface



You can use the eight DIP switches on the CP12 interface card to set up:

the device address, the baud rate and the parity

Device address

	A1	A2	A3	A4	A5
0	OFF	OFF	OFF	OFF	OFF
1[1)]	ON	OFF	OFF	OFF	OFF
2	OFF	ON	OFF	OFF	OFF
3	ON	ON	OFF	OFF	OFF
.	.	.	.	.	.
31	ON	ON	ON	ON	ON

Baud rate

Baud rate	**B1**	**B2**
300	ON	ON
1200	OFF	ON
9600[1)]	OFF	OFF
19200	ON	OFF

Parity

Parity	**Switch position**
EVEN[1)]	ON
NONE	OFF

If you use a CP32 communications processor, this and other settings can be entered using the AB22A or the BDR and ADS commands.

[1)] Factory setting

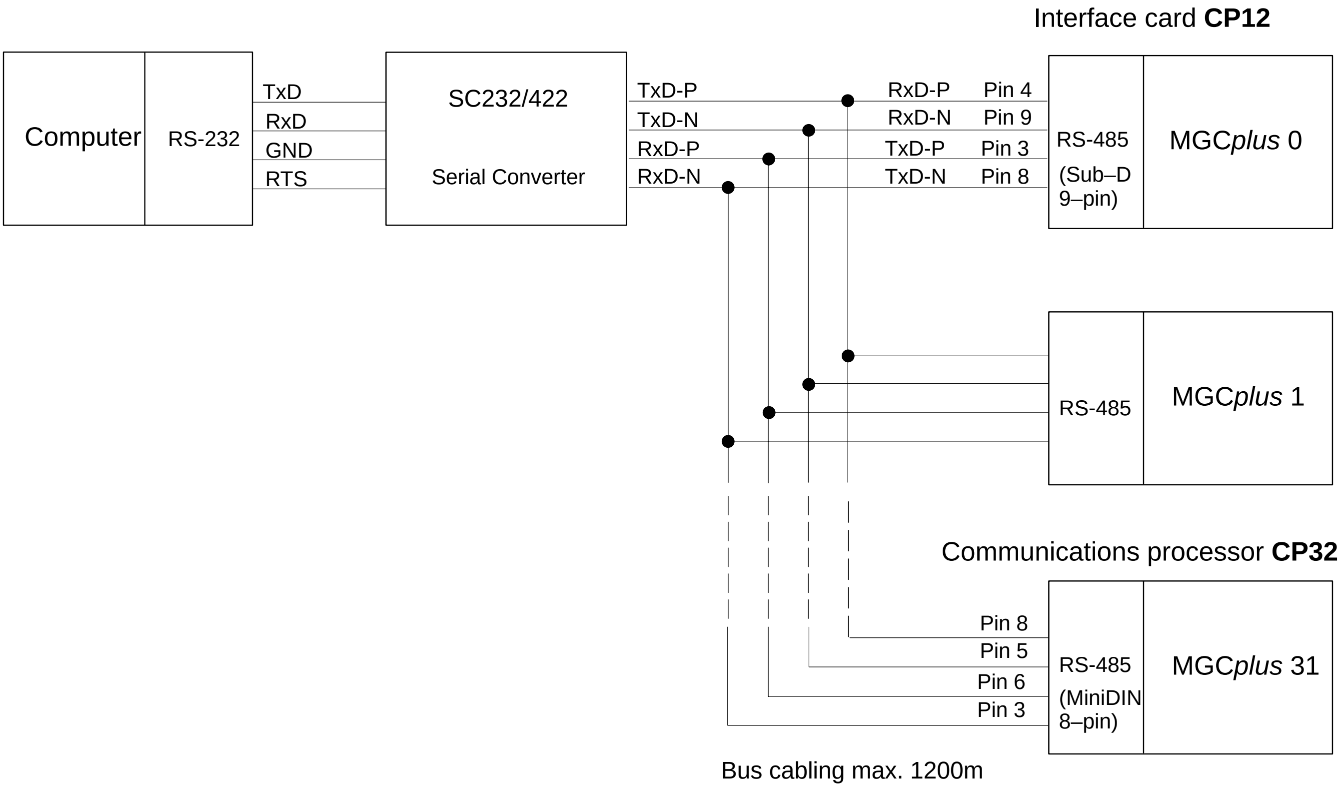
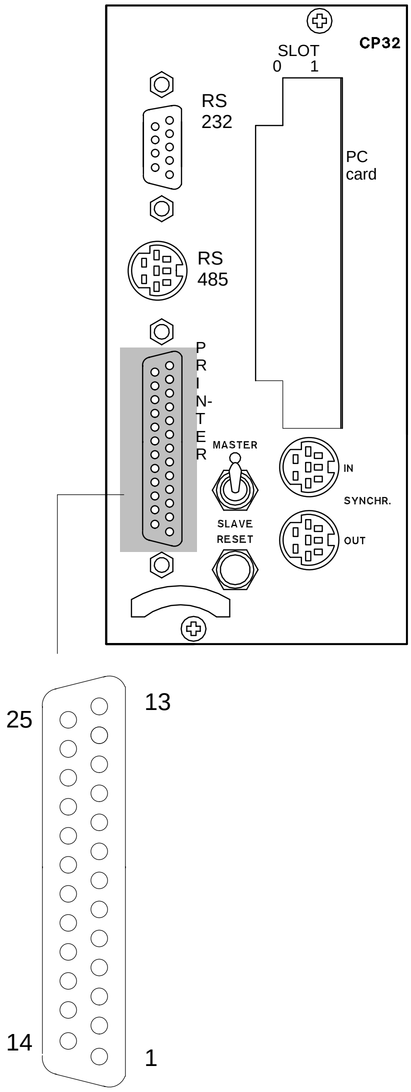


Fig. E1: RS485/422 bus with RS232C interface in the computer and Serial Converter SC232/422

4 Activating the printer interface

The HBM Interpreter is activated or deactivated by the same control characters as those used with the RS232C interface. Similarly, the information demanded is output as soon as it has been fully loaded into the output buffer.

4.1 Interface assignment



Printer interface assignment

Remotes	Assignment
1	STROBE X
2	Data bit 0
3	Data bit 1
4	Data bit 2
5	Data bit 3
6	Data bit 4
7	Data bit 5
8	Data bit 6
9	Data bit 7
10	ACKNX
11	BUSY
12	PAPER END
13	SELECT
14	AUTO LINE-FEEDX
15	ERRORX
16	INITIALIZE PRINTERX
17	SELECT INX
18...25	GROUND
Shield	Connection to housing by connector gland

Data transmission requires an 11–pin special cable (Interlink), which is included among the components supplied with the CP32 communications processor.

Assignment **SLOT 0**: Hard disk

Assignment **SLOT 1**: IEEE488 interface

5 Activating the IEEE 488–78 interface

The interface message Remote Enable (REN, which is usually set automatically by the computer when the IEEE 488–78 interface is initialised) activates the HBM Interpreter in conjunction with the initial command.
The MGC*plus* then goes into remote control mode, during which nothing except its display functions can still be operated by the AB22A/AB32 operating panel.

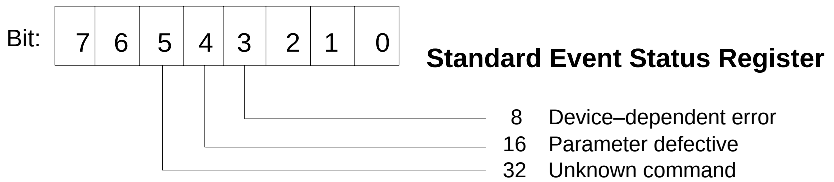
You may use one of the following commands to enable the operating panel again. You then exit from the HBM Interpreter.

- HBM command DCL
- Interface message DCL (Device Clear)
- Interface message GTL (Go To Local)
- Switch off the REN line

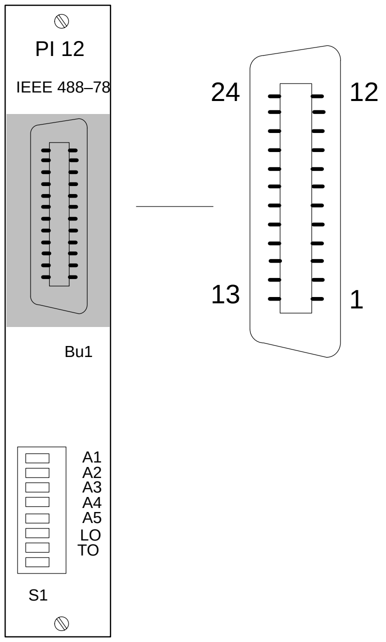
This puts the device into local status. You can now operate the MGC*plus* again using the AB22A/AB32 display and control panel.
The address set up in the factory for HBM devices is 4 (see page F-17).
If several devices are operated on this bus, they must be set up with different addresses.

The computer must respond to the message **Data Present** with **INPUT** or **ENTER** and then accept the data. The HBM device cannot interpret the next command until all the data connected with the current command has been output.

Errors are recorded in the Standard Event Status Register and may be queried with the HBM command *ESR? .
Errors are also summarised in the Status Byte Register under an error sum bit (ESB). When the Standard Event Status Register is read out, the individual error bits and the sum bit are cleared.



5.1 Interface assignment

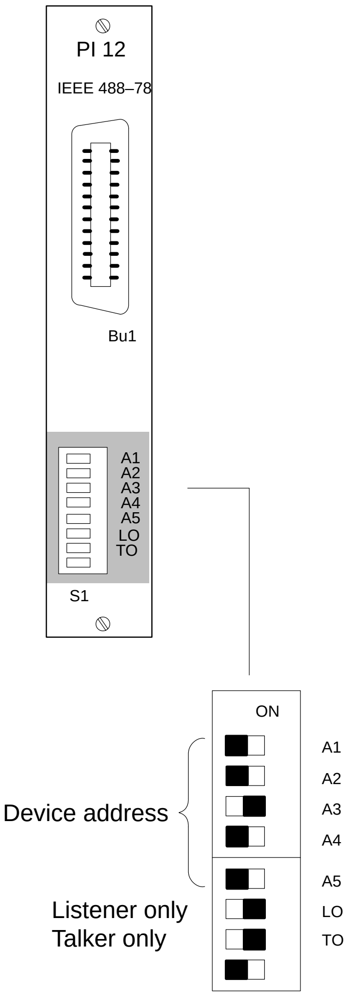


Assignment IEEE interface 4888[1)]

Remotes	Assignment
1	DIO1
2	DIO2
3	DIO3
4	DIO4
5	EOI
6	DAV
7	NRFD
8	NDAC
9	IFC
10	SRQ
11	ATN
12	Shield
13	DIO5
14	DIO6
15	DIO7
16	DIO8
17	REN
18	Ground 6
19	Ground 7
20	Ground 8
21	Ground 9
22	Ground 10
23	Ground 11
24	Ground

1) applies only to CP32 communications processor with PCMCIA–GPIB board (National Instruments, not included on list of components supplied)

5.2 Setting up addresses on the MGC*plus*



You can use the eight DIP switches on the PI12 interface card (optional) to set up:

the device address and the interface configuration.

	A1	A2	A3	A4	A5
0	OFF	OFF	OFF	OFF	OFF
1	ON	OFF	OFF	OFF	OFF
2	OFF	ON	OFF	OFF	OFF
3	ON	ON	OFF	OFF	OFF
4[1)]	OFF	OFF	ON	OFF	OFF
5	ON	OFF	ON	OFF	OFF
6	OFF	ON	ON	OFF	OFF
7	ON	ON	ON	OFF	OFF
8	OFF	OFF	OFF	ON	OFF
9	ON	OFF	OFF	ON	OFF
10	OFF	ON	OFF	ON	OFF
11	ON	ON	OFF	ON	OFF
12	OFF	OFF	ON	ON	OFF
13	ON	OFF	ON	ON	OFF
14	OFF	ON	ON	ON	OFF
15	ON	ON	ON	ON	OFF
16	OFF	OFF	OFF	OFF	ON
17	ON	OFF	OFF	OFF	ON
18	OFF	ON	OFF	OFF	ON
19	ON	ON	OFF	OFF	ON
20	OFF	OFF	ON	OFF	ON
21	ON	OFF	ON	OFF	ON
22	OFF	ON	ON	OFF	ON
23	ON	ON	ON	OFF	ON
24	OFF	OFF	OFF	ON	ON
25	ON	OFF	OFF	ON	ON
26	OFF	ON	OFF	ON	ON
27	ON	ON	OFF	ON	ON
28	OFF	OFF	ON	ON	ON
29	ON	OFF	ON	ON	ON
30	OFF	ON	ON	ON	ON
31	–	–	–	–	–

[1)] Factory setup

If you use a CP32 communications processor, the address can be set up using the AB22A and the command ADS.

Talker/Listener setting:

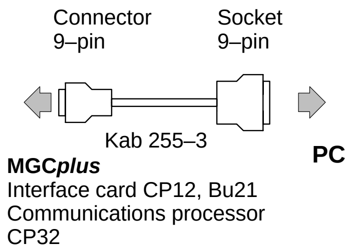
Switch	not allowed	only Listener	only Talker	Addressable [1]
LO	OFF	ON	OFF	ON
TO	OFF	OFF	ON	ON

[1] Factory setting

6 Connect computer and device

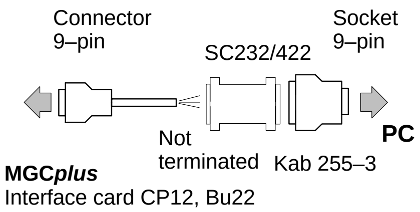
Connect the MGC*plus* and computer to the mains supply

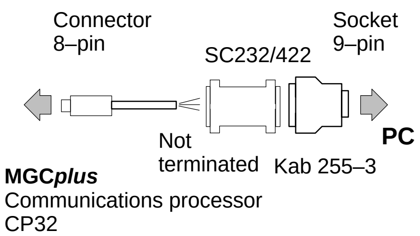
Leave the MGC*plus* and computer switched off at first



RS232C interface:

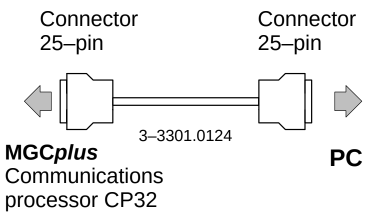
- Use the cable Kab 255-3 to connect the MGC*plus* to the computer (cable Kab 255-3 is included among the components supplied)
- Set baud rate (9600)
- Set parity (ON)
- Switch on MGC*plus* and computer





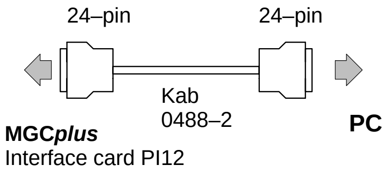
RS485 interface:

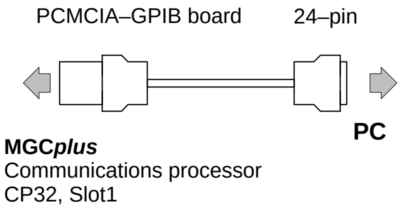
- Connect a cable with 9–pin connector (CP12), with 8–pin connector (CP32), unterminated, to the MGC*plus* and connect to the Serial Converter SC232/422; connect these to the computer directly or with the inserted adapter plug Kab 255-3 (the unterminated cable and the Serial Converter are not included in the list of components supplied)
- Set baud rate (9600)
- Set parity (ON)
- Set up address
  1.MGC*plus*: bus address 1
  2.MGC*plus*: bus address 2 etc.
- Switch on MGC*plus*, computer and converter power pack



Printer interface:

- Using the cable (3–3301.0124) included among the components supplied with the CP32, connect the MGC*plus* and the computer
- Switch on MGC*plus* and computer





IEEE interface 488:

- Using the cable Kab 0488-2 or the PCMCIA–GPIB board cable, connect the MGC*plus* to the computer (the cable Kab 0488-2 and the board are not included with the components supplied)

Caution: Do in any case mount the supplied folding ferrites onto the connection cable!

- Set up address:
  - 1.MGC*plus*: bus address 4
  - 2.MGC*plus*: bus address 5 etc.
- Switch on MGC*plus* and computer

The computer and MGC*plus* are now ready to take measurements.
(Please take note of the chapter called ’Connections’ in the Operating Manual).

Note:

Assignment **SLOT 0**: Hard disk

Assignment **SLOT 1**: IEEE488 interface

G Program creation

1 IBM PC and compatibles

In order to make getting started with program creation a little easier for you, here are the routines for programming the interfaces of IBM and compatible PCs.

CAUTION

- Firstly, make sure all the cables are connected
- Only when you have done this should you switch on the devices involved

1.1 Communication via the RS232C interface

- Connect the PC to the MGC*plus* with the Kab255-3 cable.
- On the AB22A/AB32, set the baud rate of the RS232 interface to 9600Baud and the format to 8E1 (8 data bits, even parity, 1 stop bit).

Test program (Quick-BASIC 4.5)
Ext. triggering of a measurement run with 3 measured value lines, one measured value line pre–trigger and a total of 4 measurements. The gross signal of channels 3 and 4 and the time channel are measured and scaled.

```
DECLARE SUB WriteLine (w$)
DECLARE SUB ReadOMP ()
DECLARE SUB ReadRMB ()
DECLARE SUB ReadESR ()
DIM SHARED rd$, wert&, anzsig%, anzzln%, anzwdh%
einheit1$ = ”EUN” + CHR$(34) + ”%   ” + CHR$(34)           ’physic. unit %
einheit2$ = ”IUN” + CHR$(34) + ”mV/V” + CHR$(34)           ’Display unitmV/V
einheit3$ = ”IUN” + CHR$(34) + ”Ohm ” + CHR$(34)           ’Display unit ohm
init:
OPEN ”COM1:9600,N,8,1,CS,DS,CD,LF,RB256” FOR RANDOM AS #1  ’Set up interface
OUT &H3FB, &H1B                                            ’8 data bits, 1 stop bit, even parity
PRINT #1, CHR$(18)                                         ’Call HBM Interpreter
PRINT #1, ”STP”                                            ’Terminate current measurement run
WriteLine (”SRB1”)                                         ’Switch on acknowledgement (default)
WriteLine (”PCS3”)                                         ’Select Channel 3 (e.g. ML10)
WriteLine (”SAD14,358”)                                    ’Ub=5V,SG full bridge low level
WriteLine (”GFV0”)                                         ’no K–Factor, basic unit mV/V
WriteLine (einheit1$)                                      ’Physical unit %
WriteLine (”CAP1,0,0”)                                     ’Input characteristic point 1: 0mV/V=0%
```

```
WriteLine (”CAP2,2,100”)          ’Input characteristic point 2: 2mV/V = 100%
WriteLine (”OCP1,0,0”)            ’Output characteristic point 1: 0% = 0V
WriteLine (”OCP2,100,10”)         ’Output characteristic point 2: 100% = 10V
WriteLine (”CAV0,100”)            ’Full scale value at 100%
WriteLine (”CAP0”)                ’Calibrate
WriteLine (einheit2$)             ’Display unit mV/V
WriteLine (”SFC142,955”)          ’Cut–off frequency 100 Hz,Bessel
WriteLine (”PCS4”)                ’Select Channel 4 (e.g. ML35)
WriteLine (”SAD26,477”)           ’4–line circuit, resistance 5000 ohm
WriteLine (einheit1$)             ’physic. unit %
WriteLine (”CAP1,0,0”)            ’Input characteristic point 1: 0 ohm = 0%
WriteLine (”CAP2,5000,100”)       ’Input characteristic point 2: 5000 ohm = 100%
WriteLine (”OCP1,0,0”)            ’Output characteristic point 1: 0% = 0V
WriteLine (”OCP2,100,10”)         ’Output characteristic point 2: 100% = 10V
WriteLine (”CAV0,100”)            ’Full scale value at 100%
WriteLine (”CAP0”)                ’Calibrate
WriteLine (einheit3$)             ’Display unit ohm
WriteLine (”SFC142,921”)          ’Cut–off frequency 0.5 Hz, Bessel
WriteLine (”PCS3,4”)              ’Select channels 3 and 4
WriteLine (”AIS42”)               ’Apply measuring signal at amplifier input
WriteLine (”ICS3,110”)            ’3 decimal places, step 1
WriteLine (”SIS3,202”)            ’Display gross value of Channel 3 on AB22A
WriteLine (”CAL”)                 ’Calibrate active channels
WriteLine (”MCS3,4,17”)           ’Record channels 3, 4 and time channel
WriteLine (”MSS214”)              ’Record gross signal
WriteLine (”ICR6317”)             ’Sampling rate 2400 Hz (default)
WriteLine (”MBF1253”)             ’Measurement output format (binary, 4 byte, LSBxxxxMSB)
WriteLine (”TRR”)                 ’Remove trigger conditions
```

```
WriteLine ("CAP2,2,100")          'Input characteristic point 2: 2mV/V = 100%
WriteLine ("TRE1,3,214,6102,6005") 'Start trigger event 1, channel 3, gross signal,
                                   Ext. trigger, stat. negative
WriteLine ("TRD6500,151")          'Start trigger machine, trigger mode AND
WriteLine ("TSV3,1,4")             'Start measurement (3 measured value lines,
                                   1 measured value line pre–trigger, 4 measurements in total)
anzsig% = 3                        '3 signals (gross[3], gross[4], time channel)
anzzln% = 3                        '3 measured value lines (p1 of TSV command)
anzwdh% = 4                        '4 repetitions (p3 of TSV command)
ReadOMP                            'Measurement ended ?
ReadRMB                            'Read in measured values
end:
PRINT #1, "DCL"                    'Exit HBM Interpreter
END

SUB ReadESR                        'Error routine
  PRINT #1, "*ESR?"                'Read off error status register
  LINE INPUT #1, rd$               'Read in string incl. CR
  z$ = INPUT$(1, #1)               'Input LF
END SUB
SUB ReadOMP                        'Measurement ended ?
  PRINT "Wait for trigger!"
  PRINT
```

```
DO
  PRINT #1, "OMP?"                      'Reading pointer status ?
  LINE INPUT #1, rd$                    'Read in string incl. CR
  z$ = INPUT$(1, #1)                    'Input LF
  IF rd$ = "?" THEN                     'Error ?
    ReadESR                             'Call error routine
    PRINT : PRINT "Error: "; rd$, "OMP? command"
    STOP                                'Exit program
  END IF
  n% = 0
  FOR i = 1 TO 4                        'Extract number of measured value lines
    n% = INSTR(n% + 1, rd$, ",")
    IF i = 3 THEN m% = n%
  NEXT i
  n% = VAL(MID$(rd$, m% + 1, n% – m% – 1))
LOOP UNTIL n% = anzzln% * anzwdh%       'Number of measured value lines * repetitions achieved?
END SUB

SUB ReadRMB                             'Read in measured values
  rmbbef$ = "RMB?" + STR$(anzzln% * anzwdh%) + ",6409"
  PRINT #1, rmbbef$                     'Read off measured values (measured value lines *
repetitions)
  z$ = INPUT$(1, #1)                    '# or read in ?
  IF z$ = "?" THEN                      'Error ?
    z$ = INPUT$(2, #1)                  'Read in CR and LF
    ReadESR                             'Call error routine
    PRINT : PRINT "Error: "; rd$, "RMB? command
    STOP                                'Exit program
 END IF
```

```
z$ = INPUT$(1, #1)                                   'Read in 0
PRINT "  Channel 3 Channel 4 Time channel"           'Header
FOR i = 1 TO anzwdh%                                 'Number of repetitions
FOR j = 1 TO anzzln%                                 'Number of measured value lines
FOR k = 1 TO anzsig%                                 'Total signals of all channels
  z$ = INPUT$(1, #1) + CHR$(0)                       'Read in measured value status
  p% = CVI(z$)
  rd$ = CHR$(0) + INPUT$(3, #1)                      'Read in measured value
  IF k = 1 THEN                                      'Channel 4
  value! = ((CVL(rd$) / 256) * 2) / 7680000          'mV/V
  PRINT USING "+##.###"; value!;
  PRINT " mV/V   ";
ELSEIF k = 2 THEN                                    'Channel 4
    value! = ((CVL(rd$) / 256) * 5000) / 7680000     'Ohm
    PRINT USING "###.###"; value!;
    PRINT " Ohm    ";
ELSE                                                 'Time channel
    value! = ((CVL(rd$) / 256) * 416.7) / 1000000    's (1/2400Hz = 416.7µs)
    PRINT USING "##.##"; value!;
    PRINT " s";
  END IF
NEXT k
  PRINT
NEXT j
  PRINT
NEXT i
  GET #1, , p%                                       'Read in 2 bytes (CRLF)
```

```
END SUB
SUB WriteLine (w$)                              'Write routine
  PRINT #1, w$                                  'Output command
  LINE INPUT #1, rd$                            'Read in response incl. CR
  z$ = INPUT$(1, #1)                            'Input LF
  IF rd$ <> "0" THEN                            'Error ?
    ReadESR                                     'Call error routine
    PRINT : PRINT "Error: "; rd$, w$; "command"
    STOP                                        'Exit program
  END IF
END SUB
```

Explanations:

a) The instruction OUT &H3FB, &H1B in the initialisation block is used for setting the interface to even parity.

b) With the WriteLine () commands, a set–up command is sent to the MGC*plus* and the response to the command read in. In the event of an error message – see *ESR? command – this is output and the program stopped.

c) With the commands WriteLine (”PCS3”) to WriteLine(”CAL”), channels 3 and 4 are parameterised and calibrated with the command WriteLine (”CAL”).

d) With the commands WriteLine (”MCS3,4,17”) to WriteLine (”TRD6500,151”), the actual measurement is predefined and started with WriteLine (”TSV3,1,4”).

e) The measurement that has been started and is awaiting a trigger event can be seen as a measuring device symbol on the AB22A. In the program this message is displayed: Wait for trigger!

f) By connecting Remote 7 (Pin 8) to the earth (Pin 1) on socket Bu2 – e.g. on the AP01 connection board – the measurement is triggered.

g) If the program runs with no errors, the following display appears on the screen, depending on the channel mismatch:

Channel 3	**Channel 4**	**Time channel**	
+2.000 mV/V	350.342Ohm	5.64s	Measured value line 1 (pre–trigger)
+2.001 mV/V	350.341Ohm	5.64s	Measured value line 2 (post–trigger)
+2.000 mV/V	350.340Ohm	5.64s	Measured value line 3 (post–trigger)
+1.999 mV/V	350.341Ohm	5.64s	2nd measurement
+2.000 mV/V	350.340Ohm	5.64s	
+2.000 mV/V	350.341Ohm	5.64s	
+2.000 mV/V	350.342Ohm	5.65s	3rd measurement
+2.000 mV/V	350.341Ohm	5.65s	
+2.000 mV/V	350.342Ohm	5.65s	
+2.001 mV/V	350.340Ohm	5.65s	4th measurement
+2.001 mV/V	350.341Ohm	5.65s	
+2.000 mV/V	350.340Ohm	5.65s	

1.2 Communication via the RS485 interface

The following test program shows a measurement session (threshold value triggering) using 2 MGC*plus* devices. Both devices must be equipped with a CP32 communications processor.

- Connect the devices with a Kab261–2 synchronisation cable. The first device must be switched to ”Master” and the second one to ”Slave”.
- Using the synchronisation cable, connect the SYNCHR.OUT socket on the 1st device (master) to the SYNCHR.IN socket on the 2nd device (slave).
- Connect the PC to the MGC*plus* devices – according to Chapter 6 ”Connecting computer and device”. The device addresses, baud rate and data format are to be set on the AB22A/AB32 in set–up mode (System, Interface, RS485...).

Test program: (Quick-BASIC 4.5)

Threshold value triggering of a measurement with 3 measured value lines, one measured value line pre–trigger and a total of 2 measurements. The gross signal of channels 3 and 4 of device no. 1 (master), the net signal of channels 2 and 3 of device no. 2 (slave) and the time channels are measured and scaled.

```
DECLARE SUB WriteLine (w$)
DECLARE SUB ReadOMP ()
DECLARE SUB ReadRMB1 ()
DECLARE SUB ReadRMB2 ()
DECLARE SUB ReadESR ()
DIM SHARED rd$, wert&, anzsig%, anzzln%, anzwdh%
einheit1$ = ”EUN” + CHR$(34) + ”%   ” + CHR$(34)          ’physic. unit %
einheit2$ = ”EUN” + CHR$(34) + ”µm/m” + CHR$(34)          ’physic.unit µm/m
einheit3$ = ”IUN” + CHR$(34) + ”µm/m” + CHR$(34)          ’Display unit µm/m
einheit4$ = ”IUN” + CHR$(34) + ”mV/V” + CHR$(34)          ’Display unit mV/V
init:
OPEN ”COM1:9600,N,8,1,CS,DS,CD,LF,RB256” FOR RANDOM AS #1 Set up interface
OUT &H3FB, &H1B                                           ’8 data bits, 1 stop bit, even parity
PRINT #1, CHR$(18);                                       ’Call HBM Interpreter
PRINT #1, ”S33”                  ’subsequent commands go to both devices, but only device 1 responds.
```

```
PRINT #1, "STP"                 'Terminate current measurement run
WriteLine ("SRB1")              'Switch on acknowledgement (default)
WriteLine ("S02")               'Get response from Device 2
device1:
PRINT #1, "S01"                 'Select Device 1
WriteLine ("PCS3")              'Select Channel 3 (e.g. ML10)
WriteLine ("SAD14,358")         'Ub=5V,SG full bridge low level
WriteLine ("GFV0")              'no K–Factor, basic unit mV/V
WriteLine (einheit1$)           'physic. unit %
WriteLine ("CAP1,0,0")          'Input characteristic point 1: 0mV/V = 0%
WriteLine ("CAP2,2,100")        'Input characteristic point 2: 2mV/V = 100%
WriteLine ("OCP1,0,0")          'Output characteristic point 1: 0% = 0V
WriteLine ("OCP2,100,10")       'Output characteristic point 2: 100% = 10V
WriteLine ("CAV0,100")          'Full scale value at 100%
WriteLine ("CAP0")              'Calibrate
WriteLine (einheit4$)           'Display unit mV/V
WriteLine ("SFC142,955")        'Cut–off frequency 100 Hz,Bessel
WriteLine ("PCS4")              'Select Channel 4 (e.g. ML35)
WriteLine ("SAD26,477")         '4–line circuit, resistance 5000 ohm
WriteLine (einheit1$)           'physic. unit %
WriteLine ("CAP1,0,0")          'Input characteristic point 1: 0 ohm = 0%
WriteLine ("CAP2,5000,100")     'Input characteristic point 2: 5000 ohm = 100%
WriteLine ("OCP1,0,0")          'Input output characteristic point 1: 0% = 0V
WriteLine ("OCP2,100,10")       'Output characteristic point 2: 100% = 10V
WriteLine ("CAV0,100")          'Full scale value at 100%
WriteLine ("CAP0")              'Calibrate
WriteLine (einheit3$)           'Display unit ohm
WriteLine ("SFC142,921")        'Cut–off frequency 0.5 Hz, Bessel
```

```
WriteLine ("PCS3,4")          'Select channels 3 and 4
WriteLine ("AIS42")           'Apply measuring signal at amplifier input
```

WriteLine (”ICS3,110”)	’3 decimal places, step 1
WriteLine (”SIS3,202”)	’Display gross value of Channel 3 on AB22A
WriteLine (”CAL”)	’Calibrate active channels
WriteLine (”MCS3,4,17”)	’Record channels 3, 4 and time channel
WriteLine (”MSS214”)	’Record gross signal
WriteLine (”ICR6317”)	’Sampling rate 2400 Hz (default)
WriteLine (”MBF1253”)	’Measurement output format (binary, 4 byte, LSBxxxxMSB)
WriteLine (”TRR”)	’Remove trigger conditions
WriteLine (”TRE1,3,214,6006,6004,1.0”)	’Start trigger event 1, Channel 3, gross signal, measurement threshold, stat. positive, threshold value 1.0
WriteLine (”TRD6500,151”)	’Start trigger machine, trigger mode AND
device2:	
PRINT #1, ”S02”	’Select device 2
WriteLine (”PCS2”)	’Select Channel 2 (e.g. ML55)
WriteLine (”SAD14,353”)	’Ub=5V,SG full bridge
WriteLine (”GFV2”)	’k–factor=2, basic unit µm/m
WriteLine (einheit1$)	’physic. unit %
WriteLine (”CAP1,0,0”)	’Input characteristic point 1: 0µm/m = 0%
WriteLine (”CAP2,4000,100”)	’Input characteristic point 2: 4000µm/m = 100%
WriteLine (”OCP1,0,0”)	’Output characteristic point 1: 0% = 0V
WriteLine (”OCP2,100,10”)	’Output characteristic point 2: 100% = 10V
WriteLine (”CAV0,100”)	’Full scale value at 100%
WriteLine (”CAP0”)	’Calibrate
WriteLine (einheit3$)	’Display unit µm/m
WriteLine (”SFC142,955”)	’Cut–off frequency 100 Hz,Bessel
WriteLine (”PCS3”)	’Select Channel 3 (e.g. ML55)
WriteLine (”SAD14,354”)	’Ub=5V,SG half bridge
WriteLine (”GFV0”)	’no K–Factor, basic unit mV/V
WriteLine (einheit1$)	’physic. unit %

```
WriteLine (”CAP1,0,0”)            ’Input characteristic point 1: 0mV/V = 0%
WriteLine (”CAP2,2,100”)          ’Input characteristic point 2: 2mV/V = 100%
WriteLine (”OCP1,0,0”)            ’Output characteristic point 1: 0% = 0V
WriteLine (”OCP2,100,10”)         ’Output characteristic point 2: 100% = 10V
WriteLine (”CAV0,100”)            ’Full scale value at 100%
WriteLine (”CAP0”)                ’Calibrate
WriteLine (einheit4$)             ’Display unit mV/V
WriteLine (”SFC142,955”)          ’Cut–off frequency 100 Hz,Bessel
WriteLine (”PCS2,3”)              ’Select channels 2 and 3
WriteLine (”AIS42”)               ’Apply measuring signal at amplifier input
WriteLine (”ICS1,110”)            ’3 decimal places, step 1
WriteLine (”SIS2,203”)            ’Display net value of Channel 2 on AB22A
WriteLine (”CAL”)                 ’Calibrate active channels
WriteLine (”MCS2,3,17”)           ’Record channels 2, 3 and time channel
WriteLine (”MSS215”)              ’Record net signal
WriteLine (”ICR6317”)             ’Sampling rate 2400 Hz (default)
WriteLine (”MBF1253”)             ’Measurement output format (binary, 4 byte, LSBxxxxMSB)
PRINT #1, ”S02”                   ’Select device 2, trigger slave must be started first
WriteLine (”TSV3,1,2”)            ’Start measurement (3 measured value lines, 1 measured value
                                  line pre–trigger, a total of 2 measurements)
PRINT #1, ”S01”                   ’Select device 1, trigger master
WriteLine (”TSV3,1,2”)            ’Start measurement (3 measured value lines, 1 measured value
                                  line pre–trigger, a total of 2 measurements)
anzsig% = 3                       ’3 signals channel X, channel Y, time channel)
anzzln% = 3                       ’3 measured value lines (p1 of TSV command)
anzwdh% = 2                       ’2 repetitions (p3 of TSV command)
ReadOMP                           ’Measurement ended ?
ReadRMB1                          ’Read in measured values of device 1
PRINT #1, ”S02”                   ’Select device 2
```

```
ReadRMB2                    'Read in measured values of device 2
```

```
end:
PRINT #1, "S98"                    'subsequent command goes to both devices, but neither responds
PRINT #1, "DCL"                    'Exit HBM Interpreter
END

SUB ReadESR                        'Error routine
  PRINT #1, "*ESR?"                'Read off error status register
  LINE INPUT #1, rd$               'Read in string incl. CR
  z$ = INPUT$(1, #1)               'Input LF
END SUB
SUB ReadOMP
  PRINT "Wait for trigger!"
  PRINT
DO
  PRINT #1, "OMP?"                 'Reading pointer status ?
  LINE INPUT #1, rd$               'Read in string incl. CR
  z$ = INPUT$(1, #1)               'Input LF
  IF rd$ = "?" THEN                'Error ?
    ReadESR                        'Call error routine
    PRINT : PRINT "Error: "; rd$, "OMP? command"
    STOP                           'Exit program
  END IF
  n% = 0
  FOR i = 1 TO 4                   'Extract number of measured value lines
    n% = INSTR(n% + 1, rd$, ",")
    IF i = 3 THEN m% = n%
  NEXT i
  n% = VAL(MID$(rd$, m% + 1, n% – m% – 1))
LOOP UNTIL n% = anzzln% * anzwdh%  'Number of measured value lines * repetitions achieved?
```

```
END SUB
SUB ReadRMB1                                        'Read in measured values of device 1
  rmbbef$ = "RMB?" + STR$(anzzln% * anzwdh%) + ",6409"
  PRINT #1, rmbbef$                                 'Read off measured values (lines * repetitions)
  z$ = INPUT$(1, #1)                                '# or read in ?
  IF z$ = "?" THEN                                  'Error ?
    z$ = INPUT$(2, #1)                              'Read in CR and LF
    ReadESR                                         'Call error routine
    PRINT : PRINT "Error: "; rd$, "RMB? command
    STOP                                            'Exit program
  END IF
  z$ = INPUT$(1, #1)                                'Read in 0
  PRINT "  Device 1:                                "          '
  PRINT "  Channel 3    Channel 4    Time channel"  'Header

FOR i = 1 TO anzwdh%                                'Number of repetitions
FOR j = 1 TO anzzln%                                'Number of measured value lines
FOR k = 1 TO anzsig%                                'Total signals of all channels
  z$ = INPUT$(1, #1) + CHR$(0)                      'Read in measured value status
  p% = CVI(z$)
  rd$ = CHR$(0) + INPUT$(3, #1)                     'Read in measured value
  IF k = 1 THEN                                     'Channel 4
    value! = ((CVL(rd$) / 256) * 2) / 7680000       'mV/V
    PRINT USING "+##.###"; value!;
    PRINT " mV/V   ";
  ELSEIF K = 2 THEN                                 'Channel 4
    value! = ((CVL(rd$) / 256) * 5000) / 7680000    'Ohm
    PRINT USING "###.###"; value!;
    PRINT " Ohm    ";
```

```
  ELSE                                         'Time channel
    value! = ((CVL(rd$) / 256) * 416.7) / 1000000   's (1/2400Hz = 416.7µs)
    PRINT USING "##.###"; value!;
    PRINT " s";
  END IF
NEXT k
  PRINT
NEXT j
  PRINT
NEXT i
  GET #1, , p%                                 'Read in 2 bytes (CRLF)
END SUB
SUB ReadRMB2                                   'Read in measured values of device 2
  rmbbef$ = "RMB?" + STR$(anzzln% * anzwdh%) + ",6409"
  PRINT #1, rmbbef$                            'Read off measured values (lines * repetitions)
  z$ = INPUT$(1, #1)                           '# or read in ?
  IF z$ = "?" THEN                             'Error ?
    z$ = INPUT$(2, #1)                         'Read in CR and LF
    ReadESR                                    'Call error routine
    PRINT : PRINT "Error: "; rd$, "RMB? command
    STOP                                       'Exit program
  END IF
  z$ = INPUT$(1, #1)                           'Read in 0
  PRINT "  Device 2:"
  PRINT "  Channel 2 Channel 3 Time channel"   'Header
FOR i = 1 TO anzwdh%                           'Number of repetitions
FOR j = 1 TO anzzln%                           'Number of measured value lines
FOR k = 1 TO anzsig%                           'Total signals of all channels
```

```
  z$ = INPUT$(1, #1) + CHR$(0)                          'Read in measured value status
  p% = CVI(z$)
  rd$ = CHR$(0) + INPUT$(3, #1)                         'Read in measured value
  IF k = 1 THEN                                         'Channel 3
    value! = ((CVL(rd$) / 256) * 2 / 2 * 4000) / 7680000  'µm/m (2/2 = 2/k–factor)
    PRINT USING "+####.#"; value!;
    PRINT " um/m   ";
  ELSEIF K = 2 THEN                                     'Channel 4
    value! = ((CVL(rd$) / 256) * 2) / 7680000           'mV/V
    PRINT USING "+##.#"; value!;
    PRINT " mV/V   ";
  ELSE                                                  'Time channel
    value! = ((CVL(rd$) / 256) * 416.7) / 1000000       's (1/2400Hz = 416.7µs)
    PRINT USING "##.###"; value!;
    PRINT " s";
  END IF
NEXT k
  PRINT
NEXT j
  PRINT
NEXT i
  GET #1, , p%                                          'Read in 2 bytes (CRLF)
END SUB
```

```
SUB WriteLine (w$)                                  'Write routine
  PRINT #1, w$                                      'Output command
  LINE INPUT #1, rd$                                'Read in response incl. CR
  z$ = INPUT$(1, #1)                                'Input LF
  IF rd$ <> "0" THEN                                'Error ?
    ReadESR                                         'Call error routine
    PRINT : PRINT "Error: "; rd$, w$; "command"
    STOP                                            'Exit program
  END IF
END SUB
```

Explanation:

a) The instruction OUT &H3FB, &H1B in the initialisation block is used for setting the interface to even parity.

b) Both devices are prepared to receive subsequent commands with the command PRINT #1,"S33" with device 1 responding on behalf of both devices. With WriteLine ("S02") the response of device 2 must be retrieved.

c) Device 1 is selected by using the command PRINT #1,"S01", with the WriteLine() commands a set–up command is sent to the MGC*plus* and the response to the command read in. In the event of an error message – see *ESR? command – this is output and the program stopped.

d) With the commands WriteLine ("PCS3") to WriteLine("CAL"), channels 3 and 4 of device 1 are parameterised and calibrated with the latter.

e) With the commands WriteLine ("MCS3,4,17") to WriteLine("TRD6500,151") a measurement run with device 1 is predefined.

f) With the command PRINT #1,"S02" device 2 is selected. Further sequence of operations as under c)

g) With the commands WriteLine ("PCS2") to WriteLine("CAL"), channels 2 and 3 of device 2 are parameterised and calibrated with the latter.

h) With the commands WriteLine (”MCS2,3,17”) to WriteLine(”MBF1253”) the measurement run with device 2 is predefined.

i) The commands PRINT #1,”S02” and WriteLine (”TSV3,1,2”) or PRINT #1,”S01” and’ WriteLine (”TSV3,1,2”), the measurement run finally starts, whereby device 2 (slave) must be started first.

j) The measurement run which has been started and is awaiting a trigger event can be seen as a measuring device symbol on the AB22A screen. In the program this message is displayed: Wait for trigger!

k) The measurement is triggered if the measured value of channel 3 (device 1) exceeds the threshold of 1,000 mV/V.

l) If the program runs with no errors, the following display appears on the screen, depending on the channel mismatch:

Device 1:

Channel 3	**Channel 4**	**Time channel**	
+2.000mV/V	350.342Ohm	5.641s	’Measured value line 1 (pre–trigger)
+2.001mV/V	350.341Ohm	5.641s	’Measured value line 2 (post–trigger)
+2.000mV/V	350.340Ohm	5.642s	’Measured value line 3 (post–trigger)
+1.999mV/V	350.341Ohm	5.642s	’2nd measurement
+2.000mV/V	350.340Ohm	5.642	
+2.000mV/V	350.341Ohm	5.643s	

Device 2:

Channel 2	**Channel 3**	**Time channel**	
+1728.0µm/m	–1.1mV/V	5.721s	’Measured value line 1 (pre–trigger)
+1728.1µm/m	–1.0mV/V	5.721s	’Measured value line 2 (pre–trigger)
+1728.0µm/m	–1.1mV/V	5.722s	’Measured value line 3 (pre–trigger)
+1728.1µm/m	–1.1mV/V	5.722s	’2nd measurement
+1728.1µm/m	–1.1mV/V	5.723s	
+1728.0µm/m	–1.0mV/V	5.723s	

1.3 Communication via the IEEE interface

The following test program shows a measurement session (threshold value triggering) using 2 MGC*plus* devices. Both devices must be equipped with a CP32 communications processor.

- Connect the devices with a Kab261–2 synchronisation cable. The first device must be switched to ”Master” and the second one to ”Slave”.
- Using the synchronisation cable, connect the SYNCHR.OUT socket on the 1st device (master) to the SYNCHR.IN socket on the 2nd device (slave).

Also necessary for each device is a PCMCIA–GPIB board from the ’National Instruments’ company (not part of standard supply).

- Plug the PCMCIA–GPIB board into **Slot1** on the CP32.
- Connect the MGCplus devices and the PC with the GPIB connecting cables and then in set–up mode of the AB22A/AB32 (System, Interface, IEC...) set up the device addresses for both devices.

  Device 1: Address 4

  Device 2: Address 5

Setting up the GPIB board in the PC with National Instruments GPIB-PC software to:

GPIB0

Primary GPIB Address	0 (other addresses are also possible)
Secondary GPIB Address	none
Timeout setting	T10μs
EOS-byte	00H (other bytes also possible)
Terminate Read on EOS	no
Set EOI with EOS on write	no
Type of compare on EOS	7 bit
Set EOI w/last byte of write	yes
GPIB-PC Model	PC2A (or as appropriate)
Board is System Controller	yes
Local Lockout on all devices	no (yes is also an option)
Disable Auto Serial Polling	yes
Disable Device Unaddressing	yes
High-speed timing	yes (no is also an option)
Interrupt jumper setting	none
Base I/O Address	02E1H (or as appropriate)
DMA channel	none
Internal Clock Freq (in MHz)	8

The following set–up is to be made for HBM devices:

	DEV4
Primary GPIB address:	04H
Secondary GPIB address:	none
Timeout setting:	T10s
EOS byte:	00H
Terminate Read on EOS	no
Set EOI with EOS on write	no
Type of compare on EOS	7-bit
Set EOI w/last byte Write	no

	DEV5
Primary GPIB-Address:	05H
Secondary GPIB-Address:	none
Timeout setting:	T10s
EOS byte:	00H
Terminate Read on EOS	no
Set EOI with EOS on write	no
Type of compare on EOS	7-bit
Set EOI w/last byte Write	no

More devices can be defined as required, as long as they are given another primary address. After these settings have been saved on exiting from IBCONF[1)], a warm start must be effected on the computer in order to activate the settings.

Note:

Other versions of the GPIB-PC software may possibly have another sequence for the parameters to be entered or questions that are phrased differently. In that case, the settings should be undertaken in a similar manner.

1) Configuration program for the GPIB board

Test program: (Quick-BASIC 4.5)

Threshold value triggering of a measurement with 3 measured value lines, one measured value line pre–trigger and a total of 2 measurements. The gross signal of channels 3 and 4 of device no. 1 (master), the net signal of channels 2 and 3 of device no. 2 (slave) and the time channels are measured and scaled.

```
$INCLUDE: 'c:\at–gpib\qbasic\qbdecl4.bas'
DECLARE SUB IFinit ()
DECLARE SUB WriteLine (WR$, n%)
DECLARE SUB Load ()
DECLARE SUB ReadOMP ()
DECLARE SUB ReadRMB1 ()
DECLARE SUB ReadRMB2 ()
DECLARE SUB ReadESR ()
DIM SHARED i, j, BD%, dev%(2), n%, st%, anzsig%, anzzln%, anzwdh%
DIM SHARED RD$, WR$
einheit1$ = "EUN" + CHR$(34) + "%   " + CHR$(34) + ";"        'physic. unit %
einheit2$ = "EUN" + CHR$(34) + "um/m" + CHR$(34) + ";"        'physic.unit µm/m
einheit3$ = "IUN" + CHR$(34) + "um/m" + CHR$(34) + ";"        'Display unit µm/m
einheit4$ = "IUN" + CHR$(34) + "mV/V" + CHR$(34) + ";"        'Display unit mV/V
DATA DEV4,DEV5
CLS                                                           'Clear screen
init:
  CALL IFinit
  IF j >= 0 THEN
    FOR i = 1 TO 2
      st% = ILWRT(dev%(i), "STP;", 4)                         'Terminate current measurement run
      CALL WriteLine("SRB1;", 5)                              'Switch on acknowledgement
    NEXT i
```

```
device1: i = 1
  CALL WriteLine(”PCS3;”, 5)                 ’Select Channel 3 (e.g. ML10)
  CALL WriteLine(”SAD14,358;”, 10)           ’Ub=5V,SG full bridge low level
  CALL WriteLine(”GFV0;”, 5)                 ’no K–Factor, basic unit mV/V
  CALL WriteLine(einheit1$, 10)              ’physic. unit %
  CALL WriteLine(”CAP1,0,0;”, 9)             ’Input characteristic point 1: 0mV/V = 0%
  CALL WriteLine(”CAP2,2,100;”, 11)          ’Input characteristic point 2: 2mV/V = 100%
  CALL WriteLine(”OCP1,0,0;”, 9)             ’Output characteristic point 1: 0% = 0V
  CALL WriteLine(”OCP2,100,10;”, 12)         ’Output characteristic point 2: 100% = 10V
  CALL WriteLine(”CAP0;”, 5)                 ’Calibrate
  CALL WriteLine(einheit4$, 10)              ’Display unit mV/V
  CALL WriteLine(”SFC142,955;”, 11)          ’Cut–off frequency 100 Hz,Bessel

  CALL WriteLine(”PCS4;”, 5)                 ’Select Channel 4 (e.g. ML35)
  CALL WriteLine(”SAD26,477;”, 10)           ’4–line circuit, resistance 5000 ohm
  CALL WriteLine(einheit1$, 10)              ’physic. unit %
  CALL WriteLine(”CAP1,0,0;”, 9)             ’Input characteristic point 1: 0 ohm = 0%
  CALL WriteLine(”CAP2,5000,100;”, 14)       ’Input characteristic point 2: 5000 ohm = 100%
  CALL WriteLine(”OCP1,0,0;”, 9)             ’Output characteristic point 1: 0% = 0V
  CALL WriteLine(”OCP2,100,10;”, 12)         ’Output characteristic point 2: 100% = 10V
  CALL WriteLine(”CAP0;”, 5)                 ’Calibrate
  CALL WriteLine(einheit3$, 10)              ’Display unit ohm
  CALL WriteLine(”SFC142,921;”, 11)          ’Cut–off frequency 0.5 Hz, Bessel

  CALL WriteLine(”PCS3,4;”, 7)               ’Select channels 3 and 4
  CALL WriteLine(”AIS42;”, 6)                ’Apply measuring signal at amplifier input
  CALL WriteLine(”ICS3,110;”, 9)             ’3 decimal places, step 1
  CALL WriteLine(”SIS3,202;”, 9)             ’Display gross value of Channel 3 on AB22A
  CALL WriteLine(”CAL;”, 4)                  ’Calibrate active channels
```

```
  CALL WriteLine("MCS3,4,17;", 10)              'Record channels 3, 4 and time channel
  CALL WriteLine("MSS214;", 7)                  'Record gross signal
  CALL WriteLine("ICR6317;", 8)                 'Sampling rate 2400 Hz (default)
  CALL WriteLine("MBF1253;", 8)                 'Measurement output format (binary, 4 byte, LSBxxxxMSB)
  CALL WriteLine("TRR;", 4)                     'Remove trigger conditions
  CALL WriteLine("TRE1,3,214,6006,6004,1.0;", 25) 'Start trigger event 1, Channel 3, gross signal
                                                'Measurement threshold, Stat. positive, threshold value 1.0
  CALL WriteLine("TRD6500,151;", 12)            'Start trigger machine, trigger mode AND

device2: i = 2
  CALL WriteLine("PCS2;", 5)                    'Select Channel 2 (e.g. ML55)
  CALL WriteLine("SAD14,353;", 10)              'Ub=5V,SG full bridge
  CALL WriteLine("GFV2;", 5)                    'K factor=2, basic unit µm/m
  CALL WriteLine(einheit1$, 10)                 'physic. unit %
  CALL WriteLine("CAP1,0,0;", 9)                'Input characteristic point 1: 0µm/m = 0%
  CALL WriteLine("CAP2,4000,100;", 14)          'Input characteristic point 2: 4000µm/m = 100%
  CALL WriteLine("OCP1,0,0;", 9)                'Output characteristic point 1: 0% = 0V
  CALL WriteLine("OCP2,100,10;", 12)            'Output characteristic point 2: 100% = 10V
  CALL WriteLine("CAP0;", 5)                    'Calibrate
  CALL WriteLine(einheit3$, 10)                 'Display unit µm/m
  CALL WriteLine("SFC142,955;", 11)             'Cut–off frequency 100 Hz,Bessel

  CALL WriteLine("PCS3;", 5)                    'Select Channel 3 (e.g. ML55)
  CALL WriteLine("SAD14,354;", 10)              'Ub=5V,SG half bridge
  CALL WriteLine("GFV0;", 5)                    'no K–Factor, basic unit mV/V
  CALL WriteLine(einheit1$, 10)                 'physic. unit %
  CALL WriteLine("CAP1,0,0;", 9)                'Input characteristic point 1: 0mV/V = 0%
```

```
CALL WriteLine(”CAP2,2,100;”, 11)        ’Input characteristic point 2: 2mV/V = 100%
CALL WriteLine(”OCP1,0,0;”, 9)           ’Output characteristic point 1: 0% = 0V
CALL WriteLine(”OCP2,100,10;”, 12)       ’Output characteristic point 2: 100% = 10V
CALL WriteLine(”CAP0;”, 5)               ’Calibrate
CALL WriteLine(einheit4$, 10)            ’Display unit mV/V
CALL WriteLine(”SFC142,955;”, 11)        ’Cut–off frequency 100 Hz,Bessel

CALL WriteLine(”PCS2,3;”, 7)             ’Select channels 2 and 3
CALL WriteLine(”AIS42;”, 6)              ’Apply measuring signal at amplifier input
CALL WriteLine(”ICS1,110;”, 9)           ’1 decimal place , step 1
CALL WriteLine(”SIS2,203;”, 9)           ’Display net value of Channel 2 on AB22A
CALL WriteLine(”CAL;”, 4)                ’Calibrate active channels

CALL WriteLine(”MCS2,3,17;”, 10)         ’Record channels 2, 3 and time channel
CALL WriteLine(”MSS215;”, 7)             ’Record net signal
CALL WriteLine(”ICR6317;”, 8)            ’Sampling rate 2400 Hz (default)
CALL WriteLine(”MBF1253;”, 8)            ’Measurement output format (binary, 4 byte, LSBxxxxMSB)

CALL WriteLine(”TSV3,1,2;”, 9)           ’Start measurement in device 2 (slave)
                                         ’3 measured value lines, 1 line pre–trigger, a total of 2
                                         measurements
i = 1
CALL WriteLine(”TSV3,1,2;”, 9)           ’Start measurement in device 1 (master)
                                         ’3 measured value lines, 1 line pre–trigger, a total of 2
                                         measurements
anzsig% = 3                              ’3 signals channel X, channel Y, time channel)
anzzln% = 3                              ’3 measured value lines (p1 of TSV command)
anzwdh% = 2                              ’2 repetitions (p3 of TSV command)
```

```
  ReadOMP                                   'Measurement ended ?
  ReadRMB1                                  'Read in measured values of device 1
  i = 2                                     'Select device 2
  ReadRMB2                                  'Read in measured values of device 2
end:
    FOR i = 1 TO 2
        st% = ILWRT(dev%(i), "SRB0;", 5)    'Switch off acknowledgement
        st% = ILWRT(dev%(i), "DCL;", 4)     'Exit HBM Interpreter
    NEXT i
  END IF
END
 SUB IFinit                                 'Initialise IEEE interface
    BD% = ILFIND("GPIB0")                   'Board address
    IF BD% < 0 THEN
        PRINT : PRINT "'GPIB0' board not found, please check with IBCONF"
        PRINT "and carry out computer warm start."
        j = –1
    ELSE
        FOR i = 1 TO 2                      '2 devices
            READ dev$                       'Device names
            dev%(i) = ILFIND(dev$)          'Device address
            IF dev%(i) < 0 THEN
              PRINT : PRINT "Board '" + dev$ + "' not found, please check with IBCONF"
              PRINT "and carry out computer warm start."
              j = –1
            END IF
        NEXT i
    END IF
END SUB
```

```
SUB Load STATIC                                 'Read routine
    RD$ = SPACE$(80)                            'Delete string
    st% = ILRD(dev%(i), RD$, 80)                'Read in string
    IF st% AND &H8000 THEN                      'GPIB error occurred ?
        PRINT : PRINT "GPIB error device: "; i
        IF st% AND &H4000 THEN                  'Time out error occurred ?
            PRINT "Time out!"
            STOP                                'Stop program
        END IF
    END IF
END SUB

SUB ReadESR STATIC                              'Error routine
    st% = ILWRT(dev%(i), "*ESR?;", 6)           'Output command
    RD$ = SPACE$(10)                            'Delete string
    st% = ILRD(dev%(i), RD$, 10)                'Read in string
END SUB

SUB ReadOMP                                     'Measurement ended ?
    PRINT "Wait for trigger!"
    PRINT
    i = 1
    DO
        CALL WriteLine("OMP?;", 5)              'Reading pointer status ?
        n% = 0
        FOR j = 1 TO 4                          'Extract number of measured value lines
            n% = INSTR(n% + 1, RD$, ",")
```

```
        IF j = 3 THEN m% = n%
      NEXT j
      n% = VAL(MID$(RD$, m% + 1, n% – m% – 1))
  LOOP UNTIL n% = anzzln% * anzwdh%            'Number of measured value lines * repetitions achieved?
END SUB

SUB ReadRMB1                                   'Read in measured values of device 1
  rmbbef$ = "RMB?" + STR$(anzzln% * anzwdh%) + ",6409;"
  CALL WriteLine(rmbbef$, 12)                  'Read off measured values (measured value lines *
                                               repetitions)
  PRINT "  Device 1:"                          'Header
  PRINT "  Channel 3   Channel 4   Time channel"  'Header

  x = 4
  FOR j = 1 TO anzwdh%                         'Number of repetitions
  FOR k = 1 TO anzzln%                         'Number of measured value lines
  FOR l = 1 TO anzsig%                         'Total signals of all channels
    z$ = CHR$(0) + MID$(RD$, x, 3)             'Read in measured value
    IF l = 1 THEN                              'Channel 4
      value! = ((CVL(z$) / 256) * 2) / 7680000   'mV/V
      PRINT USING "+##.###"; value!;
      PRINT " mV/V   ";
    ELSEIF l = 2 THEN                          'Channel 4
      value! = ((CVL(z$) / 256) * 5000) / 7680000   'Ohm
      PRINT USING "###.###"; value!;
      PRINT " Ohm    ";
    ELSE                                       'Time channel
      value! = ((CVL(z$) / 256) * 416.7) / 1000000   's (1/2400Hz = 416.7µs)
      PRINT USING "##.###"; value!;
```

```
            PRINT " s";
        END IF
        x = x + 4
    NEXT l
    PRINT
    NEXT k
    PRINT
    NEXT j
END SUB

SUB ReadRMB2                                        'Read in measured values of device 2
    rmbbef$ = "RMB?" + STR$(anzzln% * anzwdh%) + ",6409;"
    CALL WriteLine(rmbbef$, 12)                     'Read off measured values (measured value lines *
                                                    repetitions)
    PRINT "  Device 2:"                             'Header
    PRINT "  Channel 2    Channel 3    Time channel"  'Header

    x = 4
    FOR j = 1 TO anzwdh%                            'Number of repetitions
    FOR k = 1 TO anzzln%                            'Number of measured value lines
    FOR l = 1 TO anzsig%                            'Total signals of all channels
        z$ = CHR$(0) + MID$(RD$, x, 3)              'Read in measured value
        IF l = 1 THEN                               'Channel 3
            value! = ((CVL(z$) / 256) * 2 / 2 * 4000) / 7680000          'µm/m (2/2 = 2/k-Factor)
            PRINT USING "+####.#"; value!;
            PRINT " µm/m   ";
        ELSEIF l = 2 THEN                           'Channel 4
            value! = ((CVL(z$) / 256) * 2) / 7680000    'mV/V
            PRINT USING "+##.#"; value!;
```

```
            PRINT " mV/V   ";
        ELSE                                              'Time channel
            value! = ((CVL(z$) / 256) * 416.7) / 1000000  's (1/2400Hz = 416.7µs)
            PRINT USING "##.###"; value!;
            PRINT " s";
        END IF
        x = x + 4
    NEXT l
    PRINT
    NEXT k
    PRINT
    NEXT j
END SUB

SUB WriteLine (WR$, n%)
    st% = ILWRT(dev%(i), WR$, n%)                         'Output command
    CALL Load                                             'Read in string
    IF LEFT$(RD$, 1) = "?" THEN                           'Error ?
        CALL ReadESR                                      'Call error routine
        PRINT : PRINT WR$; "command", "Error: "; RD$
        STOP
    END IF
END SUB
```

Explanations:

a) The same device designations must be used in the DATA instruction as were allocated for the appropriate addresses in the device map in the IBCONF file.

b) By means of the startup routine, it is first of all established whether all the devices can be addressed under the device names assigned to them. If this is not the case, an error message appears. Nevertheless, should no data exchange be achieved with the devices, then this is pointed out by means of the message: ”Time out!”.

c) In the FOR-NEXT loop, for each device separately, if a measurement run is in progress it is terminated and ’command acknowledgement’ is switched on.

d) In the program blocks ”device1:” and ”device2:”, the set–up commands are sent to the MGCplus devices and the responses to the commands read in. In the event of an error message, see *ESR? command, this is output and the program stopped.

e) With the commands WriteLine(”MCS3,4,17;”,10) to WriteLine(”TRD6500,151;”,12) the measurement run on Device 1 is predefined.

f) With the commands WriteLine(”MCS2,3,17;”,10) to WriteLine(”MBF1253;”,8) the measurement run on Device 2 is predefined.

g) The commands WriteLine(”TSV3,1,2;”,9) finally start the measurement run, whereby device 2 (slave) must be started first.

h) The measurement run which has been started and is awaiting a trigger event can be seen as a measuring device symbol on the AB22A screen. In the program, the message ”Wait for trigger !” appears.

i) The measurement is triggered if the measured value of channel 3 (device 1) exceeds the threshold of 1,000 mV/V.

j) If the program runs with no errors, the following display appears on the screen, depending on the channel mismatch:

Device 1:

Channel 3	**Channel 4**	**Time channel**	
+1.000mV/V	346.742Ohm	9.791s	’Measured value line 1 (pre–trigger)
+1.001mV/V	346.741Ohm	9.792s	’Measured value line 2 (post–trigger)
+1.000mV/V	346.740Ohm	9.792s	’Measured value line 3 (post–trigger)
+0.999mV/V	346.741Ohm	9.792s	’2nd measurement
+1.000mV/V	346.740Ohm	9.793s	
+1.000mV/V	346.741Ohm	9.793s	

Device 2:

Channel 2	**Channel 3**	**Time channel**	
+1729.4µm/m	–1.1mV/V	9.84 5s	’Measured value line 1 (pre–trigger)
+1729.3µm/m	–1.0mV/V	9.845s	’Measured value line 2 (pre–trigger)
+1729.3µm/m	–1.1mV/V	9.845s	’Measured value line 3 (pre–trigger)
+1729.3µm/m	–1.0mV/V	9.846s	’2nd measurement
+1729.3µm/m	–1.1mV/V	9.846s	
+1729.3µm/m	–1.0mV/V	9.847s	

H HBM Interpreter commands

1 Important conventions

These conventions and general hints make it easier for you to work with HBM Interpreter commands.

Notation

- All commands can be entered in lower or upper case letters.

Abbreviated commands

- Abbreviated commands consist of 3 to 5 characters and, depending on the command, of a list of parameters, which are comma–delimited.

e.g. PCS2(x)

Spaces (blanks)

- In the case of parameters, any prefixed or suffixed blank spaces are suppressed.

IEEE commands

- Standard IEEE commands start with an asterisk (*).

e.g. *ESE

Command types: – Set–up commands – Query commands

- Set–up and query commands are effective on all selected (i.e. active) channels (please see CHS and PCS commands in this respect).
- Query commands – are used for reading off information – they are identified by means of a question mark (?).

e.g. ADS?1302

Responses

- The standard response for all set–up commands is:

Response:

Acknowledge	Meaning
0	Command has been executed
?	Error

The response was omitted when writing the command.

Character strings

- Character strings must be enclosed in inverted commas when inputting them. Inverted commas are also used for the response.

e.g. UCC”TEST”(x)

Command terminator

when inputting:

- The command terminator is identified with (x). Permitted command terminators are:

  ’;’, LF, LFCR, CRLF, <EOI>

in the case of the response:

- The end marker in the case of a response is (y). The end marker is always CRLF and in the case of the IEEE interface also <EOI>.
- Responses are only displayed for one channel.

I/O with numbers

- Handling numbers:
  according to the Standard IEEE 488.2, all numeric parameters can be entered in floating decimal–point format, even if dealing with whole numbers or fixed decimal–point format.

  The figures entered are converted into the number type of the particular parameter and, where necessary, rounded off.
- Numbers are always output with a fixed decimal point.

Interfaces

- In the case of serial and printer interfaces, communication via computer starts with the permissible control characters.

  ’CTRL R’ or ’CTRL B’ and ends with ’CTRL A’ or
  the command DCL.
- The software handshaking procedures ’CTRL Q’ (X-ON) and ’CTRL S’ (X-OFF) are supported.
- Each command generates a response in the case of serial and printer interfaces.

Behaviour on acknowledgement

You can choose whether or not a response is to be output from the MGC*plus* in the case of set–up commands (see SRB command). Query commands – identified by a ’?’ – always give rise to a response (output data).

In the case of set–up commands the following is output:

A ’0’ for error–free execution or

A ’?’ if an error occurs (also applies to unrecognised commands)

e.g. AIS42(x)
0(y)

Activate Interpreter

- If the HBM Interpreters is activated, manual operation via the AB22A/AB32 display and control panel is blocked apart from a few exceptions (Display <=>).

Modifying parameters

- If parameters are modified which have an effect on the measurement itself, then a calibration is carried out after the entry which can last 1 to 3 secs., depending on the amplifier.

Norms and standards

- The Standard IEEE 488.2, which defines codes, formats and some general commands was on the whole considered as a possibility.

1.1 Command construction

All the commands used have a specific structure.

Basically there are two command types:

- **Set–up commands:**

The MGC*plus* is set up via the computer.
Example: SBR1407,1351,1,1300(x)
0(y)
The current Interface has been set to 4800 baud, even parity and 1 STOP bit. The word length is always 8 bits.

- **Query commands:**

Measured values or device settings are read off from the MGC*plus* and are displayed on the screen.
Example: SBR?1300(x)
1407,1351,1,1301(y)
The RS232 interface is set to 4800 baud, even parity and 1 stop bit.

1.1.1 Command structure

Abbreviated commands	Parameters	End markers
***TTT?**	**p1, p2, ...p n**	**(x)**

Example:
*PRE?(x)

*	only in the case of standard IEEE commands
TTT	Abbreviated commands as alphabetical characters (a ... z)
?	only in the case of query commands
p1, p2...pn	Parameter values, consisting of operating sign (+/–) and figures (0...9) or character strings (always in inverted commas ” ”). A positive operating sign can also be omitted.
,	Separator
(x)	Command terminator: Line Feed (LF), Semi–colon (;), Carraige Return/Line Feed (CRLF) or Line Feed/Carriage Return (LFCR). In the event of operation via an IEEE488 interface, also activation of the EOI line on the transfer of the last character.

CR	ASCII characters Carriage Return = decimal 13
LF	ASCII characters Line Feed = decimal 10
;	ASCII characters Semi–colon = decimal 59

If a supplementary parameter – e.g. parameter 2 – is left out, then at least a separator must be entered.

e.g. CAP1,,0(x)

If all of the supplementary parameters are omitted from a specific point, the input can be finished off using the command terminator.

These commands are always effective on all selected (active) channels (see CHS and PCS commands).

1.1.2 Data output structure (responses)

q1, q2...qn(y)

Example 1:

*IDN?(x)
HBM,MGCplus–CP32,0,P1.10(y)

Example 2:

PCS?(x)
16(y)

The responses sent by the MGC*plus* are printed *in italics* in this documentation (second line in the examples).
Output values:

q1,q2...qn	Numerical values with operating sign, character strings (always in " ") or
'?'	as an error message
,	Separator
:	Block separator
(y)	End sequence (CRLF). In the case of an IEEE interface also EOI.

1.1.3 Description of individual commands

Note:

In addition to the MGC*plus* commands, all of the MGC commands are listed (Chapter H).

In **Chapter I** you will find the global parameter list.

On the following pages, each command is listed, its structure broken down and an explanation given by means of an example.

Command

The character string which you must enter to operate the MGC*plus*.

e.g. DCL

Syntax

Command notation to which you must adhere:

e.g. SAD p1,p2(x)

Parameter

The significance of any parameters is explained:

e.g. if, in the case of the command SAD, the parameter p1=11, this means:

1V bridge excitation voltage

Effect

e.g. Explanation how the MGC*plus* is set up.

Response

The MGC*plus* responds to your input. In the case of terminal operation, you will see this response on your screen (always in the case of output commands, but only on request in the case of input commands), see also page 4.

Example

The example shows you the command input and the response from the MGC*plus*. The response is always shown in italics.

You will find the individual commands in **Chapters J and K** sorted alphabetically and listed according to function.

Related commands

All commands relating to a specific application will be executed (see also: functional list of commands). Several groups of related commands may be listed.

Communication processor

The communication processor to be used.

Amplifier type

The amplifier to be used.

MC	(MC01, MC10, MC30, MC35, MC50, MC55, MC60)
MLxx	(ML01, ML10, ML30, ML35, ML50, ML55, ML60)
MLxxB	(ML01B, ML10B, ML30B, ML35B, ML50B, ML55B, ML60B)

ML801 (multi-channel amplifier) (exclusively)
ML71 (exclusively)
ML77 (exclusively)
ML78 (exclusively)

2 Communications behaviour

2.1 Addressing

Control characters (only in the case of RS232C / RS-485 and printer interfaces):

CTRL R: begin communication via computer (ASCII code 18 decimal)

When CTRL R is sent for the first time, there is a wait time of up to 7 seconds, depending on the device configuration, before you can send the next command (typically, this is 2 – 3 seconds).

CTRL B: begin communication via computer (ASCII code 2 decimal)

Once you have input one of these control characters, with a few exceptions the MGC*plus* can no longer be operated by the AB22A/AB32 display and control panel (display '<=>').

CTRL A: end communication via computer (ASCII code 01 decimal)

Once you have input this control character, the MGC*plus* can be operated by the AB22A/AB32 display and control panel again.

DCL

Device Clear
Terminate communication

Syntax: DCL (x) or in the case of RS232C / RS485 and printer interface, control character CTRL A (ASCII code 01 decimal).

Parameter: none

Effect: remote control operation is terminated

Response: none

Note:

You cannot input another command after this one until approx. 3s have elapsed.

Related commands: **ADR?, CHS, DCL, DGM, PCS, RES, *RST, Sxx**

Communications processor: **CP12, CP32**

ADS

Address Setup **For CP32B only**

Setting up the interface addresses

Syntax: ADSp1,p2(x)

Parameter:

p1	Interface
1302	RS485
1303	IEEE488
1305	Ethernet

p2	Address
0–31	for RS485 interface
0–30	for IEEE488 interface
±N	for Ethernet interface (in preparation)
N:	IP address (Internet protocol)

The Ethernet address (IP address) is a positive or negative integer value, which correlates to the dot notation as follows:

Conversion example

172.16.3.235 dot – long AC 10 03 E8 intel – mot. EB 03 10 AC hex – dec –352120660

Related commands: **ADS, CFS, CGP, CMF, DEL, DIR, DSD, EES?, HCF?, ICR, IHD, IPA, MBF, MCS, MRG, MSS, MVF, OMP, PHD, PSC, RHD?, RMB?, RMS?, RMV?, SBR, SCM, SHD, SIS, SML, SMS, SNO?, SPP, SPS, SRP, STD, TCD, TRD, TRE, TRG, TRR, TSV**

Communications processor: **CP12, CP32**

ADS?

Address Setup Query For CP32B only
Output the interface addresses

Syntax: ADS?p1(x)

Parameter: p1: interface in accordance with ADS command (p1)

Response: *q1(y):* for address see ADS command (p2)

q1=–1, no interface card present

Communications processor: **CP12, CP32**

IPA

IP Address Setup
Sets the IP address for the CP32

Syntax: IPA p1(x)

Parameter: p1: IP address in point notation
Example: 172.16.3.235

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

Communications processor: **CP32, CP32B**

IPA?

IP Address Setup
Outputs the IP address for the CP32

Syntax: IPA ?(x)

Response: q1(y)
IP address in point notation

Sxx

Select
Selects the MGC*plus* with the address xx

Syntax: S00(x) to S99(x)

Parameter: none

Effect: the Select command lets you individually address a maximum of 32 MGC*plus* devices interconnected on an RS422/485 bus.

There are 32 usable addresses 0...31 (set them up with switches A1-A5 on the CP12 connection board, or with the ADS command in the case of a CP32B communications processor, or with an ABxx display and control panel). In the case of the Select command, these addresses are repeated as addresses 32...63 and 64...95, i.e. the commands S00, S32, S64 correspond to MGC*plus* devices with the address 00, but have a different effect on them.

Addresses 96...99 are intended for special functions.

Address Sxx	**Device with specified address Sxx**		**All other devices**	
	Execute command	Responses	Execute command	Responses
00...31	yes	yes[2)]	no	no
32...63	yes	yes[2)]	yes	no[1)]
64...95	yes	no[1)]	as for last address selected	
96	no	no	no	no
97,98	yes	no[1)]	yes	no[1)]
99[3)]	yes	yes[2)]	yes	yes[2)]

1) The response to the previous command is internally stored.
2) The stored response to the previous command is then output.
3) Factory setting

Explanation:

S00...S31(x)

Only the device with the specified address receives commands, executes them and responds.

S32...S63(x)

All devices receive all commands and execute them. Only the device with the specified address (S32 = device 0) responds on behalf of all devices.

S64...S95(x)

The device with the specified address is set up as an additional member that receives and executes all commands but does not send a response.

S96(x)

All devices wait for Select and do not send responses.

S97 (x) or S98(x)

All devices receive and execute all commands, but do not send responses.

S99(x)

All devices on the bus are active, receive all commands and send responses (leads to collision on the bus when there are several devices). This is the default setting when devices are switched on.

Related commands: **ADR?, CHS, DCL, DGM, PCS, RES, *RST , Sxx**

Communications processor: **CP12, CP32**

RES

Reset
Execute warm restart

Syntax:	RES (x)
Parameter:	none
Response:	none
Effect:	the device executes a warm restart. Communication is terminated.

Note:

The command RES is an HBM command.

Related commands: **ADR?, CHS, DCL, DGM, PCS, RES, *RST Sxx**

Communications processor: **CP12, CP32**

*RST

Reset
Execute warm restart

Syntax:	*RST (x)
Parameter:	none
Response:	none
Effect:	the device executes a warm restart. Communication is terminated.

Note:
The command *RST is a standard IEEE command.

Related commands: **ADR?, CHS, DCL, DGM, PCS, RES, *RST, Sxx**
Communications processor: **CP12, CP32**
Amplifier type: **MC, MLxx**

2.2 Communication computer/MGC*plus*

SBR

Set Baud Rate
Set baud rate, parity and number of stop bits

Syntax: SBRp1,p2,p3,p4(x)

Parameter: p1: baud rate (see Tab.22)
p2: parity (see Tab.21)
p3: number of stop bits (1 or 2)

see 'Global Tables' (parameter list)

p4	Interface
1300	current interface
1301	RS-232
1302	RS-485

Related commands: **BDR, SBR, SRB**
Communications processor: **CP12, CP32**

SBR?

Set Baud Rate Query
Output baud rate, parity and number of stop bits

Syntax: SBR?p1(x)

Parameter: p1: interface (see table 22)

Response: *q1,q2,q3,q4(y)*

SRB

Select Response Behavior
Select the response behaviour of the current interface

Syntax: SRB p1(x)

Parameter:

p1	Switch response output on/off
0	Switch off response output
1	Switch on response output

Effect: There are two types of command:

a.) Query commands (e.g. MSV?) are identified by a question mark and generate output data regardless of the response behaviour selected for the interface. It is not possible to stop such data being output with a command of this kind.

b.) Set-up commands (e.g. SRB) generate acknowledgment data (0 or ?). You can define whether such data is output with this kind of command by switching the option on or off.

After switching on the MGC*plus*, the following factory settings apply:

Interface	p1	Note
IEEE	0	off
RS232C	1	on
RS-485	1	on
Printer	1	on
Ethernet	1	on

The IEEE interface does not respond to set-up commands. If a response is required, the option must be switched on with this command (SRB).

Response:

Response	Meaning
0	The command is executed if SRB 1(x) has been input
?	Error if SRB 1(x) has been input
none	The command is executed or error if SRB 0(x) has been input

Related commands: **BDR**, **SBR**, **SRB**

Communications processor: **CP12, CP32**

SRB?

Select Response Behavior Query
Output the response behaviour of the current interface

Syntax: SRB?(x)

Parameter: none

Response: *q1(y)*

q1	Switch response output on/off
0	off
1	on

2.3 Error handling, status register

***ESR?**

Standard Event Status Register
Output the error status register

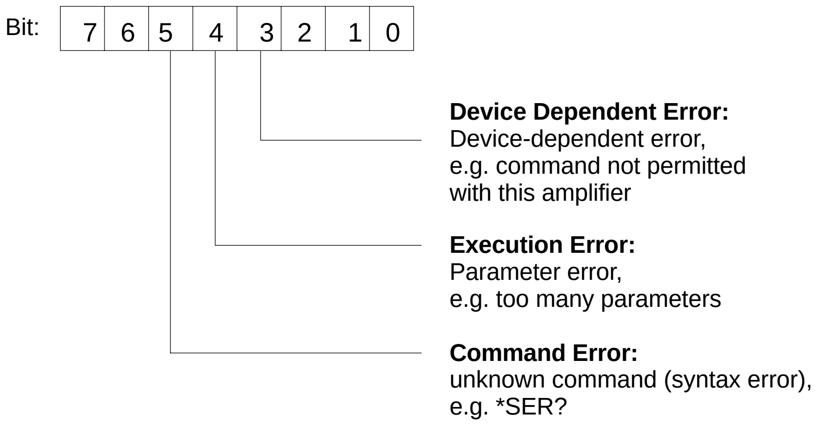
Syntax: *ESR? (x)

Parameter: none

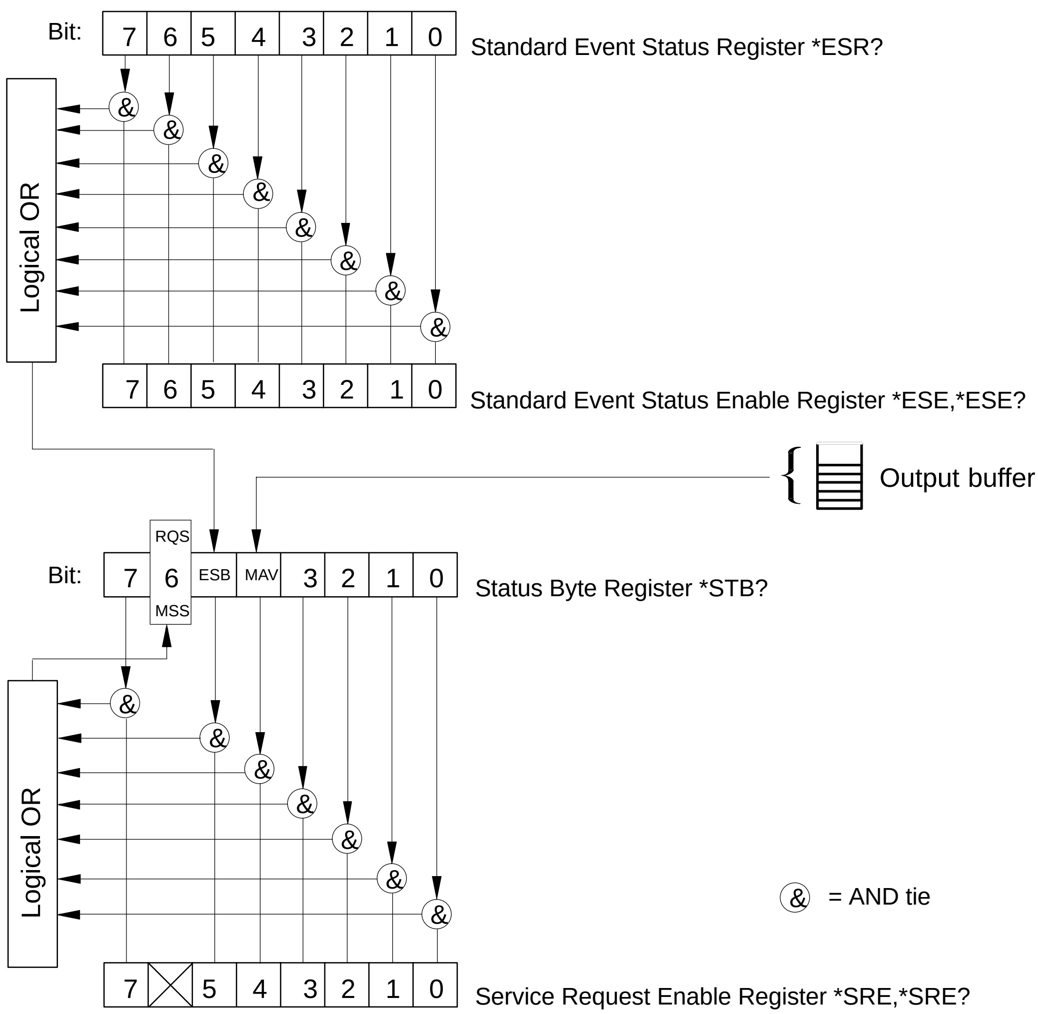
Effect: the contents of the Standard Event Status Registers (ESR) are output in decimal equivalent. The Standard Event Status Register (ESR) is set when communication errors occur. Different fault causes set different bits, so that errors can be precisely identified.

Response: *q1(y)*

q1: 8, 16 or 32



All other bits are unassigned.



Explanations about the various registers can be found under the commands SRE, STB, ESR, ESE and in the glossary.

RQS	Request Status: Service Request has been made.
ESB	Event Summary Bit: Summary of all the Standard Event Status Register bits released in the Standard Event Status Enable Register.
MAV	Message available: The output buffer contains a message ready to be retrieved.
MSS	Master Summary Status Summary of all the Status Byte Register bits released in the Service Request Enable Register.

Related commands: ***CLS**, ***ESE**, **ESR?**, **IST?**, **PPM**, ***PRE**, ***SRE**, **STB?**, **EST?**

Communications processor: **CP12, CP32**

*ESE

Standard Event Status Enable Register
Input the ESR enable bit mask

Syntax: *ESE p1(x)

Parameter:

p1	Decimal equivalents of the 8-bit ESE register
	0 – 255

Effect: this command sets the ESE masks. It enables error messages to be suppressed. An error bit is set in the Standard Event Status Register. This only causes the event summary bit (ESB) to be set in the Status Byte Register if the associated bit is set in the Standard Event Status Enable Register (see also command *ESR?). This allows you to define which causes of faults will give rise to a Service Request.

Initial status:

255 (free), i.e. every error that occurs in the ESR gives rise to an event summary bit (ESB) in the Status Byte Register (STB).

Related commands: ***CLS, *ESE, ESR?, IST?, PPM, *PRE, *SRE, STB?, EST?**

Communications processor: **CP12, CP32**

*ESE?

Standard Event Status Enable Query
Output the ESR enable bit mask

Syntax: *ESE?(x)

Parameter: none

Effect: outputs the current contents of the Standard Event Status Enable (ESE) Register.

Response: *q1(y)*

q1	Contents of ESE register
	0 – 255

PPM

Parallel Poll Mode
Input the parallel poll response (IEEE-488 only)

Syntax: PPM p1(x)

Parameter:

p1	Parallel poll response
0	No parallel poll response (sense = 1)
1 – 8	Parallel poll response (sense = 1)
9 – 16	Parallel poll response (sense = 0)
17	No parallel poll response (sense = 0)

Effect: specifies on which data line and at what level a device is to respond in the event of a parallel poll. This command must be used instead of the IEEE interface commands PPE and PPD.

Related commands: ***CLS**, ***ESE**, **ESR?**, **IST?**, **PPM**, ***PRE**, ***SRE**, **STB?**, **EST?**

Communications processor: **CP12, CP32**

PPM?

Parallel Poll Mode Query
Output the parallel poll response (IEEE-488 only)

Syntax: PPM?(x)

Parameter: none

Effect: the setting for the parallel poll response is output.

Response : *q1(y)*

q1	Parallel poll response
0	No parallel poll response (sense = 1)
1 –8	Parallel poll response (sense = 1)
9 –16	Parallel poll response (sense = 0)
17	No parallel poll response (sense = 0)

Effect: specifies the data line on which a device is to respond in the event of a parallel poll, and the response required.

*STB?

Status Byte Register Query
Output the STB register

Syntax: *STB?(x)

Parameter: none

Effect: outputs the Status Byte Register.

The Status Byte Register contains information on whether there is a message in the output buffer, an error has occurred, or a Service Request has been made (see also command *ESR?).

Similarly, in the event of Serial Poll over the IEEE interface, the Status Byte Register for the addressed device is output.

Response: *q1(y)*

q1	Contents of STB register	
16	MAV	Message available in output buffer.
32	ESB	Error sum bit set.
64	RQS	Service Request requested.

Related commands: ***CLS, *ESE, ESR?, IST?, PPM, *PRE, *SRE, STB?, EST?**

Communications processor: **CP12, CP32**

*SRE

Service Request Enable Register
Input the STB enable bit mask

Syntax: *SRE p1(x)

Parameter:

p1	Decimal equivalents of the 8-bit SRE register
	0 – 63, 128 – 191

Effect: this command sets the SRE register bits. It allows Service Requests to the IEEE interface to be enabled or disabled.

If a bit in the Status Byte Register is set, and if the associated bit in the Service Request Enable Register is masked, this has the following effects:

- The master summary status byte (MSS) in the Status Byte Register (STB) is set.
- A Service Request (RQS) is made (see also command *ESR?).

Initial status:

191 (free), i.e. all possible status changes in the STB register give rise to an MSS bit (Master Summary Status bit) in the STB register. This leads to a Service Request.

Related commands: ***CLS**, ***ESE**, **ESR?**, **IST?**, **PPM**, ***PRE**, ***SRE**, **STB?**, **EST?**

Communications processor: **CP12, CP32**

*SRE?

Service Request Enable Query
Output the STB enable bit mask

Syntax: *SRE?(x)

Parameter: none

Effect: outputs the current contents of the Service Request Enable (SRE) register.

Response: *q1(x)*

q1	Contents of SRE register
	0 – 63, 128 – 191

*CLS

Clear Status
Clear all queues and event register

Syntax:	*CLS(x)
Parameter:	none
Effect:	all Event Status Registers showing a sum bit in the Status Byte Register are cleared, along with the output buffer.
Response:	none
Example:	*CLS(x)
	The ESR register, ESB bit and output buffer are cleared.

Related commands: ***CLS**, ***ESE**, **ESR?**, **IST?**, **PPM**, ***PRE**, ***SRE**, **STB?**, **EST?**
Communications processor: **CP12, CP32**

***IST?**

Individual Status Query
Output the response standby status in the event of parallel poll

Syntax: *IST?(x)

Parameter: none

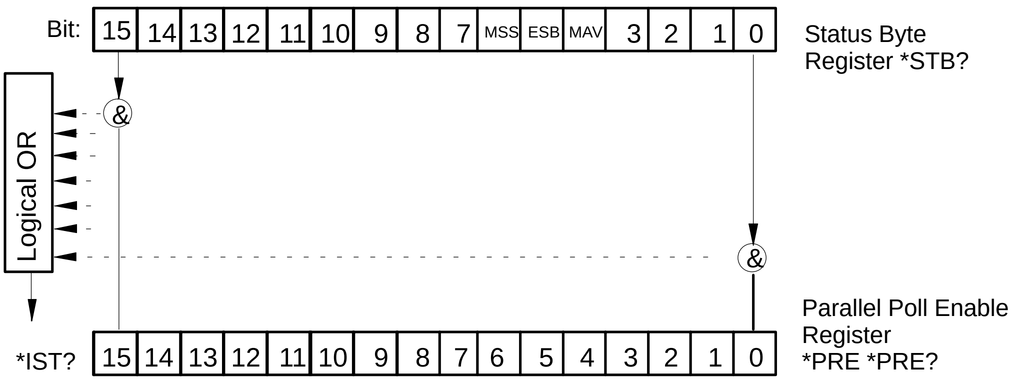
Effect: output the response standby status in the event of parallel poll (i.e. the sum bit from a tie of the Status Byte Register and the Parallel Poll Enable Register).

Response: *q1(y)*

q1	Response standby status in the event of parallel poll
0	The MGC*plus* does not respond in the event of a parallel poll query
1	The MGC*plus* responds in the event of a parallel poll query

Example: *IST?(x)
1(y)

The MGC*plus* responds in the event of a parallel poll.



Related commands: ***CLS, *ESE, ESR?, IST?, PPM, *PRE, *SRE, STB?, EST?**

Communications processor: **CP12, CP32**

*PRE

Parallel Poll Enable Register
Input the PRE bit mask

Syntax: *PRE p1(x)

Parameter:

p1	Decimal equivalents of the 16-Bit PRE register
	0 – 65 535

Effect: this command sets the PRE register bits. Each bit in this register is assigned to a bit in the Status Byte Register. If a bit in the Status Byte Register and the associated bit in the Parallel Poll Enable Register are set, then in the event of a Parallel Poll query, the parallel poll response set up with the command PPM is output (see also command *IST?).

Initial status:

65 535 (free), i.e. all possible status changes in the STB register give rise to a response in the event of a Parallel Poll query.

Example 1: *PRE0(x)

0(y)

No response in the event of parallel poll

Example 2: *PRE64(x)

0(y)

The MGC*plus* is required to respond in the event of Parallel Poll if the MSS bit (Master Summary Status bit) is set.

Related commands: ***CLS, *ESE, ESR?, IST?, PPM, *PRE, *SRE, STB?, EST?**

Communications processor: **CP12, CP32**

***PRE?**

Parallel Poll Enable Register Query
Output the PRE bit mask
Syntax: *PRE?(x)
Parameter: none
Effect: outputs the current contents of the PRE register.
Response: *q1(y)*

q1	Decimal equivalents of the 16-Bit PRE register
	0 – 65 535

Example: **PRE?(x)*
64(y)
Bit 6 of the PRE register is set.
The MGC*plus* responds in the event of Parallel Poll if the MSS bit (Master Summary Status bit) has been set in the Status Byte Register.

EST?

Error Status Query
Error status output

Syntax: EST?p1(x)
Parameter: p1: 0
Cyclic query

p1	Error/warning status set
p2	The content of the amplifier data has changed

p1: 1
Output of the existing errors and warnings in list form, separated by commas. A maximum of 16 errors are output in a list. The most serious error is at the top of the list. The entry ”additional errors” indicates that more than 16 errors are set. If the CalError bit is set in the measurement status, there is a serious error and you can query its cause. For the error status, see table 30.
see ’Global Tables’ (parameter list)

Related commands: ***CLS, *ESE, ESR?, IST?, PPM, *PRE, *SRE, STB?, EST?**
Amplifier type: **MLxxB**

2.4 Identification

AMT?

Amplifier Type Query
Output the amplifier type

Syntax: AMT?(x)
Parameter: none
Response: *q1(y)*

q1	Amplifier type
5000	ML30
5001	ML50
5002	ML01
5003	ML55
5004	ML60
5005	ML35
5006	ML10
5007	ML51
5008	ML55S6
5009	ML38

Related commands: **AID?**, **AMT?**, **CBT?**, **DID?**, ***IDN?**, **IDS?**
Amplifier type: **MLxx**

AID?

Amplifier Identification Query
Output the amplifier identification

Syntax: AID?(x)
Parameter: none
Response: *String*(y)*:* 22 characters per active amplifier
Example: AID?(x)
HBM,RD002-ML30,0,P1.30(y)
Company, device designation, serial number, version number.

Related commands: **AID?**, **AMT?**, **CBT?**, **DID?**, ***IDN?**, **IDS?**
Amplifier type: **MC, ML**

IDS?

Identifier Settings Query
Read out parameter value identification string (global table)

Syntax: IDS?p1(x)

Parameter: p1: parameter value

Response *q1(y):* english language string (max. 20 characters)

Note:

The only parameter values to be defined are new additions, or those used in special versions which were not defined in the first versions of MGC*plus* .

From P4.10:67:	CTRL MEAN VALUE
188:	MEAN GROSS VALUE
189:	MEAN NET VALUE
1150:	FREQ 100kHz
others:	NOT FOUND

Example: IDS?1150(x)

FREQ 100kHz(y)

Related commands: **AID?, AMT?, CBT?, DID?, *IDN?, IDS?**

Amplifier type: **MLxx**

CBT?

Connected Board Type Query
Output the connection board type

Syntax: CBT?(x)

Parameter: none

Response: *q1(y): connection board*

q1	Connection board type
5500	AP01, AP03, AP04, AP07, AP11, AP12, AP13 (standard connection board)
5501	AP05, AP06 (connection board with Zener barrier)
5502	AP08 (charge amplifier)
5503	AP14 (quarter bridge supplementary circuit)
5504	AP18 (connection board with current feed for active piezoelectric transducer)
5505	AP09 (connection board with benchmark measuring point for TC)

Related commands: **AID?**, **AMT?**, **CBT?**, **DID?**, ***IDN?**, **IDS?**

Amplifier type: **MLxx**

*IDN?

Identification Query
Output the device identification

Syntax: *IDN?(x)

Parameter: none

Response: *String(y):* 16 characters

Example: *IDN?(x)
HBM,CP32B,0,P1.12(y)

Company, device designation, serial number, version number.

Related commands: **AID?**, **AMT?**, **CBT?**, **DID?**, ***IDN?**, **IDS?**

Communications processor: **CP12, CP32**

DID?

Display Identification Query
Indicator identification output

Syntax:	DID?(x)
Parameter:	none
Response:	*String(y):* display identification string (16 characters)
Example:	DID?(x)
	HBM,AB22A,0,P3.00(y) Company, device designation, serial number, version number.

Related commands: **AID?, AMT?, CBT?, DID?, *IDN?, IDS?**

Communications processor: **CP12, CP32**

SML

Select Message Language
Sett the message output

Syntax: SMLp1(x)

Parameter: p1 Language

1500 German

1501 English

1502 French

From firmware version P1.23, when critical internal events occur, the CP32B is able to output relevant messages on the AB22A panel, which the user has to acknowledge. You can use the SML command to set the message output language.

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

Communications processor: **CP12, CP32**

SML?

Select Message Language Query
Query the language setting

Syntax: SML?(x)

Response: q1 current language setting

EES?

Extended Error Status
Queries the internal error status of the CP32 and the associated error texts

Syntax: EES?p1(x)

Parameter: p1 0 is there an error, a warning?

Response: q1, q2 (y)

q1	0: no error, 1: error
q2	0: no warning, 1: warning

p1 1 List of all errors and warnings

Response: *q1,...qn(y)* Error codes (see table)

p1 ≥ 16000 outputs error text

Response: q1 (y)

q1: error text

Link resource conflict:

EES?16001(x) returns a binary block with the unavailable link resources

Binary block format:

INT16 count count * {

INT8 Channel

INT8 Subchannel

INT16 Signal {

Example: EES?1(x)

1(y)

EES?16003(x)

The hard disk is full!(y)

Communications processor: **CP32, CP32B**

Related commands: **ADS, CFS, CGP, CMF, DEL, DIR, DSD, EES?, HCF?, ICR, IHD, IPA, LPR, MBF, MCS, MRG, MSS, MVF, OMP, PCS, PHD, PSC, RHD?, RMB?, RMS?, RMV?, SBR, SCM, SHD, SIS, SML, SMS, SNO?, SPP, SPS, SRP, STD, TCD, TRD, TRE, TRG, TRR, TSV**

SNO?

Serial Number Query (from P1.30)
Read out the serial number of the CP32B

Syntax: SNO?(x)

Parameter: none

Response: *q1(y):* serial number of the CP32B

Related commands: **ADS, CFS, CGP, CMF, DEL, DIR, DSD, EES?, HCF?, ICR, IHD, IPA, LPR, MBF, MCS, MRG, MSS, MVF, OMP, PCS, PHD, PSC, RHD?, RMB?, RMS?, RMV?, SBR, SCM, SHD, SIS, SML, SMS, SNO?, SPP, SPS, SRP, STD, TCD, TRD, TRE, TRG, TRR, TSV**

Communications processor: **CP12, CP32**

2.5 Time and date

STD

Set Time and Date
Set up system time and date

Applies to CP32B only

Syntax: STD p1,p2(x)

Parameter:

p1	System time in the form hh:mm:ss
p2	System date in the form dd.mm.yyyy

Related commands: **ADS, CFS, CGP, CMF, DEL, DIR, DSD, EES?, HCF?, ICR, IHD, IPA, LPR, MBF, MCS, MRG, MSS, MVF, OMP, PCS, PHD, PSC, RHD?, RMB?, RMS?, RMV?, SBR, SCM, SHD, SIS, SML, SMS, SNO?, SPP, SPS, SRP, STD, TCD, TRD, TRE, TRG, TRR, TSV**

Communications processor: **CP12, CP32**

STD?

Set Time and Date Query
Output the system time and date

Syntax: STD?(x)

Parameter: none

Response: *q1,q2(y)*

q1	System time in the form hh:mm:ss
q2	System date in the form dd.mm.yyyy

Example: STD?(x)

15:45:31,07.05.1997(y)

Communications processor: **CP12, CP32**

CGP

Configure GPIB port (from P1.30)
Set GPIB Write timeout

Syntax: CGPp1(x)

Parameter: p1: timeout as a multiple of 55ms.

0: no timeout

Note:

Compared to the other interfaces, the special feature of the GPIB Interface is that it only allows half-duplex communication. This means that once the interface has received a command, it must be switched to Send to issue a response. If the response is not answered, it will not be possible to accept and handle any further commands. So that you still have the oportunity to enter further commands without having to answer the previous response, you can use the CGP command to set up a Write timeout, after which the interface will automatically be switched back to Receive.

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

Communications processor: **CP12, CP32**

CGP?

Configure GPIB port query (from P1.30)
Establish GPIB Write timeout

Syntax: CGP?(x)

Parameter: none

Response: *q1(y):* current timeout

3 Amplifier settings

Note:
Amplifier function commands only have an effect on the selected channels (see PCS command). **Whenever you switch on, all channels are selected (i.e. active).**

3.1 Amplifier input

AIS

Amplifier Input Signal
Select amplifier input signal

Syntax: AISp1(x)

Parameter:

p1	Input signal
40	Internal zero signal
41	Internal calibration signal
42	Measurement signal
43	Reference point (ML01, AP09)
46	Measurement signal without excitation (AP810, AP814)

Related commands: **AIS**, **ASA**, **ASS**, **HPS**, **SAD**, **SAF**

Amplifier type: **MLxx**

AIS?

Amplifier Input Signal Query
Output amplifier input signal

Syntax 1: AIS?(x)

Parameter: none

Response: *q1(y): input signal*

Syntax 2: AIS??(x)

Parameter: none

Response: *q1, ... ,qn(y):* available input signals

HPS

High pass Select
Select cut-off frequency of high pass filter

Syntax: HPSp1(x)

Parameter:

p1	Filter cut-off frequencies
1200	High pass filter off
1201*)	0.1Hz; as from low pass: 0.2Hz Bessel, 5Hz Butterworth
1202*)	1.0Hz; as from low pass: 2.5Hz Bessel, 5Hz Butterworth
1203*)	1.0Hz; as from low pass: 15Hz Bessel, 40Hz Butterworth

***) applies only to MLxxB amplifiers**

Related commands: **AIS, ASA, ASS, HPS, SAD, SAF**

Amplifier type: **MLxx, MLxxB**

HPS?

High pass Select Query
Output set cut-off frequency of high pass filter

Syntax 1: HPS?(x)

Parameter: none

Response: *q1(y):* set filter cut-off frequencies

Syntax 2: HPS??(x)

Parameter: none

Response: *q1, ... ,qn(y):* available filter cut-off frequencies

SAD

Sensor Adaption
Select transducer adaptation

Syntax: SADp1,p2(x)

Parameter:

p1	Excitation voltage (or current) as per tables 2 to 5
p2	Transducer type as per table 16

Related commands: **AIS**, **ASA**, **ASS**, **HPS**, **SAD**, **SAF**

Amplifier type: **MLxx**

SAD?

Sensor Adaption Query
Output set transducer adaptation

Syntax 1: SAD?(x)

Parameter: none

Response: *q1,q2(y)*

q1	Excitation voltage (or current) as per tables 2 to 5
q2	Transducer type as per table 16

Syntax 2: SAD??(x)

Parameter: none

Response: *q1,..,qn(y):* possible excitation voltage or similar as per Tab.2–5

Syntax 3: SAD?,?(x)

Response: *q1..qn(y):* possible transducer type as per Tab.16

SAF

Sensor Adaptation Frequency
Transducer frequency adaptation

Syntax: SAF(x)

Parameter:

p1	Glitch Filter	IDS_ON/IDS_OFF	eliminates pulse widths < 1.6 ms
p2	Frequency quadrupling	IDS_ON/IDS_OFF	evaluates each edge
p3	F2	IDS_ON/IDS_OFF	F2 signal is evaluated
p4	zero index input	IDS_ON/IDS_OFF	enabled hardware input
p5	Transducer error input	IDS_ON/IDS_OFF	enabled hardware input
p6	Switching output 1	IDS_OFF IDS_FREQ1 (232) IDS_PULSE (235)	returns Limit1 at Limit1 returns each activated edge at Limit1 returns F1 count signal (1.6 ms) at Limit1
p7	Switching output 2	IDS_FREQ2 (233) IDS_FREQ2 (233) IDS_DIRECTION(234)	returns F2 at Limit2 returns F2 at Limit2 returns direction of rotation at Limit2 (high = pos.)

Effect:

Glitch Filter
Input signals with pulse widths < 1.6 ms are not evaluated.

Factory settings: Off.

Frequency quadrupling
All edges adjacent to F1 and F2 are evaluated. If F2 is not connected, there is frequency doubling.
Factory settings: Off.

F2 evaluation
Activates detection and evaluation of direction of rotation.
Factory settings: On.

Zero index input
Hardware input for the zero index signal. Relevant for incremental transducers. In count mode, the zero index signal (falling slope) sets the counter reading to 0.
Factory settings: Off.

Transducer error input
Hardware input for transducer error detection. Low active.
Factory settings: Off.

Switching output 1
Off:
Normal limit value functionality at switching output Limit1 (factory settings).

F1 at Limit1:
F1 signal is applied to switching output Limit1.
F1 count signal at Limit1: F1 count signal (1.6 ms pulse width) applied to switching output Limit1.

Switching output 2
Off:
Normal limit value functionality at switching output Limit2 (factory settings).

F2 at Limit2
F2 signal is applied to switching output Limit2.
Direction of rotation signal at Limit2: high: positive direction of rotation, low: negative direction of rotation.

Related commands: **AIS**, **ASA**, **ASS**, **HPS**, **SAD**, **SAF**

Amplifier type: **MLxx**

DSD

Data Set Device
Input of amplifier settings in the case of ML amplifiers

Syntax:	DSDp1(x)
Parameter:	p1: amplifier settings (as a hexadecimal string ”___”)
Effect:	this command is used for reading all of the amplifier settings which were saved with DSD? .
	In the case of multi-channel amplifiers (e.g. ML801), the subchannel number must be included in the DSD string.

Related commands: **ADS, CFS, CGP, CMF, DEL, DIR, EES?, DSD, HCF?, ICR, IHD, IPA, LPR, MBF, MCS, MRG, MSS, MVF, OMP, PCS, PHD, PSC, RHD?, RMB?, RMS?, RMV?, SBR, SCM, SHD, SIS, SML, SMS, SNO?, SPP, SPS, SRP, STD, TCD, TRD, TRE, TRG, TRR, TSV**

Communications processor: **CP12, CP32**

DSD?

Data Set Device Query
Output of amplifier settings in the case of ML amplifiers

Syntax:	DSD?p1,p2(x)
Parameter:	p1: channel number of amplifier (1...16)
	p2: subchannel number (default 1)
Response:	*”___hexadecimal string___”(y):* approx. 178 bytes=356 characters)
Effect:	this command is used to save all of the amplifier settings.
Example:	DSD?10,7(x)
	”0a0401....”(y)
	Output amplifier settings from channel 10, subchannel 7

3.2 Filter setting

SFC

Signal Filtering Characteristic
Input cut-off frequency and filter characteristics

Syntax: SFCp1,p2(x)

Parameter:

p1	Filter characteristics as per table 11
p2	Cut-off frequency as per table 17

Related commands: **AFS**, **ASF**, **SFC**

Amplifier type: **MLxx**

SFC?

Sensor Filtering Query
Output cut-off frequency and filter characteristics

Syntax 1: SFC?(x)

Parameter: none

Response: *q1,q2(y)*

q1	Filter characteristics
q2	Cut-off frequency

Syntax 2: SFC??(x)

Parameter: none

Response: *q1,q2(y):* possible filter characteristics

Syntax 3: SFC?141,?(x)

Response: *q1,..,qn(y):* available Butterworth frequencies

Syntax 4: SFC?142,?(x)

Response: *q1,...,qn(y):* available Bessel frequencies

In the following tables you will find the available cut-off frequencies with Bessel or Butterworth characteristics depending on the particular amplifier.

p1=141	**Butterworth frequency (Hz)**						
p2	ML01	ML10	ML30	ML35	ML38	ML50/51/55	ML60
927					1.000		
930					1.500		
932					2.500		
933					3.000		
935	5.000	5.000	5.000	5.000	5.000		5.000
936					6.000		
940					9.000		
941	10.00	10.00	10.00	10.00	10.00	10.00	10.00
945	20.00	20.00	20.00	15.00		20.00	20.00
948	40.00	40.00	40.00			40.00	40.00
953	80.00	80.00	80.00			80.00	80.00
958			200.0				
959	250.0	250.0				250.0	250.0
962	500.0	500.0				500.0	500.0
969	1000.	1000.				1000.	1000.
972						1500.	
973	2000.	2000.					2000.
974	2400.						

p1=142	Bessel frequency (Hz)						
p2	ML01	ML10	ML30	ML35	ML38	ML50/51/55	ML60
906					0.030		
908	0.050	0.050	0.050	0.050	0.050	0.050	0.050
914	0.100	0.100	0.100	0.100	0.100	0.100	0.100
917	0.200	0.200	0.200	0.200	0.200	0.200	0.200
921	0.500	0.500	0.500	0.500	0.500	0.500	0.500
926					0.900		
929	1.250	1.250	1.250	1.250		1.250	1.250
930					1.500		
932	2.500	2.500	2.500	2.500		2.500	2.500
935	5.000	5.000	5.000	5.000		5.000	5.000
941	10.00	10.00	10.00	10.00		10.00	10.00
945	20.00	20.00	20.00	15.00		20.00	20.00
948	40.00	40.00	40.00			40.00	40.00
955	100.0	100.0	100.0			100.0	100.0
958	200.0	200.0				200.0	200.0
961	400.0	400.0				400.0	400.0
968						900.0	900.0
969		1000					
970	1100						
1100		10000					
1140		50000					
1150*		100000					

* Special version (only for SG full bridges)

SPS

Subchannel Programming Select
Subchannel selection mask

Syntax: SPS p1,..., p128 (x)

Parameter: p1,...,p1281,...128 subchannel selection

SPS 0 (x) selects all the subchannels of a module

This command sets the subchannel selection mask for display. The modules to be set should already be selected with PCS.

Related commands: **ADS, CFS, CGP, CMF, DEL, DIR, EES?, DSD, HCF?, ICR, IHD, IPA, LPR, MBF, MCS, MRG, MSS, MVF, OMP, PCS, PHD, PSC, RHD?, RMB?, RMS?, RMV?, SBR, SCM, SHD, SIS, SML, SMS, SNO?, SPP, SPS, SRP, STD, TCD, TRD, TRE, TRG, TRR, TSV**

Communications processor: **CP32, CP32B**

SPS?

Subchannel Programming Select Query
Subchannel selection mask output

Syntax: SPS?p1(x)

Parameter: p1: output mode

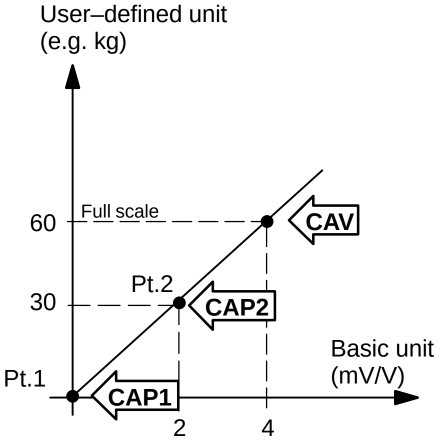
0 existing subchannels
1 selected subchannels

Response: *q1,...,q128*
List of existing or selected ML channels

3.3 Calibration

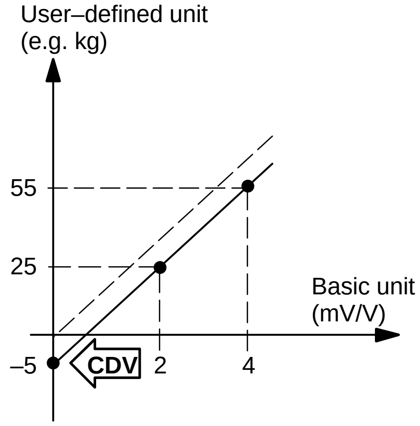
Define two points to determine the input characteristic. Each point is specified by the input signal in the basic unit (mV/V) and in the desired user–defined unit (e.g. kg).

Example for a sequence of calibration commands:



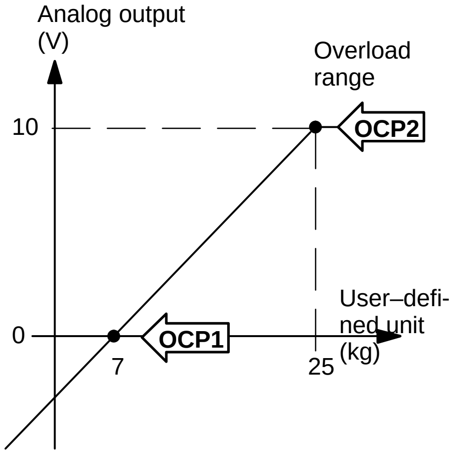
EUN"kg"	Specify user–defined unit, e.g. kg
CAP1,0,0 CAP2,2,30	Enter two points for the calibration straight line Point 1: 0kg at 0mV/V Point 2: 30kg at 2mV/V
CAV0,60	Specify full scale The measuring range is to include 60 kg (the first parameter is insignificant)
CAP0	Activate calibration values in the amplifier Response: 0 Characteristic has been set Response: ? Amplifier range has been exceeded (upper/lower limit)

Use the CDV command for shifting the input characteristic's zero point (in user–defined unit). A positive parameter shifts the zero point into the negative range, a negative one into the positive range.



CDV5	Point 1: –5kg at 0mV/V

A characteristic can be assigned to the input and also to the output. Define two points to determine the output characteristic. Each point is specified by the input signal in the user–defined unit (e.g. kg) and the output signal in Volt.



OCP1,7,0 OCP2,25,10	7kg are to yield 0V at the output 25kg are to yield 10V at the output
CAP0	Activate calibration values in the amplifier Response: 0 Characteristic has been set Response: ? Amplifier range has been exceeded (upper/lower limit)

Special characteristics of the ML01B, ML35B amplifier modules:
The ML01B and ML35B amplifier modules enable a thermocouple or resistance thermometer linearization to be activated. If active, the resulting basic unit always is ”°C ”.
If ”°C ”, ”°F ” or ”K ” have been selected as the user–defined unit, you will obtain a fix input calibration straight line for the appropriate conversion. All input calibration point inputs will be ignored!
The characteristic can be shifted (CDV) and the analog output can be scaled (OCP) as described.

ACL

Autocal
Switch on/off cyclical autocalibration

Syntax: ACLp1(x)

Parameter:

p1	Autocalibration
0	Off
1	On

Related commands: **ACL, CAL, CAN, CAP, CAV CDT, CDV, EUN, GFV, ISV?, IUN, MVA?, OCP, SCL**

Amplifier type: **MC, MLxx**

ACL?

Autocal Query
Output status of cyclical autocalibration

Syntax 1: ACL?(x)

Parameter: none

Response: *q1(y):* current status of cyclical autocalibration

Syntax 2: ACL??(x)

Parameter: none

Response: *q1, q2(y):* possible stati of cyclical autocalibration

CAN

Cal Not
Suppression of cyclical autocalibration

Syntax: CANp1(x)

Parameter:

p1	Suppression of cyclical autocalibration
0	NO
1	YES

Related commands: **ACL, CAL, CAN, CAP, CAV ,CDT, CDV, EUN, GFV, ISV?, IUN, MVA?, OCP, SCL**

Amplifier type: **MLxx**

CAN?

Cal Not Query
Output suppression of cyclical autocalibration

Syntax 1: CAN?(x)

Parameter: none

Response: *q1(y):* suppression of cyclical autocalibration

Syntax 2: CAL??(x)

Parameter: none

Response: *q1,q2(y)*

Possible stati for the suppression of cyclical autocalibration

CAL

Calibrate
Calibrate amplifier

Syntax: CAL(x)

Parameter: none

Note:

With all active amplifiers, this command triggers calibration that freezes measurement value updating for 1...3s.

Related commands: **ACL, CAL, CAN, CAP, CAV , CDT, CDV, EUN, GFV, ISV?, IUN, MVA?, OCP, SCL**

Amplifier type: **MC, MLxx**

CAV

Calibration value
Enter full scale

Syntax: CAVp1, p2(x)

Parameters: p1: insignificant
p2: full scale in user–defined unit

Effect: Use the CAV command for entering the full scale of the input characteristic. (Afterwards the CAP0 command must be transmitted!).

Related commands: **ACL, CAL, CAN, CAP, CAV, CDT, CDV, EUN, GFV, ISV?, IUN, MVA?, OCP, SCL**

Amplifier type: **MLxxB**

SCL

Shunt Calibration
Switch shunt on/off

Only with XM001

Syntax: SCLp1(x)

Parameter:

p1	Switching on shunt
0	Off
1	On

Related commands: **ACL, CAL, CAN, CAP, CAV , CDT, CDV, EUN, GFV, ISV?, IUN, MVA?, OCP, SCL**

Amplifier type: **MC, MLxx**

SCL?

Shunt Calibration Query
Output status of shunt switch setting

Syntax 1: SCL?(x)

Parameter: none

Response: *q1(y):*current status of shunt switch setting

Syntax 2: SCL??(x)

Parameter: none

Response: *q1,q2(y):* possible stati of shunt switch setting

CDV

Calibration Dead Load Value
Enter zero displacement of input (transducer) characteristic

Syntax: CDVp1,p2(x)

Parameter: p1: zero point value in displayed units

without p1: current measured value is set to p2

No parameters:
current measured value is set to the target value entered with CDT

Effect: additional zero point value which shifts the total characteristic curve.

Related commands: **ACL**, **CAL**, **CAN**, **CAP**, **CAV** , **CDT**, **CDV**, **EUN**, **GFV**, **ISV?**, **IUN**, **MVA?**, **OCP**, **SCL**

Amplifier type: **MLxx**

CDV?

Calibration Dead Load Value
Output zero displacement of input characteristic

Syntax 1: CDV?(x)

Parameter: none

Response: *q1(y):* current zero point value in displayed units

Syntax 2: CDV??(x)

Parameter: none

Response: *q1,q2(y)*: permissible range for zero point value in displayed units

CDT

Calibration Dead Load Target
Enter target value for zero displacement of input characteristic (for CDV command)

Syntax: CDTp1(x)

Parameter: p1: target value should be set to that of current measured value

Effect: value in displayed units, to which the amplifier is to be set using a remote control contact (RIF command) or the command CDV (no parameters). Factory setting 0.

Related commands: **ACL, CAL, CAN, CAP, CAV , CDT, CDV, EUN, GFV, ISV?, IUN, MVA?, OCP, SCL**

Amplifier type: **MLxx**

CDT?

Calibration Dead Load Target Query
Output target value for zero displacement of input characteristic (for CDV command)

Syntax: CDT?(x)

Parameter: none

Response: *q1(y):* target value to which the current measured value is set

CAP

Calibration Point
Input of transducer (input) characteristic points

Syntax: CAPp1,p2,p3(x)

Parameter: p1: point number (1 or 2)

p2: measurement signal (unit depending on amplifier), if no input value, then the current measured value is adopted

p3: display value

Special case: p1= 0: calibration values are active in the amplifier

Effect: the input characteristic curve is defined by 2 points. The input signal and associated display value must be entered for each point.

Example ML55: 0mV/V≙0%; 2mV/V≙100%

With the input CAP0(x), the calibration is executed. If the values lie within the permitted range, they are accepted into the calibration table and if not, an error message appears.

Related commands: **ACL**, **CAL**, **CAN**, **CAP**, **CAV** , **CDT**, **CDV**, **EUN**, **GFV**, **ISV?**, **IUN**, **MVA?**, **OCP**, **SCL**

Amplifier type: **MLxx**

CAP?

Calibration Point Query
Output of input (transducer) characteristic points

Syntax: CAP?<p1>(x)

Parameter: p1: number of point (1 or 2)

Response: *q1,q2,q3(y)*

q1: point number (1 or 2)

q2: measurement signal (unit depends on amplifier)

q3: value in displayed units

OCP

Output Calibration Point
Define analogue output characteristic points (setting up of analogue output)

Syntax: OCPp1,p2,p3(x)

Parameter: p1: point number (1 or 2)

p2: value in displayed units

p3: voltage at analogue output

Effect: the output characteristic is defined by 2 points. The output signal and associated display value must be entered for each point.

Example: 0%≙0V; 100%≙10V

With the input CAP0(x), the calibration is executed. If the values lie within the permitted range, they are accepted into the calibration table and if not, an error message appears.

Related commands: **ACL**, **CAL**, **CAN**, **CAP**, **CAV** , **CDT**, **CDV**, **EUN**, **GFV**, **ISV?**, **IUN**, **MVA?**, **OCP**, **SCL**

Amplifier type: **MLxx**

OCP?

Output Calibration Point Query
Output analogue output characteristic points (setting up of analogue output)

Syntax: OCP?p1(x)

Parameter: p1: number of point (1 or 2)

Response: *q1,q2,q3(y)*

q1: point number (1 or 2)

q2: value in displayed units

q3: voltage value

ISV?

Indication Scaling Values Query
Output display scaling values

Syntax: ISV?(x)

Parameter: none

Response: *q1,q2(y)*

q1: display zero point

q2: input range

Both values are output in floating format.

Displayed measured value = (raw binary value/7680000) x input range minus display zero point

Related commands: **ACL**, **CAL**, **CAN**, **CAP**, **CAV** , **CDT**, **CDV**, **EUN**, **GFV**, **ISV?**, **IUN**, **MVA?**, **OCP**, **SCL**

Amplifier type: **MLxx**

MVA?

Measuring Value Absolute Query
Output absolute measured value (gross)

Syntax: MVA?(x)

Parameter: none

Response: *q1(y)*
Current gross value (unit depends on amplifier), output in floating format

Related commands: **ACL**, **CAL**, **CAN**, **CAP**, **CAV** , **CDT**, **CDV**, **EUN**, **GFV**, **ISV?**, **IUN**, **MVA?**, **OCP**, **SCL**

Amplifier type: **MLxx**

EUN

Engineering Unit
Input of physical unit

Syntax: EUNp1(x)

Parameter: p1: ”Unit” (max. 4 characters)

Related commands: **ACL, CAL, CAN, CAP, CAV , CDT, CDV, EUN, GFV, ISV?, IUN, MVA?, OCP, SCL**

Amplifier type: **MLxx**

EUN?

Engineering Unit Query
Output of physical unit

Syntax: EUN?(x)

Parameter: none

Response: *q1(y):* ”Unit”

IUN

Indication Unit
Input of unit to be displayed

Syntax: IUNp1(x)

Parameter: p1: ”Unit” (max. 4 characters)

Related commands: **ACL, CAL, CAN, CAP, CAV , CDT, CDV, EUN, GFV, ISV?, IUN, MVA?, OCP, SCL**

Amplifier type: **MLxx**

IUN?

Indication Unit Query
Output of unit displayed

Syntax: IUN?p1(x)

Parameter:

p1	Unit
0	3 possible units
1	Unit (from EUN or IUN)

Response: *q1(y):* ”3 possible units” or ”Unit (from EUN or IUN)”

GFV

Gauge Factor Value
Input K-Factor

Syntax: GFVp1(x)

Parameter: p1: K-Factor

p1=0: No K-Factor

Related commands: **ACL**, **CAL**, **CAN**, **CAP**, **CAV** , **CDT**, **CDV**, **EUN**, **GFV**, **ISV?**, **IUN**, **MVA?**, **OCP**, **SCL**

Amplifier type: **MC, MLxx**

GFV?

Gauge Factor Value Query
Read off K-Factor

Syntax: GFV?(x)

Parameter: none

Response: *q1(y):* K-Factor

3.4 Tare

TAV

Tare value
Enter tare value

Syntax: TAVp1,p2(x)

Parameter: p1: tare value in displayed units (floating)

without p1: current measured value is set to p2

without parameters:
Current measured value is set to value entered with TAT. The input is possible from –109% to +109% of the input range. The tare value should only be set up after calibration.

Effect: tare value which displaces the total characteristic curve

Related commands: **TAR**, **TAT**, **TAV**

Amplifier type: **MLxx**

TAV?

Tara Value Query
Output tare value

Syntax 1: TAV?(x)

Parameter: none

Response: *q1(y)*: tare value in displayed units

Syntax 2: TAV??(x)

Parameter: none

Response: *q1,q2(y)*
Input range for tare value –109% to +109% of end value (floating)

TAT

Tare target
Enter target tare value (for TAV command)

Syntax: TATp1(x)

Parameter: p1: target value to which the current measured value is to be set (floating)

Effect: value in displayed units, to which the amplifier is to be set using a remote control contact (RIF command) or the command TAV (no parameters). Factory setting 0.

TIP: always set CDT and TAT to the same value.

Related commands: **TAR, TAT, TAV**
Amplifier type: **MLxx**

TAT?

Tare Target Query
Output target tare value (for TAV command)

Syntax: TAT?(x)

Parameter: none

Response: *q1(y):* target value, to which the current measured value is set (floating).

3.5 Analogue outputs

OSP

Output Signal Path
Define signal source (gross, net, peak values) for analogue outputs

Syntax: OSPp1,p2(x)

Parameter: p1: analogue output (1 or 2)

p2: signal (as per Tab.14) **valid only for MLxxB**
see 'Global Tables' 'Global Tables' (parameter list)

Related commands: **OPS, OSP, SAO**

Amplifier type: **MLxx**

OSP?

Output Signal Path Query
Output signal sources of analogue outputs

Syntax 1: OSP?p1(x)

Parameter: p1: analogue output (1 or 2)

Response: *q1,q2(y)*

q1: analogue output (1 or 2)

q2: signal as per Tab.14

Syntax 2: OSP??(x)

Parameter: none

Response: *q1,q2(y):* available analogue outputs (1 and 2)

Syntax 3: OSP?,?(x)

Parameter: none

Response: *q1,...,qn(y):* possible settings (see Tab.14)
see 'Global Tables' (parameter list)

See OCP and OCP? commands

SAO

Set Analogue Output
Define signal source (user-defined) for the analogue outputs

Syntax 1: SAOp1,p2(x)

Parameter: p1: analogue output (1 or 2)

p2: voltage of –10,000 to +10,000 (see Tab. 14)
see Global Tables (parameter list)

Related commands: **OPS, OSP, SAO**

Amplifier type: **MLxxB**

3.6 Peak value store

PSM

Peak Store Mode
Set up peak value store

Syntax: PSMp1,p2,p3(x)

Parameter: p1: peak value store (1 or 2)

p2	Input signal, mode
180	Minimum gross
181	Minimum net
182	Maximum gross
183	Maximum net

p3: time constant of the envelope curve function in seconds, set-up range from 0.01 to 10001; 0 means OFF.

Related commands: **CPV, HPV, MPV, MVC, PCM, PSM, PVS**

Amplifier type: **MLxx**

PSM?

Peak Store Mode Query
Output peak value store settings

Syntax 1: PSM?p1(x)

Parameter: p1: peak value store (1 or 2)

Response: *q1,q2,q3(y)*

q1: peak value store 1 or 2

q2: input signal, mode

q3: time constant of the envelope curve function in seconds

Syntax 2:	PSM??(x)
Parameter:	none
Response:	*q1,q2(y):* available peak value stores (1 and 2)
Syntax 3:	PSM?,?(x)
Parameter:	none
Response:	*q1..q4(y)*: possible input signals, modes (see PSM command)
Syntax 4:	PSM?,,?(x)
Parameter:	none
Response:	*q1,q2(y)* possible range of envelope curve decay time (0,0,10001.0)

CPV

Clear Peak Value
Clear peak value store

Syntax: CPVp1(x)

Parameter:

p1	Clears
with-out	Peak value stores 1 and 2
1	Peak value store 1
2	Peak value store 2

Note:
On clearing, peak value stores are set to the current measured value.

Related commands: **CPV**, **HPV**, **MPV**, **MVC**, **PCM**, **PSM**, **PVS**

Amplifier type: **MC, MLxx**

CPV?

Clear Peak Value Query
Which peak value stores can be cleared

Syntax: CPV??(x)

Parameter: none

Response: *q1,q2(y):* available peak value stores (1 and 2)

MPV

Mode Peak Value
Enter peak value store mode

Syntax: MPVp1(x)

Parameter: p1=1: peak value store 1 or peak value store 2
p2=1: current/instantaneous value
p2=0: peak value (Default)
Whenever you switch on, the mode is set to “peak value”.

Related commands: **CPV**, **HPV**, **MPV**, **MVC**, **PCM**, **PSM**, **PVS**

Amplifier type: **MLxx**

MPV?

Mode Value Query
Read out peak value store mode

Syntax1: MPV?p1(x)

Parameter: p1: peak value store 1 or 2

Response: *q1,q2(y):*
q1: peak value store 1 or 2
q2: 1: current/instantaneous value
0: peak value

Syntax 2: MPV??(x)
1,2(y): available peak value stores (1 or 2)

Syntax 3: MPV?,?(x)
0,1(y): possible modes (0= peak value,
1= current/instantaneous value)

The command returns the peak value store mode, which can be set by the CPV remote control functions or by the MPV command.

HPV

Hold Peak Value
Suspend/enable peak value store updating

Syntax: HPVp1,p2(x)

Parameter: p1: peak value store 1 or peak value store 2
p2=1: suspend updating
p2=0: enable updating (default)

Whenever you switch on, the status is set to “enable updating”.

Related commands: **CPV, HPV, MPV, MVC, PCM, PSM, PVS**
Amplifier type: **MLxx**

HPV?

Hold Value Query
Read out peak value store updating

Syntax1: HPV?p1(x)

Parameter: p1: peak value store 1 or 2

Response: *q1,q2(y):*

q1: peak value store 1 or 2

q2: 1: updating suspended
0: updating enabled

Syntax 2: HPV??(x)
1,2(y): available peak value stores (1 or 2)

Syntax 3: HPV?,?(x)
0,1(y): possible activation (0= enable, 1= suspend)

The command returns the peak value store status, which can be set by the HLD remote control functions or by the HPV command.

PCM

Peak Combine Mode
Combine peak value stores

Syntax: PCMp1(x)

Parameter:

p1	Type of combination of both peak value stores to a third peak value store
0	Not combined
186	Peak value store3=Peak value store1 – Peak value store2
187	Peak value store3 = (Peak value store1 + Peak value store2)/2 [Mean value]
188[1)]	Peak value store3 = integral gross signal [mean value]
189[1)]	Peak value store3 = integral net signal [mean value]

[1)] The mean-value calculation (integration) is possible with these amplifiers: ML01 (not in the case of linearisation), ML10, ML30, ML35 (not in the case of linearisation), ML50, ML51, ML55, ML55S6, ML60 (not in counting mode). It is not possible with the ML38.
These functions calculate the mean value from up to 4.29 billion measured values. The additions rate is defined via the low-pass filter that is set up and is 1200 Hz from 5 Hz Bessel or 10 Hz Butterworth. In the case of 0.05 Hz Bessel it is reduced to 18.75 Hz. In this way, mean-value calculations from 41 to 2651 days are possible. The reaction time on starting and stopping is approx. 2ms with fast low-pass filters and approx. 100ms in the case of 0.05 Hz Bessel. Mean-value calculation is controlled via a remote control contact – see RIF command,

Level 0V: Start mean-value calculation
Level 5V: Stop mean-value calculation

reversed logic in the case of AP12 and AP13

or the command MVCp1)x)

p1=1: Start mean-value calculation
p1=0: Stop mean-value calculation

The reaction time with an MVC command, irrespective of filtering, is approx. 100ms.

Related commands: **CPV, HPV, MPV, MVC, PCM, PSM, PVS**

Amplifier type: **MLxx**

PCM?

Peak Combine Mode Query
Output combination of peak value stores

Syntax 1:	PCM?(x)
Parameter:	none
Response:	*q1(y):* set combination
Syntax 2:	PCM??(x)
Parameter:	none
Response:	*q1,...,q5(y):* possible combinations (see PCM command)

MVC

Mean Value Control
Start or stop mean-value calculation (integration)

Syntax: MVCp1(x)

Parameter: p1=1: start mean-value calculation
p1=0: stop mean-value calculation

Effect: start or stop mean-value calculation (integration) selected with the PCM command.

Related commands: **CPV**, **HPV**, **MPV**, **MVC**, **PCM**, **PSM**, **PVS**
Amplifier type: **MLxx**

MVC?

Mean Value Control Query
Query status of mean-value calculation

Syntax: MVC?(x)

Parameter: none

Response: *q1(y):* status of mean-value calculation

3.7 Limit value monitoring

SOP

Set Output
Set limit switch outputs

Syntax: SOPp1,p2(x)

Parameter: p1: select the limit switch output (1...4)

p2: status of output (OFF =0 or ON=1)

Note:
This command is only active if the relevant limit switch monitoring is switched off.

Related commands: **LIV, LVD, LVL, LVS, SLC, SOP, LSS?**

Amplifier type: **MLxx**

SOP?

Set Output Query
Output of set limit switch outputs

Syntax 1: SOP?p1(x)

Parameter: p1: select limit switch output (1...4)

Response: *q1,q2(y)*

q1: number of limit switch output
q2: status, to which the output is set.

Syntax 2: SOP??(x)

Parameter: none

Response: q1,q2(y): available limit switch outputs (1...4)

Syntax 3: SOP?,?(x)

Parameter: none

Response: *q1,q2(y):* available stati OFF=0, ON=1

LVD

Limit Value Delay
Enter settling time for limit switch thresholds

Syntax: LVDp1,p2,p3,p4,p5,p6,p7,p8,p9(x)

Parameter: p1: time from 0 to 60000ms

p2...p9: limit switch thresholds which should be affected by the settling time

p2..p9	Limit value switching thresholds
200	No switching threshold
206	Activation threshold Limit1
207	Deactivation threshold Limit1
208	Activation threshold Limit2
209	Deactivation threshold Limit2
210	Activation threshold Limit3
211	Deactivation threshold Limit3
212	Activation threshold Limit4
213	Deactivation threshold Limit4

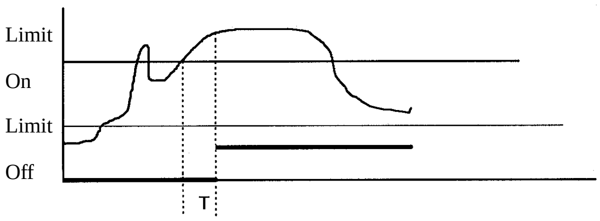
Note:
You can enter the thresholds in any order.

Example: LVD100,206,207(x)

0(y)

settling time 100ms for the activation and deactivation threshold of limit switch 1.

Related commands: LIV, LVD, LVL, LVS, SLC, SOP, LSS?
Amplifier type: MLxx



LVD?

Limit Value Delay Query
Read out settling time for limit switch thresholds

Syntax1:	LVD?(x)
Parameter:	none
Response:	*q1,q2,q3...,q9(y)*
	q1: settling time in ms
	q2...q9: limit switch thresholds, see LVD command
Syntax 2:	LVD??(x)
	0,60000(y)
	available input area for settling time
Syntax 3:	LVD?,?(x)
	200,206,207,208,209,210,21 1,212,213(y))
	Possible switching thresholds for limit switches

LVS

Limit Value Switch
Parameterise limit switches

Syntax: LVSp1,p2,p3,p4,p5,p6,p7(x)

Parameter: p1: numbers of limit switches (1...4)

p2: OPERATION (ON =1 or OFF=0)

p3	Input signal
214	Gross, dynamic
215	Net, dynamic
204	Peak value 1
205	Peak value 2
217	Combined peak value

p4	Direction
130	Above limit
131	Below limit

p5: level in displayed units (floating)

p6: hysteresis in displayed units (floating)

p7	Output logic
135	Positive logic
136	Negative logic

Related commands: **LIV**, **LVD**, **LVL**, **LVS**, **SLC**, **SOP**, **LSS?**

Amplifier type: **MLxx**

LVS?

Limit Value Switch Query
Output parameter assignment of limit switches

Syntax1: LVS?p1(x)

Parameter: p1: numbers of limit switches (1...4)

Response: *q1...q7(y)*

q1: number of limit switch (1...4)

q2: enabled status (ON=1, OFF=0)

q3: input signal

q4: operating direction (see LVS command)

q5: level in displayed units (floating)

q6: hysteresis in displayed units (floating)

q7: output logic (see LVS command)

Syntax 2: LVS??(x)

Parameter: none

Response: *q1,q2(y):* available limit switches (1..4)

Syntax 3: LVS?,?(x)

Parameter: none

Response: *q1, q2(y)*: possible enabled stati OFF=0 or ON=1

Syntax 4: LVS?,,?(x)

Parameter: none

Response: *q1,q2,q3,q4,q5(y):* available input signals (see LVS command)

Syntax 5:	LVS?,,,?(x)
Parameter:	none
Response:	*q1,q2(y):* possible operating direction (see LVS command)
Syntax 6:	LVS?,,,,?(x)
Parameter:	none
Response:	*q1,q2(y):* possible input range of level (floating)
Syntax 7:	LVS?,,,,,?(x)
Parameter:	none
Response:	*q1,q2(y):* possible input range of hysteresis (floating)
Syntax 8:	LVS?,,,,,,?(x)
Parameter:	none
Response:	*q1,q2(y):* possible output logic (see LVS command)

LVL

Limit Value Level
Input limit value level

Syntax: LVLp1,p2(x)

Parameter: p1: numbers of limit switches (1...4)

p2: level in displayed units (floating)

Related commands: **LIV**, **LVD**, **LVL**, **LVS**, **SLC**, **SOP**, **LSS?**

Amplifier type: **MLxx**

LVL?

Limit Value Level Query
Output limit value level

Syntax 1: LVL?p1(x)

Parameter: p1: numbers of limit switches (1...4)

Response: *q1,q2(y)*

q1: number of limit switch (1...4)

q2: level in displayed units (floating)

Syntax 2: LVL??(x)

Parameter: none

Response: *q1,q2(y):* available limit switches (1...4)

Syntax 3: LVL?,?(x)

Parameter: none

Response: *q1,q2(y):* possible input range of level (floating)

SLC

Set Logic Combination
Combination of limit value outputs

Syntax: SLCp1,p2,p3,p4,p5,p6(x)

Parameter:

p1	Limit value output for result of combination
0	Function switched off
1...4	Limit value output 1...4

p2	Combination parameter 1
300	Does not go into combination
301	Limit1
302	Limit1 inverted
309	Remote contact 1
310	Remote contact 1 inverted

p3	Combination parameter 2
300	Does not go into combination
303	Limit2
304	Limit2 inverted
311	Remote contact 2
312	Remote contact 2 inverted

p4	Combination parameter 3
300	Does not go into combination
305	Limit3
306	Limit3 inverted
313	Remote contact 3
314	Remote contact 3 inverted

Related commands: **LIV**, **LVD**, **LVL**, **LVS**, **SLC**, **SOP**, **LSS?**

Amplifier type: **MLxx**

p5	Combination parameter 4
300	Does not go into combination
307	Limit4
308	Limit4 inverted
315	Remote contact 4
316	Remote contact 4 inverted

p6	Combination type
151	AND
152	OR
153	EXOR
154	NAND
155	NOR
156	EXNOR

The type of combination applies to all combination parameters. You can never create more than one combination.

Example:

SLC3,301,304,313,300,152(x)

If one of the conditions, Limit1 or Limit2 inverted or remote contact 3 is satisfied, the result is stored in Limit3 and limit value output 3 switches to H signal (5V).

SLC?

Set Logic Combination
Output combination of limit value outputs

Syntax 1:	SLC?(x)
Parameter:	none
Response:	*q1,q2,q3,q4,q6(y):* combination currently set up
Syntax 2:	SLC??(x)
Parameter:	none
Response:	*q1, ... ,q5(y):* possible settings for limit value output
Syntax 3:	SLC?,?(x)
Parameter:	none
Response:	*q1, ... ,q5(y):* possible settings for combination parameter 1
Syntax 4:	SLC?,,?(x)
Parameter:	none
Response:	*q1, ... ,q5(y):* possible settings for a combination parameter 1
Syntax 5:	SLC?,,,?(x)
Parameter:	none
Response:	*q1, ... ,q5(y):* possible settings for combination parameter 3
Syntax 6:	SLC?,,,,?(x)
Parameter:	none
Response:	*q1, ... ,q5(y):* possible settings for combination parameter 4
Syntax 7:	SLC?,,,,,?(x)
Parameter:	none
Response:	*q1, ... ,q6(y):* possible settings for combination type

LSS?

Limit Switch Status Query
Query limit value status

Syntax:	LSS?(x)	
Parameter:	p1:	LIV1–Status OFF or ON: 0 or 1
p2:		LIV2–Status OFF or ON: 0 or 1
p3:		LIV3–Status OFF or ON: 0 or 1
	p4:	LIV4–Status OFF or ON: 0 or 1

Related commands: **LIV**, **LVD**, **LVL**, **LVS**, **SLC**, **SOP**, **LSS?**
Amplifier type: **MLxxB**

3.8 Transfer of amplifier settings and comment

TDD

Transmit Device Data
Save or read amplifier settings and comment

Syntax: TDDp1,p2(x)

Parameter:

p1	Amplifier settings
0	Load factory setup
1	Load parameter set (Recall Data)
2	Save parameter set (Save Data)
3	Read external parameter set (Recall Data Ext)
4	Save external parameter set (Save Data Ext)
5	Load comment
6	Save comment
7	Load external comment (Recall Comment Ext)
8	Save external comment (Save Comment Ext)

p2	Parameter set
1...8	Required parameter set (only in the case of p1= 1,2,5 and 6)
9	External EEPROM (only in the case of p1=3, 4, 7 and 8)

The active parameter set always stays in the RAM. The amplifier continuously remembers the last input of p2, so that a subsequent TDD command without p2 works on the parameter set selected previously. After a set-up this parameter is set to 1.

Related commands: **MDD**, **TDD**, **UCC**

Amplifier type: **MC, MLxx**

TDD?

Transmit Device Data Query
Query, where amplifier set-up comes from

Syntax: TDD?p1(x)

Parameter:

p1	Amplifier setting
0	Source of amplifier set-up data
1	Active parameter set
3	Test, whether XM001 is present

Response: *q1(y):* in the case of p1=0

q1	Source of amplifier set-up data
0	Factory setup
1	Parameter set (Recall)
2	User

q1(y): in the case of p1=1

q1	Active parameter set
0	Parameter set 1...8

q1(y): in the case of p1=3

q1	Test, whether XM001 is present
0	XM001 present
–1	XM001 not present

UCC

User Channel Comment
Input comment

Syntax: UCCp1(x)

Parameter: p1: any string ”_____”, max. 45 characters

Related commands: **MDD, TDD, UCC**

Amplifier type: **MC, MLxx**

UCC?

User Channel Comment Query
Output comment

Syntax: UCC?(x)

Parameter: none

Response: *”__(String)__”(y):* stored string, 45 characters

Amplifier type: **MC, MLxx**

3.9 Remote control

LOR

Local Remote
Local/Remote switching

Syntax: LORp1(x)

Parameter:

p1	Status
0	REMOTE
1	LOCAL

Effect: activates/deactivates remote control contacts (1...8).

Related commands: **LOR, RFP, RIF, RIP?, DFL**

Amplifier type: **MC, MLxx**

LOR?

Local Remote
Output Local/Remote status

Syntax 1: LOR?(x)

Parameter: none

Response: *q1(y):* current status

Syntax 2: LOR??(x)

Parameter: none

Response: *q1,q2(y):* possible stati (see LOR command)

RIF

Remote Input Function
Assign remote control contact functions

Syntax: RIFp1,p2(x)

Parameter: p1: number of remote control contacts (1...8)

p2	Function
50	No function
51	Autocalibration On/Off
52	Tare
53	Current value/Peak value1
54	Hold Peak value1
55	Current value/Peak value2
56	Hold Peak value2
57	Zeroing amplifier
58	Shunt On/Off
59	Print On/Off
60	Switch on calibration signal
61	Switch on zero signal
62	Operating sign change
63	Select parameter set – Bit1
64	Select parameter set – Bit2
65	Select parameter set – Bit4
66	Remote/Local
67	Start/Stop Integration

Note:
Remote contacts are activated/deactivated with the LOR command.

Related commands: **LOR, RFP, RIF, RIP?, DFL**
Amplifier type: **MLxx**

RIF?

Remote Input Function Query
Output allocation of remote control contacts

Syntax 1: RIF?p1(x)

Parameter:

p1	Remotes
1...8	Number of remote 1...8

Response: *q1,q2(y)*
q1: number of remote contact (1...8)
q2: function (see RIF command)

Syntax 2: RIF??(x)

Parameter: none

Response: *q1,q2(y):* available remote contacts (1...8)

Syntax 3: RIF?,?(x)

Parameter: none

Response: *q1,...,qn(y):* available functions (see RIF command)

RIP?

Read Input
Output status of remote contacts

Syntax:	RIP?(x)	
Parameter:	none	
Response	*q1(y):*	status of the remote control contact inputs (8 bit binary coded, at q1=255 all inputs to ”H”).

Note:

This command is not affected by the local/remote switch setting.

Related commands: **LOR, RFP, RIF, RIP?, DFL**

Amplifier type: **MLxx**

DFL

Direct Function Lock
Disable function key

Syntax: DFLp1,(x)

Parameter:

p1	
0 bits	Zeroing function using function key AB22, remote contact and command CDV without first parameter, disabled
1 bits	Taring function using function key AB22, remote contact and command TAV without first parameter, disabled

Related commands: **LOR**, **RFP**, **RIF**, **RIP?**, **DFL**

Amplifier type: **MLxxB**

DFL?

Direct Function Lock Query
Output status of the function key

Syntax: DFL?(x)

Parameter: p1

Response *q1(y):* **0 bits**

q2(y): **1 bits**

4 Amplifier functions

Note:
Amplifier function commands only have an effect on the selected channels (see PCS command). **Whenever you switch on, all channels are selected (i.e. active).**

4.1 Output format, measurement output

ISR

Input Sampling Rate
Specify measurement transfer rate

Syntax: ISR p1(x)

Parameter: p1

Divisor for the data transfer rate in the case of measurement output via an interface.

The display of measured values on the AB22A/AB32 operator panel must be switched off (see SIS command).

The relationship between the parameter p1 and the output rate comes from the calculation below.

Note:

With B amplifiers, the ISR command no longer affects the sample times for peak value stores and limit-value monitoring. The sample times are only defined by the filter frequency.

The default value when you switch on is always 1.

The following calculation applies when establishing the input value for p1:

$$p1 = \frac{\text{Internal cycle frequency}}{\text{Measurement transfer rate}}$$

The internal cycle frequency depends on the low pass cut-off frequency selected:

Low pass cut-off frequency (Hz)	**Internal cycle frequency (Hz)**	
from 1.25	600	
0.5	300	150*
0.2	75	75*
0.1	37.5	37.5*
0.05	18.75	18.75*

* when linearising thermocouples or resistance thermometers

The value for the measurement transfer rate depends on the number of amplifier channels selected with the CHS command. The following maximum values must not be exceeded:

Number of amplifiers	**max. Measurement transfer rate**	
	4 byte format	2 byte format
1	150	200
2	50	55
3	38	50
4	35	45
5	29	40
6	26	35
7	22	31
8	20	29
9	18	26
10	17	24
11	15	22
12	14	21

Example:
Measurement transfer rates in the case of a low pass cut-off frequency greater than 1.25Hz and binary measurement output in 4 byte format.

ISR value	Number of amplifiers											
	1	2	3	4	5	6	7	8	9	10	11	12
60	10	10	10	10	10	10	10	10	10	10	10	10
40	15	15	15	15	15	15	15	15	15	15		
30	20	20	20	20	20	20	20	20				
25	24	24	24	24	24	24						
20	30	30	30	30								
15	40	40										
1..4	150*											

* Maximum measurement transfer rate for one channel.

The following commands start an output of 6 channels in a time raster of 20 samples/s:

Command	Function
PCS1,2,3,4,5,6(x)	Select channels 1...6
SIS1,200(x)	Switch off AB22A/AB32 display
ISR30(x)	Measurement transfer rate of 20 per second
MVF1253(x)	Output format 4 byte binary LSB..MSB
RMV?214,0(x)	Continues measurement output gross signal, dynamic
STP(x)	Stop measurement output

Related commands: **CDW**, **CMR**, **COF**, **IMR**, **ISR**, **MSV?**, **RMV?**, **STP**, **TEX**

Amplifier type: **MC**

ISR?

Input Sampling Rate Query
Output of measurement transfer rate

Syntax: ISR?(x)

Parameter: none

Response: *q1(y):* measurement transfer rate (see ISR command table)

STP

Stop
Terminate measurement output and data acquisition

Syntax: STP(x)

Parameter: none

Response: none

Related commands: **CDW**, **CMR**, **COF**, **IMR**, **ISR**, **MSV?**, **RMV?**, **STP**, **TEX**

Communications processor: **CP12, CP32**

RMV?

Read Measurement Value
Output of measured value

Syntax: RMV?p1,p2,p3(x)

Parameter:
- p1: signal (after Tab.14, apart from values 200, 201, 216)
- p2: number of measured values 1...65535; 0=Send continuously
- p3: sequence time in seconds (0.1s to 60.0s). Output time in seconds between measured values (only in the case of binary output).

Effect: the measured value of the required signal p1 is output. The output format depends on the last MVF and TEX command. You use the PCS command to specify from which channel the measured values are to be output.

Response: *measured value(y):*output format see MVF command.

Note:

This command is identical to the MSV? command right up to signal encoding (p1).

Example: see MSV? command

Related commands: **CDW, CMR, COF, IMR, ISR, MSV?, RMV?, STP, TEX**

Related commands: **ADS, CFS, CGP, CMF, DEL, DIR, DSD, EES?, HCF?, ICR, IHD, IPA, LPR, MBF, MCS, MRG, MSS, MVF, OMP, PCS, PHD, PSC, RHD?, RMB?, RMS?, RMV?, SBR, SCM, SHD, SIS, SML, SMS, SNO?, SPP, SPS, SRP, STD, TCD, TRD, TRE, TRG, TRR, TSV**

Communications processor: **CP32**

MVF

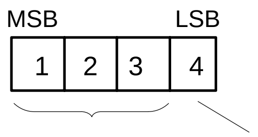
Measuring Value Format
Specify measurement output format (for RMV? command)

Syntax: MVF p1(x)

Parameter:

p1	Measurement output format
1250	Measured value, channel, status (ASCII format)
1251	Measured value (ASCII format)
1252	4 byte binary measurement output (MSB XX XX LSB)
1253	4 byte binary measurement output (LSB XX XX MSB)
1254	2 byte binary measurement output (MSB LSB)
1255	2 byte binary measurement output (LSB MSB)

4 byte output:



3 byte measured value   1 byte status

7 680 000 ADC units = full scale value (unit)

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

Communications processor: **CP32**

Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Status byte

1	Limit switch 1 On
2	Limit switch 2 On
4	Limit switch 3 On
8	Limit switch 4 On
16	Gross overflow (S1)
32	Net overflow (S2)
64	Calibration error
128	Setup modification (e.g. using remote control sig Value not available

2 byte output

MSB	LSB
1	2

2 byte measured value

30000 ADC units = full scale value (unit)

Note:
This command always applies to all channels of a device.

Note:

With ASCII format, the measured values are output scaled, with binary format in ADC units.

MVF?

Measuring Value Format Query
Query measurement output format

Syntax: MVF?(x)

Parameter: none

Response: *q1(y):* code number of output format (see MVF command).

TEX

Define Terminator
Define measured value separator

Syntax: TEX p1,p2(x)

Parameter:

p1	Parameter separator
	1 - 126 Default: 44, ASCII ”,”

p2	Block separator
	1 - 126 Default: 13, ASCII ”CR”

Related commands: **CDW, CMR, COF, IMR, ISR, MSV?, RMV?, STP, TEX**

Communications processor: **CP12, CP32**

TEX?

Define Terminator Query
Output measured value separator

Syntax: TEX?(x)

Parameter: none

Response: *q1,q2(y):* parameter separator, block separator

4.2 Display functions

ICS

2.Indication Set-up
Enter display adaptation (decimal places, step)

Syntax:	ICSp1,p2(x)
Parameter:	p1: number of decimal places (0...6)
	p2: step (as per Tab.8)

Related commands: **ENU**, **IAD**, **ICS**, **ISS**, **SIS**, **STL**
Amplifier type: **MLxx**

ICS?

3.Indication Set-up Query
Output display adaptation (decimal places, step)

Syntax 1:	ICS?(x)
Parameter:	none
Response:	*q1,q2(y)*
	q1: number of decimal places (0..6)
	q2: step (as per Tab.8)
Syntax 2:	ICS??(x)
Parameter:	none
Response:	*q1,q2(y):* possible decimal places (0...6)
Syntax 3:	ICS?,?(x)
Parameter:	none
Response:	*q1,q2(y):* possible steps as per Tab.8

SIS

Select Indication Signal
Selection of channel and signal display on AB22A/AB32

Syntax: SIS p1, p2, p3 (x)

Parameter: p1: channel number (1...16)
p2: signal code number

p2	Signal that is to be displayed
200	no measured value
202	Gross (S1)
203	Net (S2)
204	Peak value 1 (S3)
205	Peak value 2(S4)
206*	Limit switch 1 (Limit1)
208*	Limit switch 2 (Limit2)
210*	Limit switch 3 (Limit3)
212*	Limit switch 4 (Limit4)

*) Only if limit switches are on

p3: subchannel

Related commands: **ENU**, **IAD**, **ICS**, **ISS**, **SIS**, **STL**

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

Communications processor: **CP32**

SIS?

Select Indication Signal Query
Output of channel and signal display on AB22A/AB32

Syntax: SIS?(x)

Parameter: none

Response: *q1,q2,q3(y):*
Channel number, signal code number, subchannel

STL

Status LED
Selection of status/level display

Syntax: STLp1(x)

Parameter:

p1	Display status
0	Level display
1	Status display

Related commands: **ENU, IAD, ICS, ISS, SIS, STL**

Amplifier type: **MC, MLxx**

STL?

Status LED Query
Output status/level display

Syntax 1: STL?(x)

Parameter: none

Response: *q1(y):* current status of display

Syntax 2: STL??(x)

Parameter: none

Response: *q1,q2(y):* possible status of displays

5 Print functions

PRF

Print Format
Define print format

Syntax: PRFp1,p2,p3,p4,p5(x)

Parameter:

p1..p5	Signals that are to be printed
200	No signal
202	Gross (S1)
203	Net (S2)
204	Peak value 1 (S3)
205	Peak value 2 (S4)
216	Limit switches (status)
217	Combined peak value (S5)
No parameters	Print function off

A maximum of five signals can be printed. The printing sequence is always:

Gross, net, peak value 1, peak value 2, combined peak value, limit value status, regardless of the input sequence.

Note:

The set-up affects the printed output via the AB22A/AB32 as well as the LPR command.

Related commands: **LPR**, **PFS**, **PRF**, **PRT**, **SPP**

Amplifier type: **MLxx**

PRF?

Print Format Select Query
Query the print format

Syntax 1:	PRF?(x)
Parameter:	none
Response:	*q1,q2,q3,q4,q5(y):* signals that are to be printed
Syntax 2:	PRF??(x)
Parameter:	none
Response:	*q1, ... ,qn(y)* Signals that are available to be printed (p1 ...p5)

SPP

Set Printer Port
Define the output interface

Syntax: SPPp1(x)

Parameter:

p1	Interface
1301	RS232
1302	RS485
1306	Printer (Centronics)

Related commands: **LPR, PFS, PRF, PRT, SPP**

Related commands: **ADS, CFS, CGP, CMF, DEL, DIR, DSD, EES?, HCF?, ICR, IHD, IPA, LPR, MBF, MCS, MRG, MSS, MVF, OMP, PCS, PHD, PSC, RHD?, RMB?, RMS?, RMV?, SBR, SCM, SHD, SIS, SML, SMS, SNO?, SPP, SPS, SRP, STD, TCD, TRD, TRE, TRG, TRR, TSV**

Communications processor: **CP12, CP32**

SPP?

Set Printer Port Query
Determine the output interface

Syntax: SPP?(x)

Parameter: none

Response: *q1(y):* output interface (printer port – see command SPP)

LPR

Line Print
Print measured values

Syntax: LPRp1(x)

Parameter:

p1	Signals that are to be printed
202	Gross (S1)
203	Net (S2)
204	Peak value 1 (S3)
205	Peak value 2 (S4)
216	Limit switches (status)

If p1 is not sent, the signals which are output are those that were defined with the aid of command PRF or via the AB22A/AB32 menu ”Print, Select”.

Related commands: **LPR, PFS, PRF, PRT, SPP**

Communications processor: **CP32**

6 Fast data acquisition

The following commands are supported only by the CP32B.

MCS

Measuring Channel Select
Channel selection for the channels that are to be recorded

Syntax: MCSp1,...,p18(x)

Parameter: p1,...,p18: channel numbers (1...16,17 and 18=time stamp)

MCS(x) re-releases reserved resources

Related commands: **ADS, CFS, CGP, CMF, DEL, DIR, DSD, EES?, HCF?, ICR, IHD, IPA, LPR, MBF, MCS, MRG, MSS, MVF, OMP, PCS, PHD, PSC, RHD?, RMB?, RMS?, RMV?, SBR, SCM, SHD, SIS, SML, SMS, SNO?, SPP, SPS, SRP, STD, TCD, TRD, TRE, TRG, TRR, TSV**

Communications processor: **CP32**

MCS?

Measuring Channel Select Query
Output the channel selection for the channels to be recorded

Syntax: MCS?p1(x)

Parameter: p1: output mode

0: existing ML channels
1: active channels

Response: *q1,...,q18(y):*

List of the existing or active ML channels

MCS cannot be used for selecting during data acquisition.

Channels 17 and 18 are the internal time stamps for the CP32. They are two 24-bit counters, incremented at 76.8kHz. With measurement output in floating format the time stamps are scaled in seconds. When a recording is made to the hard disk, the scaling factors contained in the file header are specified.

Channel 18 extends the time stamp from 24 to 48 bits. It counts the overflows from channel 17. The time stamps can only be selected together, that is, both must always be recorded.

With effect from Version 1.40, the MCS command will reserve the required link resources. It is possible, in this type of situation, that an ML77 will not be able to measure all the channels which have been assigned parameters, as they can no longer be downloaded by using the links. The MCS command can fail as well, as an ML77 has reserved other resources. An external PC should re-release the resources when exiting communication (MSC(x)).

SMS

Subchannel Measurement Select
Set subchannel mask

Syntax: SMS p1,..., p128 (x)

Parameter: p1,...,p128

1,...128 subchannel selection

SMS 0 (x) selects all the existing subchannels

This command sets the subchannel selection mask for display. The modules to be set should already be selected with PCS.

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

Communications processor: **CP32**

SMS?

Subchannel Measurement Select Query
Query subchannel mask

Syntax: SMS?p1(x)

Parameter: p1: output mode

0 existing subchannels
1 selected subchannels

Response: *q1,..., q128*
List of existing or selected ML channels

TRE

Trigger Event
Set up the trigger conditions

Syntax TREp1,...,p8(x)

Parameter: p1: number of the trigger event
1...4: start trigger events
5...8: stop trigger events
9: measuring rate trigger Start Event
10: measuring rates Stop Event

p2: channel
1...16: amplifier channel

p3: amplifier signal (see MSS command)

p4:

p4	Trigger event/Mode
0	none (off)
6006	Measured value threshold
6007	Measured value window
6008	Limit switch1
6009	Limit switch2
6010	Limit switch3
6011	Limit switch4
6102	External trigger (Remote 7)

p5:

p5	Edge
6004	Static positive ($\geq$ or within)
6005	Static negative ($\leq$ or outside)
6002	Dynamic positive
6003	Dynamic negative

p6: threshold value 1

p7: threshold value 2

Threshold value Define physical quantity in float p8: subchannel

If p8 is omitted, subchannel 1 is used.

Note: Triggering can only occur on an amplifier channel which is also being used for data acquisition (see MCS command).

In the case of ”External Trigger” (6102) and ”Static Negative” (6005), measurement and/or recording only continues so long as remote contact 7 is connected with 0V (24V for the AP12 and AP13) and until the number of measurements is reached (TSV command, parameter p3).

In the case of ”External Trigger” (6102) and ”Dynamic Positive” (6002), each positive signal edge at remote contact 7 (negative edge for the AP12 and AP13) causes a measurement run to be started and/or recorded. The number of pulses must correspond to the number of measurements (TSV command, parameter p3).

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

Communications processor: **CP32**

TRE?

Trigger Event Query
Output the trigger conditions

Syntax: TRE?p1(x)

Parameter: p1: trigger event (1...10)

0: separated by ' : ' , all events are output where mode ≠ 0. Inactive events are suppressed. If no events are active, the response is '0'.

1,...,10: the selected event is output.

Response: *q1,...,q7: q1,...,q7(y):* p1=0

q1,...,q7(y): p1=1...10

q1: number of the trigger event (1...10)

q2: channel number (1...16)

q3: amplifier signal

q4: mode

q5: edge

q6: threshold value 1

q7: threshold value 2

q8: subchannel

Note:

The threshold values are specified in float (ASCII string) as a physical quantity. In the case of window triggers, threshold value 1 always indicates the mathematically smaller threshold value. If trigger events are defined that refer to channels or signals which are off, they have no effect. If channels or signals of this kind are included later in the data acquisition session, these trigger events take effect.

TRR

Trigger Remove
Delete the trigger conditions

Syntax: TRR(x)

Parameter: none

Response: *q1(y)*

0: command successfully executed

1: error occurred

Related commands: **ADS, CFS, CGP, CMF, DEL, DIR, DSD, EES?, HCF?, ICR, IHD, IPA, LPR, MBF, MCS, MRG, MSS, MVF, OMP, PCS, PHD, PSC, RHD?, RMB?, RMS?, RMV?, SBR, SCM, SHD, SIS, SML, SMS, SNO?, SPP, SPS, SRP, STD, TCD, TRD, TRE, TRG, TRR, TSV**

Communications processor: **CP32**

MSS

Measuring Signal Select
Measuring signal selection for the channels to be recorded

Syntax: MSS p1, p2, p3, p4, p5, p6 (x)

Parameter:

p1...p6	Signal that is to be recorded
214	Gross, dynamic
215	Net, dynamic
204	Peak value 1
205	Peak value 2
217	Combined peak value
289	Status of the remote contacts

Note:

Up to a measuring rate of 2400Hz (see ICR command), all 4 signals can be recorded, but only 1 can be recorded at higher sampling rates.

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

Communications processor: **CP32**

MSS?

Measuring Signal Select Query
Output the measurement signal selection for the channels to be recorded

Syntax: MSS?(x)

Parameter: none

Response: list[i]: list[j]:...: list [k]
list [x]
q1,...,q6(y) signals of a channel

TRD

Trigger Definition
Define the trigger-event links

Syntax: TRDp1,p2,p3(x)

Parameter:

p1	Trigger machine
6500	Start trigger machine
6501	Stop trigger machine
6502	Measuring rate trigger machine

p2	Trigger mode
0	Never
151	AND (for type 6502 On, since only one event)
152	OR (for type 6502 On, since only one event)
6100	Immediate (trigger is immediately active)

p3	Number
0,.., N	Only analysed for the measurement rate trigger definition. If a measuring rate trigger is initiated, the "Number" of measured values is recorded at measuring rate 2 (ICR command), before the measuring rate trigger machine re-checks the measuring rate stop trigger event.

Related commands: **ADS, CFS, CGP, CMF, DEL, DIR, DSD, EES?, HCF?, ICR, IHD, IPA, LPR, MBF, MCS, MRG, MSS, MVF, OMP, PCS, PHD, PSC, RHD?, RMB?, RMS?, RMV?, SBR, SCM, SHD, SIS, SML, SMS, SNO?, SPP, SPS, SRP, STD, TCD, TRD, TRE, TRG, TRR, TSV**

Communications processor: **CP32**

TRD?

Trigger Definition Query
Output the trigger-event links

Syntax: TRD?p1(x)

Parameter:

p1	Trigger link
0	Setup for all trigger machines
6500	Setup for the start trigger machine
6501	Setup for the stop trigger machine
6502	Setup for the measuring rate trigger machine

Response: *q1,q2,q3:q1,q2,q3:q1,q2,q3(y):* p1=0

q2,q3(y): p1=6500,6501 oder 6502

q1: Trigger machine

q2: Trigger mode

q3: Number

TSV

Trigger Setup Value
Define and start the triggered measurement

Syntax: TSVp1,p2,p3(x)

Parameters
- p1: number of measured value lines
  - 1,...,n: number of measured value lines in a measurement session consisting of a pre-trigger and a post-trigger section
- p2: pre-trigger measured value lines
  - 1,...,n: number of measured value lines in the pre-trigger section
- p3: number of measurements
  - 1,...,n: number of consecutive measurements
  - 0: an infinite number of consecutive measurements (limited only by the size of the system memory)

Effect: the command sets the parameters and starts data acquisition.

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

Communications processor: **CP32**

TSV?

Trigger Setup Value Query
Output information on started or recorded measurement

Syntax:	TSV?(x)
Parameters	none
Response:	*q1,q2,q3(y)*

q1: number of measurements already recorded
q2: number of measured value lines already recorded in started measurement session
q3: trigger status of the started measurement session
0: write pre-trigger
q4: number of recorded measurements including the measurements already read out.

MRG

Measurement Rate Group
Define the measurement rate groups

This command assigns a measurement rate group to a selected channel or subchannel. Up to 3 synchronous and one asynchronous measurement rate groups are supported. The measured values of the various groups are stored in separate FIFO buffers and must be read out separately over the interface. When saving on the PCMCIA hard disk, the data of the various groups is saved in separate files.

Syntax: MRG p1 (x)

Parameter: p1: 0..3 measurement rate group

0..2 synchronous measurement rate groups

3 asynchronous measurement rate groups

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

Communications processor: **CP32**

MRG?

Measurement Rate Group Query
Output the measurement rate groups

Syntax: MRG? (x)

Response: *q1,q2(y)*

q1: measurement rate group

q2: position of the measured value of this channel in the output stream

SRP

Set Recording Parameters
Define data acquisition parameters

Syntax SRPp1,...,p7(x)

Parameter:

- p1: 1...N Number of measured value lines in a single measurement

  Pre-trigger and post-trigger
- p2: 1...N Number of pre-trigger measured value lines (post-trigger can be 0)
- p3: 1...N Number of consecutive measurements

  0: an infinite number of consecutive measurements
- p4: status of output redirection to hard disk

  0: output redirection to HD inactive

  1: output redirection to HD active
- p5: name of measurement file on the hard disk
- p6: number of measured values for output job on the hard disk
- p7: output mode for hard disk data acquisition (only 6409 reasonable, see RMB command)

Effect: this command defines the parameters (TSV, IHD and RMB? commands) for triggered data acquisition without starting the measurement.

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

Communications processor: **CP32**

SRP?

Set Recording Parameters Query
Read out data acquisition parameters

Syntax: SRP?(x)

Parameters none

Response: *q1,q2,....q7(y):* parameters set with SRP

OMP

Output Measuring Pointer
Position the reading pointer

Syntax: OMP p1, p2, p3 (x)

Parameter:

p1	Output position
6401	from the start of the single measurement actually addressed
6403	from the start of the next single measurement; release previous measurements
6404	from the trigger point of the single measurement actually addressed
6405	from the end of the single measurement actually addressed
6406	from the current reading pointer position

p2: –N....N Long offset relevant to mode (measured value lines)

p3: 0,...,2 Measurement rate group
3 Asynchronous FIFO

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

Communications processor: **CP32**

OMP?

Output Measuring Pointer Query
Output the reading pointer information

Syntax: OMP?p1(x)

Parameter: p1: 0,...,2 Measurement rate group
3 Asynchronous FIFO

Response: *q1,q2,q3,q4,q5(y)*

q1: position in the single measurement actually addressed

q2: number of measured value lines already recorded in the currently started measurement

q3: absolute position in all recorded measurements

q4: number of measured value lines in all recorded measurements

q5: data recording status

0: data acquisition not running
1: data acquisition running

If the measurement rate group is not specified, the measurement rate group 0 is addressed.

RMB?

Read Measuring Buffer
Output the measurement data

Syntax: RMB?p1,p2,p3(x)

Parameters p1: number of measured value lines to be output

p2	Mode
6400	from the start of the whole measured value memory
6406	from the current reading pointer position
6407	from the current reading pointer; release everything prior
6408	read newest values (reading pointer is unchanged)
6409	read from the current reading pointer; release everything including the block that has been read

p3: 0,...,2 Measurement rate group
3 Asynchronous FIFO

If the measurement rate group is not specified, the measurement rate group 0 is addressed.

Response: *#0 measured values (y):* for output format see MBF command

Note:

In the case of floating formats, a gross overflow is coded as 2e20, a net overflow as 3e20 and a calibration error as 4e20. In the case of the time channel, when the output is in binary format the measured value (ticks) must be multiplied by 1/measuring rate. Output in floating format is expressed in ms in the case of measurement rates up to 1Hz, and in seconds in the case of measurement rates from 2s upwards.

Example: see the MSV? command, output in binary format

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR‹**, **IHD**, **IPA**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF‹**, **OMP**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

Communications processor: **CP32**

MBF

Measuring Output Format
Define the measurement output format (for the RMB? command)

Syntax: MBFp1, p2 (x)

Parameter:

p1	Output format
1252	4 bytes binary MSB...LSB
1253*)	4 bytes binary LSB...MSB
1254	2 bytes binary MSB LSB
1255	2 bytes binary LSB MSB
1256	4 bytes floating MSB...LSB
1257	4 byte floating LSB...MSB

*) Default

p2: 0,...,2 Measurement rate group

If parameter p2 is omitted, the setting affects all measurement groups.

Note:

ADC units are used when scaling data in floating format and outputting data in binary format (7 680 000 ADC units correspond to the full scale value).

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

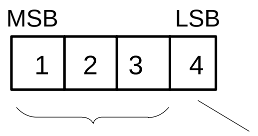
Communications processor: **CP32**

MBF?

Measuring Buffer Format Query
Query measurement output format

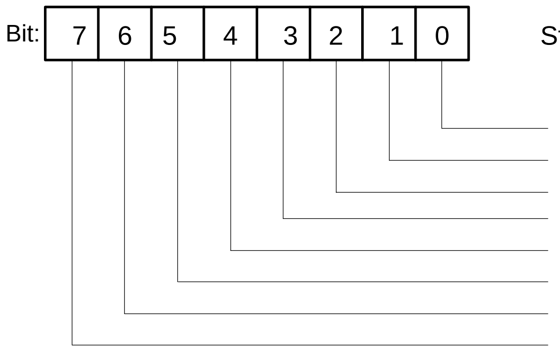
Syntax:	MBF?p1(x)
Parameter:	p1: 0,...,2 Measurement rate group
Response:	*q1(y)*
	q1:Output format
	If parameter p1 is omitted, you are given the output format of measurement rate group 0.

4 byte binary output:



3 byte measured value 1 byte status

7 680 000 ADC units = full scale value (unit)



Status byte (default)

1 Limit switch 1 On
2 Limit switch 2 On
4 Limit switch 3 On
8 Limit switch 4 On
16 Gross overflow (S1)
32 Net overflow (S2)
64 Calibration error
128 Setup modification
(e.g. using remote control signals),
Value not available

Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Status byte (ext. triggering)

1	Limit switch 1 On
2	Limit switch 2 On
4	Limit switch 3 On
8	Limit switch 4 On
16	Ext. trigger (remote contact 7)
32	Ext. trigger (remote contact 5)
64	Measured value identification bit 0
128	Measured value identification bit 1

0: Gross
1: Net
2: Peak value 1
3: Peak value 2

2 byte binary output (without status byte):

MSB	LSB
1	2

2 byte measured value

30 000 ADC units = full scale value (unit)

4 byte floating format (without status byte):

MSB			LSB
1	2	3	4

4 byte measured value
Measured values are scaled for output

ICR

Internal Channel Recording rate
Set up the measurement rates

Syntax: ICR p1, p2, p3 (x)

Parameter: p1: measuring rate 1 (in accordance with Tab.27)
p2: measuring rate 2 (in accordance with Tab.27)
p3: 0, ...,2 Measurement rate group

Effect: sets up two measurement rates, with toggling between them in the event of a defined measuring rate trigger event (see TRE command).

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

Communications processor: **CP32**

ICR?

Internal Channel Recording Rate Query
Output the measurement rate setting

Syntax: ICR?p1,p2(x)

Parameter: p1 0,1 Mode
p2: 0,...,2 Measurement rate group

Response: *q1,q2(y):*

p1=0; Measuring rate 1, Measuring rate 2

q1(y): p1=1; current measuring rate

Note:
In the case of the query command, if parameter p1 is omitted, the command works in the same way as p1=0. This provides compatibility with older versions. If parameter p3 or p2 (query) is omitted, the command affects measurement rate group 0.

RMS?

Read Master Switch
Position of the master/slave switch on the CP32B communications processor

Syntax: RMS?(x)

Parameter: none

Response: *q1(y):* position of the Master/Slave switch (1=Master, 0=Slave)

Related commands: **ADS, CFS, CGP, CMF, DEL, DIR, DSD, EES?, HCF?, ICR, IHD, IPA, LPR, MBF, MCS, MRG, MSS, MVF, OMP, PCS, PHD, PSC, RHD?, RMB?, RMS?, RMV?, SBR, SCM, SHD, SIS, SML, SMS, SNO?, SPP, SPS, SRP, STD, TCD, TRD, TRE, TRG, TRR, TSV**

Communications processor: **CP32**

TCD

Transmit Configuration Data
Save or read CP32B interface configuration

Syntax: TCD p1 (x)

Parameter:

p1	CP32B configuration
0	Load factory setup
1	Load configuration (Recall Data)
2	Save configuration (Save Data)

Related commands: **ADS, CFS, CGP, CMF, DEL, DIR, DSD, EES?, HCF?, ICR, IHD, IPA, LPR, MBF, MCS, MRG, MSS, MVF, OMP, PCS, PHD, PSC, RHD?, RMB?, RMS?, RMV?, SBR, SCM, SHD, SIS, SML, SMS, SNO?, SPP, SPS, SRP, STD, TCD, TRD, TRE, TRG, TRR, TSV**

Communications processor: **CP32**

HCF?

Hardware Configuration Query
Determine the hardware configuration

Syntax: HCF?(x)

Parameter: none

Response: *q1,q2,q3(y):*

q1: AB22A (1=present, 0=not present)

q2: PCMCIA hard disk in slot 0 (1=present, 0= not present)

q3: size of the internal measured value memory (bytes)

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

Communications processor: **CP32**

SCM

Start and Configure Measurement
Start, stop and save measurements

Syntax: SCMp1,p2, p3(x)

Parameters p1: mode

- 0 stop measurement
- 1 start measurement with current data acquisition parameters
- 2 save current data acquisition parameters
- 3 read data acquisition parameter set

p2: data acquisition parameter no. (1...16)

p3: recording status (for p1=2 only)

0: halt
1: record

Effect: saves programmed measurements with the "fast data acquisition" commands.

Note:

Measurements can also be started or stopped using a function key on the AB22A/AB32. Parameters are saved to file "HDCNFDAT.BIN" on the hard disk.

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

Communications processor: **CP32**

SCM?

Start and Configure Measurement Query
Read out the number of saved measurement programs

Syntax: SCM?(x)

Parameters none

Response: *q1,q2(y*

q1: the number of saved measurement programs
q2: recording status of the current parameter set

TRG

Trigger Command
Initiate one-time trigger (software trigger)

Syntax: TRGp1(x)

Parameter:

p1	Trigger machine
6500	Start trigger machine (default)
6501	Stop trigger machine
6502	Measuring rate trigger machine

Response: *q1(y)*

q1 = 0: command successfully executed

q1 = ?: last software trigger has not yet been initiated

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

Communications processor: **CP32**

7 MGC*press* commands

Command structure

Scaling

All the force and displacement coordinates used in the commands described below are non-scaled values in the range –30000 to +30000. 30000 corresponds to the input range of the allocated measurement channel: *scaled value = (non-scaled value)/30000*input range*

Example:

Command: **FDA? <LF>**

Response: **15000,20400<CR><LF>**

Force channel input range: **2000N**

Displacement channel input range: **25.00mm**

Force= 15000/30000*2000N = 1000N

Displacement = 20400/30000*25.00mm =17.00mm

Data format

Unless specifically stated otherwise, all parameters are input and output as ASCII strings. If there are several parameters, these are separated by commas. The end of the command can be marked by ‘;’,<LF>,<CR><LF> or <LF><CR>. Responses to commands always end with <CR><LF>.

Example:

Command: **FDH?0 <LF>**

Response: **2594,125<CR><LF>**

Additional details of the command structure can be found in the MGCplus Operating Manual “Operation with computer or terminal”

- **The ARRAY format:**

This format is used to output binary data

1. Byte:	‘#’
2. Byte:	length (as ASCII char.) of the following length specification in byte: ‘1’,’2’ or ‘3’
3. – 5.Byte:	number (as ASCII) of the following bytes: ‘0’ ...’246’

following bytes: binary data

Example:

#212<b0><b1><b2><b3><b4><b5><b6><b7><b8><b9><b10><b1 1><CR><LF>

1.Byte:	‘#’ Start character
2.Byte:	‘2’ Length of the following length specification = 2
3. – 4. Byte :	number of following bytes = 12
5. – 16. Byte:	<b0>..<b11> = 12 byte binary data
16., 17.Byte:	<CR><LF> = end identifier

- **HEXSTR format:**

Binary data is transmitted as an ASCII string

Example:

“AA55AA55AA55AA55AA55”<CR><LF>

10 bytes are transmitted as a string (the quotation marks are transmitted as well)

Caution!:
In ARRAY and HEXSTR formats, int, long and floating values are transmitted in Motorola format, i.e. the most significant byte first!

CTRL–R (ASCII code 18 decimal)

This control character activates operation via the RS232 interface.

AID?

Amplifier Identification Query
Amplifier identification output

Syntax:	AID?
Parameter:	none
Effect:	outputs the amplifier identification
Response:	string of 20 characters

Related commands: **AID?**, **BDR**, **DCL**, **DSD**, **FDA**, **FDR?**, **FDE**, **FDH**, **FDI?**, **FDS**, **FDW**, **MDD**, **PFS**, **PRQ**, **PSC**, **RES**, **TDD**, **UCC**

Amplifier type: **MC, MLxx**

BDR

Set Baud Rate
The CP12 uses this command to switch over the baud rate on the internal bus

Syntax	BDRp1,p2,p3
Parameters	p1: baud rate (19200,9600,1) With p1 == 1, the CP12 will query whether the ML85 supports baud rate 19200. A ‘0’ is sent as a response, the baud rate of the serial interface is not modified.
	p2: parity (0 = none, 1 = odd, 2 = even)
	p3: stop bits (1,2)
Effect:	switches over the baud rate of the ML85 serial interface
Response:	0: command successfully executed
	?: error

Related commands: **BDR**, **SBR**, **SRB**

Communications processor: **CP12, CP32**

DCL

Device Clear
Terminate communication

Syntax: DCL or CTRL A

Parameter: none

Effect: remote control operation is terminated

Response: none

Communications processor: **CP12, CP32**

Related commands: **AID?**, **BDR**, **DCL**, **DSD**, **FDA**, **FDR?**, **FDE**, **FDH**, **FDI?**, **FDS**, **FDW**, **MDD**, **PFS**, **PRQ**, **PSC**, **RES**, **TDD**, **UCC**

DSD

Data Set Device (from Version P4.00, with CP32B only)
Save setup

Syntax: DSDp1

Parameter: p1 amplifier setup data, queried from the amplifier with the MDD? command (format HEXSTR) 178 bytes = 356 characters (see below)

Effect: this command is used to save the complete setup.

Response: 0: command successfully executed

?: error

Communications processor: **CP12, CP32**

Contents of the parameter string:

Offset	Type	Contents
0	char	Channel number
1	char	Amplifier code
2	char	Parameter set number
3	char	Width
4	int	End position
6	int	Timeout time
8	int	Lefthand displacement end window
10	int	Righthand displacement end window
12	int	Block force or Fmin
14	int	Settling time
16	int	Lefthand displacement range window
18	int	Righthand displacement range window
20	int	Bottom force range window
22	int	Top force range window
24	int	Lefthand displacement thread-in window
26	int	Righthand displacement thread-in window
28	int	Bottom force thread-in window
30	int	Top force thread-in window
32	char	Status thread-in window (see FDW)
33	int	Lefthand displacement fitting window[2]
35	int	Righthand displacement fitting window[2]
37	int	Bottom force fitting window[2]
39	int	Top force fitting window[2]
41	int	Bottom right force fitting window[2]
43	int	Top right force fitting window[2]
45	char	Status fitting window[2] (see FDW)

Contents of the parameter string (contd.):

Offset	Type	Contents
46	Window	Data for fitting window[3] (structure as fitting window[2]
59	Window	Data for fitting window[4] (structure as fitting window[2]
72	Window	Data for fitting window[5] (structure as fitting window[2]
85	Window	Data for fitting window[6] (structure as fitting window[2]
98	Window	Data for fitting window[7] (structure as fitting window[2]
111	char	Abort condition (MessEnd)
112	int	delta_s
114	int	delta_t
116	int	Force value for transducer test
118	int	Position for transducer test
120	char	Delta value as ‰ for max. permitted deviation (see FDS)
121	char	Print format (see PFS)
122	int	Alarm force
124	int	Alarm displacement
126	char	WidthF (see FDS) (from Version 4.00)
127	char	WidthS (see FDS) (from Version 4.00)
128	char[48]	reserve

Related commands: **AID?**, **BDR**, **DCL**, **DSD**, **FDA**, **FDR?**, **FDE**, **FDH**, **FDI?**, **FDS**, **FDW**, **MDD**, **PFS**, **PRQ**, **PSC**, **RES**, **TDD**, **UCC**

DSD?

Data Set Device Query (from Version P4.00, with CP32B only**)**

Syntax: DSD?p1

Parameter: p1 Channel number of amplifier

Effect: outputs the amplifier setup parameters

Response: hexadecimal string 176 bytes = 356 characters (HEXSTR format)

Communications processor: **CP12, CP32**

FDA

Force Displacement Alarm
Set alarm limits for the /ALARM output

Syntax: FDA p1,p2,p3

Parameter:

Par	Meaning
p1	(int) maximum force (–30000 ... 30000)
p2	(int) position (–30000 ... 30000)
p3	(int) minimum force (–30000 ... 30000) (from Version P4.10)

Effect: sets the limit values for force and position; undershooting or overshooting these values triggers an alarm. (If the lefthand displacement limit of the range window is higher than the righthand one, an alarm is triggered by undershooting position p2.)

Response: 0: command successfully executed

?: error

Related commands: **AID?, BDR, DCL, DSD, FDA, FDR?, FDE, FDH, FDI?, FDS, FDW, MDD, PFS, PRQ, PSC, RES, TDD, UCC**

Amplifier type: **ML85...**

FDA?

Force Displacement Alarm Query
Set alarm limits for the /ALARM output

Syntax: FDA?

Parameter: none

Effect: outputs the limit values for force and position; undershooting or overshooting these values triggers an alarm.

Response:

Par	Meaning
q1	(int) maximum force (–30000 ... 30000)
q2	(int) position (–30000 ... 30000)
q3	(int) minimum force (–30000 ... 30000) (from Version P4.10)

FDR?

Force Displacement Buffer Query
Query measured values in compressed or non-compressed form

Syntax: FDB?p1,p2,p3,p4

Parameter: for p1 = 0...2:

Par	Meaning
p1	(char) type of diagram 0 : “force/displacement”, 1 : “force/time”, 2 : “displacement/time”
p2	(char) number of measurement pairs for compression
p3	(int) lower range limit for range to be compressed (x value)
p4	(int) upper range limit for range to be compressed (x value)

Effect: transmission of measurement pairs

Response: a1,a2

a1	Number of transmitted measurement pairs
a2	Array of measured values

Parameter: for p1 = 3, 4:

Par	Meaning
p1	Type of measurement output 3: HEXSTR format, 4: ARRAY format
p2	Number of required measurement block (0...49)

Effect: output of 15 measurement triples at a time from the measurement store (force, displacement, time) as 16-bit integer values

Response: a1,a2(for p1 = 3 as HEXSTR, for p1 = 4 as ARRAY)

Parameter: for p1 = 5:

Par	Meaning
p1	Index for measurement store

Effect: output of a measurement triple at a specific index as decimal values

Response: a1(force),a2(displacement),a3(time)

Parameter: for p1 = 6:

Par	Meaning
p1	0: displacement, 1: time

Effect: search output of a measurement triple from the measurement store for a given displacement coordinate or a given time

Response: a1(force),a2(displacement),a3(time)

Overall evaluation result
The evaluation result for the overall measurement can be queried as an integer value.

0x8000 = Initialisation value (neither OK nor NOK)

0 = result OK

All other values= result NOK

Bit	Meaning
0	1: Thread-in window NOK
1	1: Fitting window 2 NOK
2	1: Fitting window 3 NOK
3	1: Fitting window 4 NOK
4	1: Fitting window 5 NOK
5	1: Fitting window 6 NOK
6	1: Fitting window 7 NOK
7	1: End window NOK
14	Cumulative message: NOK

Individual window evaluation result
The evaluation result of each window can be queried as an integer value.

0x8000 = Initialisation value (neither OK nor NOK)

0 = result OK

All other values= result NOK

The individual bits contain additional information on the cause of the error

Bit	Meaning
0	Displacement coordinate of the first measured value inside the window
1	Displacement coordinate of the last measured value inside the window
2	End force too low
3	End of force/displacement characteristic not in end window
4	Lefthand displacement coordinate of window not in plot
5	Righthand displacement coordinate of window not in plot
6	Undershooting lower diagonal window limit
7	Overshooting upper diagonal window limit
8	Minimum force too low
9	Minimum force too high
10	Maximum force too low
11	Maximum force too high
12	Start of force/displacement characteristic not in thread-in window
13	End force too high
14	Cumulative message: window NOK

Related commands: **AID?, BDR, DCL, DSD, FDA, FDR?, FDE, FDH, FDI?, FDS, FDW, MDD, PFS, PRQ, PSC, RES, TDD, UCC**

Amplifier type: **ML85...**

FDE

Force Displacement Evaluation Start/Stop
Start/Stop evaluation

Syntax: FDEp1

Parameter: p1 = 0: stop measurement, 1: start measurement

Effect: start and stop measurement by software

Response: 0: command successfully executed

?: error

Related commands: **AID?, BDR, DCL, DSD, FDA, FDR?, FDE, FDH, FDI?, FDS, FDW, MDD, PFS, PRQ, PSC, RES, TDD, UCC**

Amplifier type: **ML85...**

FDE?

Force Displacement Evaluation Query
Query evaluation results

Syntax: FDE?p1,p2

Parameter: p1,p2 see table

Effect: this command is used to transmit the status of the last measurement

Response:

p1	p2	Response
0	–	[status] [anz_messw]
1	–	[Auswertung]
2	–	[meßdauer]
3	0	[GesErgebnis] [endweg] [endkraft] [Nachlauf] [KoordDiff]
3	1...7	[ErgFenster]
3	8	[ErgEndfnstr]
4		[GrundEnde] [AlarmWert]

Auswertung: structure in ARRAY format (see below)

AlarmWert: (int) value which led to triggering of ALARM output

anz_messw: (int) number of measured values recorded

endkraft: (int) last force measured value

endweg: (int) last displacement measured value

ErgFenster: (int) fitting window evaluation result

ErgEndFnstr: (int) end window evaluation result

GesErgebnis	(int) overall result of the evaluation (0 = OK, 0x8000 = initialisation value, >0 = NOK (see page H–152)
GrundEnde:	reason for terminating measurement:
	1: righthand displacement limit of range window reached
	2: maximum measurement duration reached
	3: settling time reached
	4: measurement suspended by external STOP input
	5: measurement suspended by FDE0 command
	6: standstill detected
	7: measurement store full
	8: measurement store pointer error (internal error)
	9: alarm force overshoot
	10: alarm displacement overshoot
KoordDiff:	(int) difference between absolute and relative displacement coordinates
	(endweg – EndPos)
meßdauer:	(int) duration of the press-fit operation as 1/2400 s
Nachlauf	(int) displacement change while settling time counter is running

status: (char): bit 0,1: 0 = stop, 1 = measurement running", 2 = "evaluation complete"

bit 2: parameter set has changed (deleted automatically by command FDE?0 query from CP32B) (from P4.11)

bit 3: new measured values (deleted automatically by command FDE?0 query from CP32B) (from P4.11)

bit 4: parameter set has changed (deleted automatically by command FDE?0 or FDH? query from AB22) (from P4.11)

bit 5: new measured values (deleted automatically by command FDE?0 or FDH? query from AB22) (from P4.11)

bit 6: parameter set has changed (deleted automatically by command FDE?0 query from external interface) (from P4.11)

bit 7: new measured values (deleted automatically by command FDE?0 query from the external interface) (from P4.11)

Contents of the [Auswertung] structure

Off set	Type	Contents
0	int	Overall result of the last measurement run:0=OK,>0=NOK
2	int	End displacement (position when measurement completed)
4	int	End force (force when measurement completed)
6	int	Overshoot (displacement change while settling time counter is running)
8	int	Total number of press-fit processes performed
10	int	Number of flawed press-fit processes
12		Result end window
14	int	Minimum force in end window
16	int	Maximum force in end window
18	int	Displacement coordinate for minimum force in end window
20	int	Displacement coordinate for maximum force in end window
22	int	Block difference (actual end position–preset end position)
24		Result window[0]‹
26		Minimum force in window[0]
28		Maximum force in window[0]
30		Displacement coordinate for minimum force in window[0]
32		Displacement coordinate for maximum force in window[0]
34	WINDO	Data for window[1] (structure as window[0])
44	WINDO	Data for window[2] (structure as window[0])
54	WINDO	Data for window[3] (structure as window[0])
64	WINDO	Data for window[4] (structure as window[0])
74	WINDO	Data for window[5] (structure as window[0])
84	WINDO	Data for window[6] (structure as window[0])
94	char	Parameter set number (from P4.10)
95	char	reserved

FDH **Force Displacement Histogram**
Delete statistics buffer

Syntax: FDHp1

Parameters p1

p1	Effect
0	Deletes the 'press-fit processes' and 'error' counters
1...7	Deletes the respective histogram values for fitting windows 1...7
8	Deletes the histogram values for the end window
9	Deletes the entire statistics buffer (0...8 combined)

Effect: see table

Response: 0: command successfully executed

?: error

Related commands: **AID?**, **BDR**, **DCL**, **DSD**, **FDA**, **FDR?**, **FDE**, **FDH**, **FDI?**, **FDS**, **FDW**, **MDD**, **PFS**, **PRQ**, **PSC**, **RES**, **TDD**, **UCC**

Amplifier type: **ML85...**

FDH?

Force Displacement Histogram Query
Query the statistics buffers

Syntax: FDH?p1

Parameters p1 (see table)

Effect: see table

Response:

p1	Response
0	[ges_zahl] [fehler_zahl]
1...7	[histo] [mittel] [s–abb] [zahl] [io]
8	[end_histo] [io]

end_histo: (u int[9]) end position histogram: shows the distribution of the end points inside the end window

fehler_zahl: (u int) number of flawed press-fit processes

ges_zahl: (u int) total number of all press-fit processes performed

histo: (u int[2][9]) histogram values for the minima and maxima of the measured forces

io: (u int) number of press-fit processes evaluated with “OK” in this window

mittel: (u int) mean values for the minima and the maxima of the measured forces

s–abb: (u int) standard deviations for the minima and the maxima of the measured forces

zahl: (u int) number of the press-fit processes, where the measurement curve runs right through the window

FDI?

Force Displacement IO Query
Output of the IO port status

Syntax: FDI?

Parameters none

Effect: outputs the status of the IO port

Response:

p1	Response
–	[Eingänge] [Ausgänge]

Bit	Input (int)	Output (int)
0	PRINT	BUSY
1	CLEAR	N OK
2	CHECK	OK
3	NF	/ALARM
4	P2	/WARN
5	P1	/THR
6	P4	No.1
7	START	No.2
8	→0← F	No.8
9	→0← s	No.4
10	CAL	–

Related commands: **AID?**, **BDR**, **DCL**, **DSD**, **FDA**, **FDR?**, **FDE**, **FDH**, **FDI?**, **FDS**, **FDW**, **MDD**, **PFS**, **PRQ**, **PSC**, **RES**, **TDD**, **UCC**

Amplifier type: **ML85...**

FDS

Force Displacement Setup
Press-fit measurement run control

Syntax: FDSp1,p2,p3,p4,p5,p6

Effect: see table

Response: 0: command successfully executed
?: error

Parameter p1, p2, p3, p4, p5, p6	Description
0 [max_t] [weite] [MessEnd] [delta_s] [delta_t]	Abort conditions
1 [x0] [y0] [delta0]	Transducer test window
2 [WeiteF] [WeiteS]	ΔF and Δs for data acquisition (from Version P4.00)

delta_s: (int) displacement difference for standstill recognition

delta_t: (int) time difference for standstill recognition

Result: (int) 0 = transducer test OK, 1 = transducer test not OK

max_t: (int) timeout time, after which measurement is aborted

MessEnd: (char) bit 0...2: abort criterion: 0 = block force+settling time,
1 = standstill recognition, 2 = external stop signal,
3 = target position+settling time

bit 3: relative displacement coordinates:
0 = relative to the end position
1 = relative to the calculated start position

bit 4: displacement coordinate output:
0 = absolute, 1 = relative

weite: (char) value from 1...255: (see WeiteF and WeiteS); this parameter is also used for data compression of the time axis (command FDB?)

WeiteF	(int) (0...255) determines the DF, to be used to record a new measurement pair.
	DF = ($F_{top\ range\ window}$ – $F_{bottom\ range\ window}$) * WeiteF/750 (from Version P4.00)
	If WeiteF = 0, the value ”weite” is used to calculate DF (compatibility with older versions).
WeiteS	(int) (0...255) determines the Ds, to be used to record a new measurement pair.
	Ds = ($s_{righthand\ range\ window}$ – $s_{lefthand\ range\ window}$) * WeiteS/750 (from Version P4.00)
	If WeiteS = 0, the value ”weite” is used to calculate Ds (compatibility with older versions).
x0:	displacement value of the zero position (transducer test)
y0:	force value of the zero position (transducer test)
delta0	permitted deviation as ‰ from the specified zero position (transducer test)

Related commands: **AID?**, **BDR**, **DCL**, **DSD**, **FDA**, **FDR?**, **FDE**, **FDH**, **FDI?**, **FDS**, **FDW**, **MDD**, **PFS**, **PRQ**, **PSC**, **RES**, **TDD**, **UCC**

Amplifier type: **ML85...**

FDS?

Force Displacement Setup Query
Query the setup

Syntax: FDS? p1

Parameters p1

Effect: outputs the setup

Response:

p1	Output
0	[max_t] [weite] [MessEnd] [delta_s] [delta_t]
1	[x0] [y0] [delta0] [Ergebnis]
2	[WeiteF] [WeiteS]

For descriptions of the output parameters, see FDS

FDW

Force Displacement Window
Change window limits

Syntax: FDWp1,p2,p3,p4,p5,p6,p7

Effect: see table

Response: 0: command successfully executed

?: error

Parameter p1, p2, p3, p4, p5, p6, p7		Meaning
[0]	[xl] [xr] [yu] [yo]	Window limits, range window
[1]	[ZustEin] [xl] [xr] [yu] [yo]	Window limits, thread-in window
[2...7]	[Zustand] [xl] [xr] [yu] [yo] [yur] [yor]	Window limits, fitting windows 2...7
[8]	[xl] [xr] [f_block] [t_setz] [EndPos]	Window limits, end window

EndPos: (int) start position or end position (depending on abort criterion selected (see FDS)

f_block: (int) block force or F_{min}

t_setz: (int) settling time in 1/2400 sec (once the minimum force is reached in the end window, measurement is aborted after expiry of the settling time)

xl: (int) lefthand displacement value

xr: (int) righthand displacement value

yo: (int) top force value

yor: (int) top right force value (for diagonal windows)

yu: (int) bottom force value

yur: (int) bottom right force value (for diagonal windows)

Zustand:	bit 0: 0 =	inactive
	1 =	active
	bit 1,2: 0 =	absolute force
	1 =	force relative to the minimum force measured in Window2 (not permitted for fitting windows 0 and 1)
	2 =	force relative to the maximum force measured in Window2 (not permitted for fitting windows 0 and 1)
	3 =	not permitted
	bit 4: 0 =	absolute displacement coordinates
	1 =	relative displacement coordinates (not permitted for fitting window 0)
	bit 5,6: 0 =	curve shape evaluation
	1 =	Fmin evaluation
	2 =	Fmax evaluation
	3 =	mean value evaluation
ZustEin:	0 =	inactive
	1 =	active

Related commands: **AID?, BDR, DCL, DSD, FDA, FDR?, FDE, FDH, FDI?, FDS, FDW, MDD, PFS, PRQ, PSC, RES, TDD, UCC**

Amplifier type: **ML85...**

FDW?

Force Displacement Window Query
Query window limits

Syntax: FDW?p1

Parameter: p1 see table

Effect: this command is used to transmit the full status of the last measurement

Response: see table

p1	Response
0	[xl] [xr] [yu] [yo]
1	[Zustand] [xl] [xr] [yu]
2...7	[Zustand] [xl] [xr] [yu] [yo] [yur] [yor]
8	[Zustand] [xl] [xr] [f_block] [t_setz] [EndPos]

For descriptions of the output parameters, see FDW

Amplifier type: **ML85...**

MDD

Memory Device Data (with CP32B only)
Save setup

Syntax: MDDp1

Parameter: p1 amplifier setup data, fetched from the amplifier with the DSD? command (as hexadecimal string) 123 bytes = 246 characters (see below)

Effect: this command is used to save the complete setup.

Response: 0: command successfully executed

?: error

CAUTION!

Not all the parameters are transmitted!! The command is merely available for reasons of compatibility –> use DSD

Contents of the parameter string:

Offset	Type	Contents
0	char	Channel number
1	char	Amplifier code
2	char	Parameter set number
3	char	Width
4	int	End position
6	int	Timeout time
8	int	Lefthand displacement end window
10	int	Righthand displacement end window
12	int	Block force or F_{min}
14	int	Settling time
16	int	Lefthand displacement range window
18	int	Righthand displacement range window
20	int	Bottom force range window
22	int	Top force range window
24	int	Lefthand displacement thread-in window
26	int	Righthand displacement thread-in window
28	int	Bottom force thread-in window
30	int	Top force thread-in window
32	char	Status thread-in window (see FDW)
33	int	Lefthand displacement fitting window[2]
35	int	Righthand displacement fitting window[2]
37	int	Bottom force fitting window[2]
39	int	Top force fitting window[2]
41	int	Bottom right force fitting window[2]

Offset	Type	Contents
43	int	Top right force fitting window[2]
45	char	Status fitting window[2] (see FDW)
46	Window	Data for fitting window[3] (structure as fitting window[2]
59	Window	Data for fitting window[4] (structure as fitting window[2]
72	Window	Data for fitting window[5] (structure as fitting window[2]
85	Window	Data for fitting window[6] (structure as fitting window[2]
98	Window	Data for fitting window[7] (structure as fitting window[2]
111	char	Abort condition (MessEnd)
112	int	delta_s
114	int	delta_t
116	int	Force value for transducer test
118	int	Position for transducer test
120	char	Delta value as ‰ for max. permitted deviation (see FDS)
121	char	Print format (see PFS)
122	int	Alarm force

Related commands: **AID?**, **BDR**, **DCL**, **DSD**, **FDA**, **FDR?**, **FDE**, **FDH**, **FDI?**, **FDS**, **FDW**, **MDD**, **PFS**, **PRQ**, **PSC**, **RES**, **TDD**, **UCC**

Communications processor: **CP12, CP32**

MDD?

Memory Device Data Query (with CP32B only)
Putputs setup parameters

Syntax: MDD?p1

Parameter: p1 Channel number of amplifier

Effect: outputs the amplifier setup parameters

Response: hexadecimal string 123 bytes = 246 characters (HEXSTR format)

CAUTION!
Not all the parameters are transmitted!! The command is merely available for reasons of compatibility –> use DSD

PFS

Print Format Select
Specifies Print Format

Syntax: PFSp1

Parameter:

p1	Information
1	Number of the press-fit process and the overall result
2	End force and end position
4	Individual window result
8	Force minima and maxima of the individual windows
16	Target values for minimum and maximum force for each measurement
32	Target values for minimum and maximum force only for windows with NOK result
64	Automatic print request after each measurement run
128	Automatic print request after each measurement run, only if overall result NOK

You can set any combinations from the above table, by adding code numbers.

Effect: specifies information to be printed. The setting affects the print output through the external PRINT input on the ML85 and through an internal print request.

Response: 0: command successfully executed

?: error

Related commands: **AID?**, **BDR**, **DCL**, **DSD**, **FDA**, **FDR?**, **FDE**, **FDH**, **FDI?**, **FDS**, **FDW**, **MDD**, **PFS**, **PRQ**, **PSC**, **RES**, **TDD**, **UCC**

Amplifier type: **MC, MLxx**

PFS?

Print Format Select Query
Output Print Format

Syntax:	PFS?
Parameter:	none
Effect:	outputs information to be printed.
Response:	p1: information which has been set (for coding see PFS command)

PRQ

Printer Request
Initiate printing

Syntax:	PRQ
Parameter:	none
Effect:	this command initiates printing (same effect as PRINT input)
Response:	0: command successfully executed
	?: error

Related commands: **AID?**, **BDR**, **DCL**, **DSD**, **FDA**, **FDR?**, **FDE**, **FDH**, **FDI?**, **FDS**, **FDW**, **MDD**, **PFS**, **PRQ**, **PSC**, **RES**, **TDD**, **UCC**

PRQ?

Printer Request Query
Query print request

Syntax:	PRQ?
Parameter:	none
Effect:	the CP 12 uses this command to establish which press-fit channel has an existing print request. The query deletes the request flag in the channel concerned.
Response:	0: no print request exists
	1: a print request exists

PSC

Press Select Channels
Select press-fit controller channels for recording

Syntax: PSC p1,..., p16 (x)

Parameter: p1,...,p16

Controller channels whose evaluations are to be recorded.

PSC 0 (x) deselects all channels !

Related commands: **AID?**, **BDR**, **DCL**, **DSD**, **FDA**, **FDR?**, **FDE**, **FDH**, **FDI?**, **FDS**, **FDW**, **MDD**, **PFS**, **PRQ**, **PSC**, **RES**, **TDD**, **UCC**

Communications processor: **CP32**

PSC?

Press Select Channels Query

Syntax: PSC?p1(x)

Parameter: p1: output mode

0: available controller channels

1: selected controller channels

Response: *q1,...q16:*

List of existing or active controller channels

RES

Reset
Execute warm restart

Syntax:	RES
Parameter:	none
Effect:	the device executes a warm restart. Communication is terminated.
Response:	none

Related commands: **AID?**, **BDR**, **DCL**, **DSD**, **FDA**, **FDR?**, **FDE**, **FDH**, **FDI?**, **FDS**, **FDW**, **MDD**, **PFS**, **PRQ**, **PSC**, **RES**, **TDD**, **UCC**

Communications processor: **CP12, CP32**

TDD

Transmit Device Data

Syntax: TDDp1,p2

Parameter:

p1	Amplifier setups		
0	ROM	→	RAM (Setup Factory Data)
1	EEPROM	→	RAM (Recall Data)
2	RAM	→	EEPROM (Save Data)
5	EEPROM	→	RAM (Recall Comment)
6	RAM	→	EEPROM (Save Comment)

p2: parameter set number under which the present amplifier setting is to be saved (1...8). p2 is accepted as the current parameter set number

Effect: cold start/warm start and data storage to the EEPROM

Response: 0: command successfully executed

?: error

Related commands: **AID?**, **BDR**, **DCL**, **DSD**, **FDA**, **FDR?**, **FDE**, **FDH**, **FDI?**, **FDS**, **FDW**, **MDD**, **PFS**, **PRQ**, **PSC**, **RES**, **TDD**, **UCC**

Amplifier type: **MC, MLxx**

TDD?

Transmit Device Data Query

Syntax: TDDp1

Parameter:

p1	Response
0	As per table below
1	Number of the current parameter set

Effect: Queries the origin of the currently active parameter setting

Response: for p1 = 0

0	Setup
1	Internal EEPROM
2	User input
?	Error

for p1 = 1: number of the current parameter set (1...8)

UCC

User Channel Comment
Enter comment

Syntax:	UCCp1
Parameter:	p1 Any string “___” (45 characters)
Effect:	you can use this command to file any comment in the evaluation channel
Response:	0: command successfully executed
	?: error

Related commands: **AID?**, **BDR**, **DCL**, **DSD**, **FDA**, **FDR?**, **FDE**, **FDH**, **FDI?**, **FDS**, **FDW**, **MDD**, **PFS**, **PRQ**, **PSC**, **RES**, **TDD**, **UCC**

Amplifier type: **MC, MLxx**

UCC?

User Channel Comment Query
Output comment

Syntax:	UCC?
Parameter:	none
Effect:	you can use this command to read out any comment filed in the evaluation channel
Response:	“______” (string with 45 characters)

8 PC-Card hard disk

CFS

Compression Factor Set (from P1.30)
Enter compression factor

Syntax: CFSp1(x)

Parameter: p1: comment selection

p1	Compression factor
0	off
480...32768	avail. setting range

Effect: while a data set (file.mea) is being recorded to the PCMCIA HDD, this command is used to create a second, accelerated data set, to be filed under the file name file.sto. This data set contains a min/max value pair for every so many measured values, in accordance with the compression factor.

Example: measured values to be recorded = 1000, selected compression factor = 500 gives 1000/500 = 2 min/max value pairs.

Note:

A value pair is not filed for measured values which are left over ($<$ 480), or if the compression factor selected is too high (CF $>$ number of measured values).

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

CFS?

Compression Factor Set Query (from P1.30)
Read out compression factor

Syntax:	CFS?(x)
Parameter:	none
Response:	*q1(y):* compression factor

CMF

Comment Measurement File (from P1.30)
Put comments in measurement file

Syntax: CMFp1,p2(x)

Parameter: p1: comment selection

- 0 p2 becomes the active comment
- 1...n comment from the Comments.txt file becomes the active comment
- –1 p2 becomes the comment of the active data acquisition setup

p2: comment, “___ASCII string___” (max. 80 characters)

Effect: the active comments on the measurement and on the data acquisition setup are put at the start of the measurement file when a measurement run is recorded on the PCMCIA hard disk.

Note:

A selection of comments on the measurement run is put in the file Comments.txt. They must be available on a line basis (max. 80 characters/line). The data acquisition setup comments are put in these automatically.

CMF?

Comment Measurement File Query (from P1.30)
Read out comments on the measurement run or on the data acquisition setup

Syntax: CMF?p1(x)

Parameter:

p1	Comment
0	Active comment
1...n	from file Comments.txt
–1...–16	...of data acquisition setup 1...16
–17	...of data acquisition setup 0 (FLASH)
–18	...of active data acquisition setup

Response: *"___comment___"(y):* max. 80 characters

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

DEL

Delete Measurement File
Delete a file from the PCMCIA hard disk

Syntax: DELp1(x)

Parameter: p1: complete file name without wildcards

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

DIR?

Directory of hard disk
Output of a directory of the measurement files on the hard disk or the specific parameters of one file, or of the hard disk itself.

Syntax: DIR?p1,p2(x)

Parameter: p1: mode

- 1: hard disk information
- 2: file information
- 3: directory information

p2: file name with or without wildcards (‘*xxx.*xx‘)

Response: p1 = 3: q1(y)

- q1: file name

In order to receive the next file name, the command must be repeated without p2.

p1 = 2: q1,q2,q3(y)

- q1: time of file creation
- q2: date of file creation
- q3: file size

p1 = 1: q1,q2(y)

- q1: capacity of PCMCIA hard disk in bytes
- q2: capacity available on PCMCIA hard disk

Example:	DIR?3,file.mea(x) cdateo.mea(y)
	DIR?2,bfile.mea(x) 14:06,22.01.97,567084(y)
	DIR?1(x) 170350600,95048364(y)
Related commands:	**ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

IHD

Integrate Hard-Disk
Divert the measurement output to the PCMCIA hard disk

Syntax:	IHDp1,p2(x)	
Parameters	p1:	mode
		0: close measurement file (file name may be omitted)
		1: open the measurement file with the specified name (the file is freshly created, i.e. if a file with this name already exists it will be overwritten)
	p2:	file name
		Name of the measurement file to be recorded, with extension in accordance with DOS convention (xxxxxxxx.xxx)

If working with various measurement rates (see MRG command in), a separate measurement file is created for each measurement rate group. The file names are assigned as follows:

Group 0:	NAME.MEA
Group 1:	NAME.ME1
Group 2:	NAME.ME2
Group 3:	NAME.ME3

If working with data reduction, the file names for the reduced data records are assigned as follows:

Group 0:	NAME.ST0
Group 1:	NAME.ST1
Group 2:	NAME.ST2

There is no data reduction for the asynchronous group.

IHD?

Integrate Hard-Disk Query
Output the data diversion information

Syntax: IHD?p1(x)

Parameter: p1: 0,...,2 measurement rate group
3 asynchronous group

Response: *q1,q2,q3,q3,q4,q5,q6(y)*

q1: status of the measurement file (1=open, 0=closed)
q2: status of the recording (1=running, 0=not running)
q3: size of the measurement file (bytes)
q4: number of measured value lines in the measurement file / channel in the measurement file
q5: name of the measurement file
q6: number of measured value lines to be recorded (RMB? command)

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

PCS

Programming Channel Select
Channel select for the immediately following set-up commands

Syntax: PCSp1,p2,...,p16(x)

Parameter: p1,p2,...,p16: channel numbers (1...16)

Related commands: **ADR?**, **CHS**, **DCL**, **DGM**, **PCS**, **RES**, **RST**, **Sxx**

Communications processor: **CP12, CP32**

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

PCS?

Programming Channel Select Query
Output channel selection for set-up commands

Syntax: PCS?p1(x)

Parameter:

p1	Output mode
0	Existing channels
1	active channels

Response: *q1,...,q16(y):* list of existing or active channels

PHD

Press Data to Hard Disk (with effect from P1.30)
Save press-fit data to CP32B–HDD/activate press-fit controller print monitoring

Syntax: PHDp1,p2(x)

Parameter: p1: data acquisition mode

- 0: finish data acquisition (p2 not required)
- 1: start recording. A set of files is defined under the specified base name (p2). For each press-fit controller, a file is defined with all the relevant parameters, to which the evaluations are saved. The file names are formed as follows:
  base name.channel number (using DOS conventions).
- 2: Print monitoring of the press-fit controller selected by the PSC command starts.

p2: base file name without extension

Effect: a file is created for each press-fit controller, for saving all the relevant setup parameters and press data. The individual file names comprise “base name.channel number”.

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

PHD?

Press data to Hard Disk Query (from P1.30)
Read press data from the CP32B HDD

Syntax:	PHD?p1(x)		
Parameter:	p1:	1...16	Channel number of an evaluating channel (ML85/3)
Response:	*q1,q2,q3,q4(y):*		
	q1:	0,1, 2	Monitoring inactive, data acquisition active, print monitoring active
	q2:	N	Size of the recording file of channel p1 (for q1 =1 only)
	q3:	N	Number of press-fit operations already saved for channel p1 (for q1=1 only)
	q4:	File name	full file name for channel p1 (for q1=1 only)

RHD?

Read from Hard Disk
Output of a measurement file via the active interface

Syntax: RHD?p1,p2,p3(x)

Parameter:
- p1: name of the measurement file to be read
- p2: number of bytes to be sent (from P1.30)
  - 0: output entire file
- p3: start position inside the file (from P1.30)
  - 0: output file from the beginning

Response: *Binary data(y):* file size (4 bytes binary) + measurement data (header + n measured value lines)

Data recording format:

Header: (512 bytes, 4351 bytes from P1.30)

File ID (6001)	(4 byte binary LSB...MSB)
Number of channels	(4 byte binary LSB...MSB)
Size of the measured value lines in bytes	(4 byte binary LSB...MSB)
Number of measured value lines in the file	(4 byte binary LSB...MSB)
Data format	(4 byte binary LSB...MSB)
Measuring rate	(4 byte binary LSB...MSB)
Data offset	(4 byte binary LSB...MSB)
Reserved	(4 byte binary LSB...MSB)

1x per channel (including time stamps):

Channel number	(4 byte binary LSB...MSB)
Scale factor	(4byte floating LSB...MSB)
Data offset (tare)	(4byte floating LSB...MSB)
Unit	(4byte ASCII)
Measuring signal bit field	(4 byte binary LSB...MSB)
Bit 0 = Gross, dynamic Bit 1 = Net, dynamic Bit 2 = Peak value 1 Bit 3 = Peak value 2	
Channel comment	(47 byte ASCII)[1)]: UCC?
Channel comment	(178 byte)[1):DSD?]
Date and time string	(25 byte ASCII)
Comment on measurement	(80 byte)[1)]: CMF?
Comment on the active data acquisition setup	(80 byte)[1)]: CMF?

1) from P1.30

Note:

The measurement file can be output during data acquisition, to the same file. During the transfer, the current measurement will be buffered in FIFO memory. If the FIFO is full, the measurement run will be terminated.

Data:

n measured values* (format as specified in the header)

*) n=[time stamp + sum of the signals (as selected with the MSS command) of all the measurement channels (as selected with the MCS command) x measured value lines (as specified for the RMB? command)]

Example:

pcs1(x)	Select channel 1
mss214(x)	Select gross signal
pcs2(x)	Select channel 2
mss214,204(x)	Select gross signal and peak value signal1
mcs1,2,17(x)	Record channels 1, 2 and time channel

.
.
Start measurement
Redirect measurement output to hard disk!
.
.

rmb?3,6406(x)	Save 3 measured value lines to hard disk

.
.
n=[1 + Gross(1) + Gross(2) + PVS1(2)] x 3 = 12 data

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

SHD

Send File to Hard Disk
Transmit file to CP32B HDD

Syntax: SHDp1,p2,p3(x)

Parameter:
- p1 block number
- p2: file name with extension
- p3: data block “___hexadecimal string___” (128 bytes = 256 characters)

Effect: With p1 = 1, the file is created and the first data block (p3) saved.

With p1 = 2...n, the next data blocks are transmitted.

p1 = 0 closes the file. (p2, p3 not required)

Example: SHD1,test.txt,“30313233343536373839”(x)

File test.txt is created and the first data block is transmitted.

SHD0(x) Close file

Response: *q1(y)*

0: without error
?: error

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

Communications processor: **CP32**

SRP

Set Recording Parameters
Define data acquisition parameters

Syntax: SRPp1,...,p7(x)

Parameter:

p1: 1...n Number of measured value lines in an individual measurement, consisting of a pre-trigger and a post-trigger

p2: 1...n Number of measured value lines in the pre-trigger (post-trigger can be 0)

p3: 1...n Number of consecutive measurements

0: an infinite number of consecutive measurements

p4: status of output redirection to the hard disk

0: output redirection to HDD inactive

1: output redirection to HDD active

p5: name of measurement file on the hard disk

p6: number of measured values for output job on the hard disk

p7: output mode for hard disk data acquisition (only 4609 reasonable, see RMB command)

Effect: this command defines the parameters (TSV, IHD and RMB? commands) for triggered data acquisition without starting the measurement.

Related commands: **ADS**, **CFS**, **CGP**, **CMF**, **DEL**, **DIR**, **DSD**, **EES?**, **HCF?**, **ICR**, **IHD**, **IPA**, **LPR**, **MBF**, **MCS**, **MRG**, **MSS**, **MVF**, **OMP**, **PCS**, **PHD**, **PSC**, **RHD?**, **RMB?**, **RMS?**, **RMV?**, **SBR**, **SCM**, **SHD**, **SIS**, **SML**, **SMS**, **SNO?**, **SPP**, **SPS**, **SRP**, **STD**, **TCD**, **TRD**, **TRE**, **TRG**, **TRR**, **TSV**

Communications processor: **CP32**

SRP?

Set Recording Parameters Query
Read out data acquisition parameters

Syntax:	SRP?(x)
Parameter:	none
Response:	*q1,q2,...,q7(y):* the parameters set with SRP

9 MGC commands (predecessor system to MGC*plus*)

9.1 Addressing

Note:

The commands from the predecessor system (MGC) are supported by MGC*plus* and are also described in this Operating Manual.

ADR?

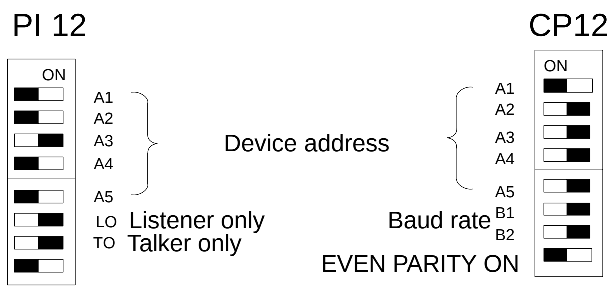
217G-Address Query **Only valid for CP12/CP13**

Output device address

Syntax: ADR?(x)

Parameter: none

Effect: depending on the interface selected, outputs the relevant address for the MGC*plus* device. This address is set with switches (A1-A5) on the CP12 connection board for the RS485 interface and on the PI12 connection board for the IEEE interface. Both address settings are required for bus control.



Response: *q1(y)*: device address 0...31

Note:

Amplifier set-up and function commands only have an effect on the selected channels. Whenever you switch on all channels are selected (active).

Related commands: **ADR?, CHS, DCL, DGM, PCS, RES, RST, Sxx**

Communications processor: **CP12, CP32**

218G-**Define group mem** **DGM**
Only valid for AB12
Allocate group addresses

Syntax:	DGMp1(x)
Parameter:	p1
	0...9
Effect:	each amplifier can be allocated to a group. You can carry out the allocation either from the AB12 (Version P13) or from the external computer. You can allocate 10 group addresses (0...9). Amplifier plug-in modules up to and including version P14, have the group address 0, which means that they are all addressed. If the ”ALL” key is activated, the only amplifiers to respond are those with a group number which is identical to the group number of the channel currently selected. This division into groups only takes effect if operation is through the AB12. The grouping has no meaning for computer control, where all the amplifiers work as if they had group address 0.

Related commands: **ADR?, CHS, DCL, DGM, PCS, RES, RST, Sxx**
Amplifier type: **MC**

DGM?

219G-Define Group Member Query
Output the amplifier group address

Syntax:	DGM?(x)
Parameter:	none
Response:	*q1(y)*: group address

CHS

220G-Channel Select
Select amplifier channels

Syntax: CHS p1(x)

Parameter:

p1	Channel coding value
	1 – 65535

Channel combinations are set by adding the relevant channel coding values.

Channel coding value p1	Channel number
1	1
2	2
4	3
8	4
16	5
32	6
64	7
128	8
256	9
512	10
1024	11
2048	12
4096	13
8192	14
16384	15
32768	16
65535	all channels

Related commands: **ADR?, CHS, DCL, DGM, PCS, RES, RST, Sxx**

Communications processor: **CP12, CP32**

CHS?

221G-Channel Select Query
Output the amplifier channels

Syntax: CHS?p1(x)

Parameter:

p1	Existing / selected channel numbers
0	Existing channels
1	Selected channels

Response: *q1(y)*

Channel coding value q1	Channel number
1	1
2	2
4	3
8	4
16	5
32	6
64	7
128	8
256	9
512	10
1024	11
2048	12
4096	13
8192	14
16384	15
32768	16
65535	all channels

Channel combinations are represented by adding the relevant channel coding values.

9.2 Communication computer/MGCplus

BDR

222G-Baud Rate
Set baud rate, parity bit and number of stop bits for the serial interfaces

Syntax: BDRp1,p2,p3,p4(x)

Parameter:

Baud rate
300
600
1200
2400
4800
9600 1)
19 200
38400 2)
57600 2)

p2	PARITY
0	no
1	Odd
2	Even 1

p3	stop bits
1	1 stop bit 1)
2	2 stop bits

p4	Interface
0	The interface from which MGC*plus* is operated
1	RS-232
2	RS-485

1) Factory setup
2) only with CP32B

The word length for transmission is always 8 bits. If you use the CP12 connection board, the serial interfaces are always configured to match the switch positions.

Related commands: **BDR, SBR, SRB**

Communications processor: **CP12, CP32**

BDR? 223G-Baud Rate Query

Output baud rate, parity bit and number of stop bits for the serial interfaces

Syntax: BDR? p1(x)

Parameter:

p1	Interface
0	The interface, from which MGCplus is operated
1	RS-232
2	RS-485

Response: *q1,q2,q3,q4(y):*

q1	baud rate *
q2	parity *
q3	stop bits *
q4	Interface identification *

*) see tables (p1...p4) for the BDR command

224G-Internal Byte Quer IBY?
Only valid for CP12/CP15

Query baud rates/address switch

Syntax: IBY? p1(x)

Parameter:

p1	
1	Output DIP switch positions
2	Perform RAM test in the CP12

Effect: on the CP12 or PI12 connection boards, you will find an eightfold DIP switch for setting the address and the interface configuration. The IBY? command outputs the ON/OFF position of the switches as decimal numbers.

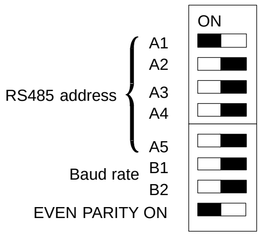
Response: *q1,q2(y):* for querying the switch position (p1=1)

q1	DIP switch on the CP12 connection board
q2	DIP switch on the PI12 connection board

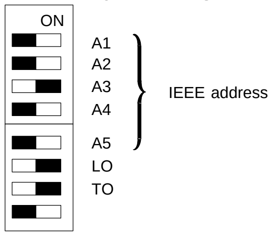
q1(y): for RAM test (p1=2)

q1	
0	No errors in RAM
others	Address of defective memory location

Example 1: the positions of the DIP switch on the CP12 connection board correspond to the factory setting.



The positions of the DIP switch on the PI12 connection board correspond to the factory setting.



IBY?1(x)
129,100(y)

Note:

129 = 1 + 128 (Address 1, 9600 Baud, Even Parity)
100 = 4 + 32 + 64 (Address 4, Addressable)

Example 2: IBY?2(x)
0(y)

RAM test ran without error.

Example 3: IBY?2(x)
8192(y)

RAM location 8192 (corresponds to 2000 Hex) contains errors.

Related commands: **BDR**, **IBY?**, **SBR**, **SRB**

Communications processor: **CP12**

9.3 Amplifier input

ASA

225G-Amplifier Sensor Adaptation
Enter excitation voltage and transducer type

Syntax: ASA p1,p2,p3(x)

Parameter:

p1	Bridge excitation voltage
1	1V (ML10, ML30, ML50, ML55)
2	2.5V (ML10, ML30, ML38, ML50, ML55)
3	5V (ML10, ML30, ML38, ML50)
4	10V (ML10)

no meaning for ML01

Applicable to the ML60: amplitude range for input frequency signal

	ML60	ML35	ML01 and ML10 with AP08
p1	Input amplitude	Measurement type	Decay time
1	typ. 5V	3-wire measurement	Short
2	higher than 100mV	4-wire measurement	Medium
3	–	–	Long

Related commands: **AIS**, **ASA**, **ASS**, **HPS**, **SAD**
Amplifier type: **MC**

Parameter:

p2	Transducer type	Amplifier type
0	DC 10V	ML01
1	S/G full bridge	ML10, ML30, ML38, ML55
2	SG half bridge	ML10, ML55
3	Inductive full bridge	ML50, ML51, ML55
4	Inductive half bridge	ML50, ML51, ML55
5	DC 20mA	ML01
6	Thermocouple type J	ML01
7	Thermocouple type K	ML01
8	Thermocouple type T	ML01
9	Thermocouple type S	ML01
10	DC 75mV	ML01
11	Not used	Not used
12	0 ...2kHz	ML60
13	0 ...20kHz	ML60
14	0 ...200kHz	ML60
15	Pulse counting	ML60
16	500Ω	ML35
17	5000Ω	ML35
18	Pt10	ML35
19	Pt100	ML35
20	Pt1000	ML35
21	Input range 0.1nC	ML01 and ML10 with AP08
22	Input range 1.0nC	ML01 and ML10 with AP08
23	Input range 10.0nC	ML01 and ML10 with AP08
24	Input range 100.0nC	ML01 and ML10 with AP08
25	Full bridge Low (to 30.6mV/V)	ML10
26	Half bridge Low (to 30.6mV/V)	ML10

p2	Transducer type	Amplifier type
27	Full bridge High (to 3060mV/V)	ML10
28	Half bridge High (to 3060mV/V)	ML10
29	SG full bridge 120Ω	ML30 and ML55 with AP14
30	SG full bridge 350Ω	ML30 and ML55 with AP14
31	SG full bridge 700Ω	ML30 and ML55 with AP14
32	SG half bridge 120Ω	ML30 and ML55 with AP14
33	SG half bridge 350Ω	ML30 and ML55 with AP14
34	SG half bridge 700Ω	ML30 and ML55 with AP14
35	SG quarter bridge 120Ω, 4-wire circuit	ML30 and ML55 with AP14
36	SG quarter bridge 350Ω, 4-wire circuit	ML30 and ML55 with AP14
37	SG quarter bridge 700Ω, 4-wire circuit	ML30 and ML55 with AP14
38	SG quarter bridge 120Ω, 3-wire circuit	ML30 and ML55 with AP14
39	SG quarter bridge 350Ω, 3-wire circuit	ML30 and ML55 with AP14
40	SG quarter bridge 700Ω, 3-wire circuit	ML30 and ML55 with AP14
41	Input range 0.1nC	ML01 and ML10 with AP18
42	Input range 1.0V	ML01 and ML10 with AP18
43	Input range 10.0V	ML01 and ML10 with AP18

Parameter:

p3	Shunt connection
0	Off
1	On

only allowed for ML10, ML30, ML38, ML50, ML51, ML55, ML60
for ML01 with AP08 and ML35 only 0 (Shunt off)

ASA?

226G-Amplifier Sensor Adaptation Query
Output excitation voltage and transducer type

Syntax: ASA? p1(x)

Parameter:

p1	
0	Output the bridge excitation voltage and transducer type settings
1	Output table of possible settings for excitation voltage and transducer type

Response: ASA?0(x)
q1,q2,q3(y)

q1	Bridge excitation voltage
q2	Transducer type
q3	Shunt connection

ASA?1(x)
q1,q2,q3(y)

q1	Possible bridge excitation voltage
q2	Possible transducer type
q3	Shunt connection

These responses are dependent on the amplifier.

Re-sponse for ASA? q2	Transducer type	Amplifier type
0	DC 10V	ML01
1	S/G full bridge	ML10, ML30, ML38, ML55
2	SG half bridge	ML10, ML55
3	Inductive full bridge	ML50, ML51, ML55
4	Inductive half bridge	ML50, ML51, ML55
5	DC 20mA	ML01
6	Thermocouple type J	ML01
7	Thermocouple type K	ML01
8	Thermocouple type T	ML01
9	Thermocouple type S	ML01
A	DC 75mV	ML01
B	Not used	Not used
C	0 ...2kHz	ML60
D	0 ...20kHz	ML60
E	0 ...200kHz	ML60
F	Pulse counting	ML60
G	500Ω	ML35
H	5000Ω	ML35
I	Pt10	ML35
J	Pt100	ML35
K	Pt1000	ML35
L	Input range 0.1nC	ML01 and ML10 with AP08
M	Input range 1.0nC	ML01 and ML10 with AP08
N	Input range 10.0nC	ML01 and ML10 with AP08

Re-sponse for ASA? q2	Transducer type	Amplifier type
O	Input range 100.0nC	ML01 and ML10 with AP08
P	Full bridge Low (to 30.6mV/V)	ML10
Q	Half bridge Low (to 30.6mV/V)	ML10
R	Full bridge High (to 3060mV/V)	ML10
S	Half bridge High (to 3060mV/V)	ML10
T	SG full bridge 120Ω	ML30 and ML55 with AP14
U	SG full bridge 350Ω	ML30 and ML55 with AP14
V	SG full bridge 700Ω	ML30 and ML55 with AP14
W	SG half bridge 120Ω	ML30 and ML55 with AP14
X	SG half bridge 350Ω	ML30 and ML55 with AP14
Y	SG half bridge 700Ω	ML30 and ML55 with AP14
Z	SG quarter bridge 120Ω, 4-wire circuit	ML30 and ML55 with AP14
A	SG quarter bridge 350Ω, 4-wire circuit	ML30 and ML55 with AP14
B	SG quarter bridge 700Ω, 4-wire circuit	ML30 and ML55 with AP14
C	SG quarter bridge 120Ω, 3-wire circuit	ML30 and ML55 with AP14
D	SG quarter bridge 350Ω, 3-wire circuit	ML30 and ML55 with AP14
E	SG quarter bridge 700Ω, 3-wire circuit	ML30 and ML55 with AP14
f	Input range 0.1nC	ML01 and ML10 with AP18
g	Input range 1.0V	ML01 and ML10 with AP18
H	Input range 10.0V	ML01 and ML10 with AP18

Example: ASA?1(x)
”01.002.505.0”,”1”,0 (y)

p1	bridge excitation voltage (V)
01.0	1.0
02.5	2.5
05.0	5.0

The index corresponds to the excitation voltage to be set (each element in the table is 4 characters in size).

Transducer type for this amplifier:

p2	Transducer type
1	Only strain gauge full bridge possible

Each element corresponds to the transducer type to be set (see allocation of code numbers to transducer type. Each element in the table is 1 character in size).

Shunt:

p3	Shunt connection
0	Shunt is off

ASS

227G-Amplifier Signal Select
Select amplifier input signal

Syntax: ASS p1(x)

Parameter:

p1	Input source
0	Internal zero signal (zero)
1	Internal calibration signal (calib)
2	Measurement signal (measure)
3	Measures the temperature reference point (ML01 only).

for all amplifiers

Note:

This command triggers the calibration of all amplifiers and only allows further communication after 1 – 3 seconds.

Related commands: **AIS**, **ASA**, **ASS**, **HPS**, **SAD**

Amplifier type: **MC**

ASS?

228G-Amplifier Signal Select Query
Output the input signal type

Syntax: ASS?(x)

Parameter: none

Effect: outputs the type of amplifier input signal.

Response: *q1(y)*

q1	Input signal source for the amplifier
0	Amplifier input is switched to the zero signal.
1	Amplifier input is switched to the calibration signal.
2	Amplifier input is switched to the measurement signal.
3	Amplifier input is switched to temperature reference point (ML01 only).

9.4 Filter setting

AFS

Amplifier Filtering Select
Filter switching (fc 1/2)

Syntax: AFS p1(x)

Parameter:

p1	Filter code number
1	fc1

CAUTION

Only one more filter available.

Related commands: **AFS, ASF, SFC**
Amplifier type: **MC**

AFS?

Amplifier Filtering Select Query
Output the filter setting

Syntax: AFS?(x)

Parameter: none

Effect: outputs the set filter

Response: *q1(y)*

q1	Filter code number
1	fc1

ASF

Amplifier Signal Filtering
Input cut-off frequency and filter characteristics

Syntax: ASF p1,p2,p3(x)

Parameter:

p1	Filter code number
1	fc1
2	fc1

p2	filter frequency
1...n	Code number for the frequency value (corresponds to the index from the frequency table, which can be output using the command ASF?0).

p3	Filter characteristics
0	Bessel
1	Butterworth

CAUTION

Only one more filter available. Parameter p1=2 affects filter 1.

Related commands: **AFS, ASF, SFC**

Amplifier type: **MC**

ASF?

Amplifier Signal Filtering Query
Output cut-off frequency and filter characteristics

Syntax: ASF? p1(x)

Parameter:

p1	Filter code number
0	Frequency table (Bessel and Butterworth)
1	Filter fc1
2	Filter fc1

Response: If p1=0
q1,q2(y)

q1	Table Bessel frequencies
q2	Table Butterworth frequencies

If p1≠0
q1,q2,q3(y)

q1	Filter number fc1
q2	Cut-off frequency of filter fc1
q3	Filter characteristics (0 = Bessel, 1 = Butterworth)

CAUTION

Only one more filter available. Parameter p1=2 affects filter 1.

In the following tables you will find the available cut-off frequencies with Bessel or Butterworth characteristics depending on the particular amplifier.

The index corresponds to the frequency to be set (each element in the table is 5 characters long).

Index	**Bessel frequency (Hz)**						
	ML01	ML10	ML30	ML35	ML38	ML50/51/55	ML60
1	0.050	0.050	0.050	0.050	0.030	0.050	0.050
2	0.100	0.100	0.100	0.100	0.050	0.100	0.100
3	0.200	0.200	0.200	0.200	0.100	0.200	0.200
4	0.500	0.500	0.500	0.500	0.200	0.500	0.500
5	1.250	1.250	1.250	1.250	0.500	1.250	1.250
6	2.500	2.500	2.500	2.500	0.900	2.500	2.500
7	5.000	5.000	5.000	5.000	1.500	5.000	5.000
8	10.00	10.00	10.00	10.00		10.00	10.00
9	20.00	20.00	20.00	20.00		20.00	20.00
10	40.00	40.00	40.00			40.00	40.00
11	100.0	100.0	100.0			100.0	100.0
12	200.0	200.0				200.0	200.0
13	400.0	400.0				400.0	400.0
14	1100.	1000.				900.0	550.0

Index	Butterworth frequency (Hz)						
	ML01	ML10	ML30	ML35	ML38	ML50/51/55	ML60
1	5.000	5.000	5.000	5.000	1.000	5.000	5.000
2	10.00	10.00	10.00	10.00	1.500	10.00	10.00
3	20.00	20.00	20.00	20.00	2.500	20.00	20.00
4	40.00	40.00	40.00		3.000	40.00	40.00
5	80.00	80.00	80.00		5.000	80.00	80.00
6	250.0	250.0	200.0		6.000	250.0	250.0
7	500.0	500.0			7.000	500.0	500.0
8	1000.	1000.			10.00	1000.	1000.
9	2000.	2000.				1500.	
10	2400.						

9.5 Range

CDW

Calibration Dead Weight
Start zeroing/enter zero value (balance)

Syntax: CDW(x) or CDW p1(x)

Parameter:

q1	Zero value in ADC units
	7 680 000 ADC units correspond to the currently set full scale value (range). With linearised measurement of thermocouples and Pt10, Pt100, Pt1000 (ºC; ºF), the output value multiplied by 1/2560 corresponds to the zero value in degrees.

Note:

If the value read out with CDW?1 (zero value plus gross signal S1) is sent for p1, the currently applied measurement signal is set to zero. With linearised measurement of thermocouples, only the analogue output is set to zero. The setting has no effect on the display value.
You also have the opportunity to set basic detuning to zero, by calculating the zero value to be entered in accordance with the following equation :

$$\text{Zero value (ADC units)} = \frac{7\,680\,000 \times \text{basic detuning (unit)}}{\text{Full scale value (unit)}}$$

This is advisable, for example, if you want to measure a relative pressure with an absolute pressure transducer.

Related commands: **CDW, CMR, COF, IMR, ISR, MSV?, RMV?, STP, TEX**

Amplifier type: **MC**

Example:

The MGC*plus* is equipped with ML30 amplifiers. Absolute pressure transducers (SG full bridge transducers) are connected.

Transducer data:

Nominal sensitivity 2mV/V
Range 5bar

You should use the CDW command to enter a zero value, so that 0V is applied at the amplifier output.

How do you do this?

Action	Command
• Select Unit *bar*	ENU1,"BAR"(x)
• Set full scale value *2mV/V*	IMR1,2.0(x)
• Set input range *5bar*	IAD1,5000,3,1(x)

The transducer will now measure 1bar at ambient pressure.
Now use the following equation to calculate which value you have to enter to balance basic detuning.

Zero value = 7 680 000 x 1bar / 5bar = 1 536 000

Now enter:
CDW1536000(x)
0(y)
The ambient pressure is now balanced.

CDW?

Calibration Dead Weight Query
Output the zero value

Syntax: CDW? p1(x)

Parameter:

p1	Zero value code number
0	Zero value
1	Zero value plus current measured value (gross, S1)

Response: *q1(y)*

q1	Zero value in ADC units
	7 680 000 ADC units correspond to the currently set full scale value (range). With linearised measurement of thermocouples and Pt10, Pt100, Pt1000 (ºC; ºF, K), the output value multiplied by 1/2560 corresponds to the zero value in degrees.

Example 1: CDW?1(x)
10000(y)
Outputs zero value and gross signal (S1).
CDW 10 000(x) would now set this signal to zero.

Example 2: thermocouple measurement, PT10, PT100, PT1000
CDW?0(x)
256000(y)
Zero value = 256 000 / 2560 = 100 degrees

CMR

Change Measuring Range
Range switch (range 1/2)

Syntax: CMR p1(x)

Parameter:

p1	Range code number
1	Range 1

CAUTION

Only one more range available.

Related commands: **CDW, CMR, COF, IMR, ISR, MSV?, RMV?, STP, TEX**

Amplifier type: **MC**

CMR?

Change Measuring Range Query
Output the range

Syntax: CMR?(x)

Parameter: none

Effect: outputs the selected range.

Response: *q1(y)*

q1	Range code number
1	Range 1

IMR

Input Measuring Range
Enter the full scale values

Syntax: IMR p1,p2(x)

Parameter:

p1	Range code number
1	Range 1
2	Range 1

p2	Full scale value in mV/V
	With linearised measurement of thermocouples, the full scale value should be entered in degrees.

For ML60:

p2	Full scale value in kHz

CAUTION

Only one more range available. Parameter p1=2 affects range 1.

Related commands: **CDW**, **CMR**, **COF**, **IMR**, **ISR**, **MSV?**, **RMV?**, **STP**, **TEX**
Amplifier type: **MC**

IMR?

Input Measuring Range Query
Output the full scale values

Syntax: IMR? p1(x)

Parameter:

p1	Range code number
0	The gross signal (S1) in ADC units
1	Range 1
2	Range 1
3	Maximum and minimum full scale values which can be set

Response: *q1,q2(y)*

p1=0	q1=0, q2	Gross signal (S1) in ADC units
p1=1	q1	1 = range code number,
p1=2	q2	the set full scale value for range 1
p1=3	q1	maximum full scale value which can be set (dependent on amplifier)
	q2	minimum full scale value which can be set (dependent on amplifier)

CAUTION

Only one more range available. Parameter p1=2 affects range 1.

9.6 Taring

TAR

229G-Tare Instruction
Start taring/enter tare value

Syntax: TARp1(x)

Parameter: p1: tare value in ADC units

Input without parameters: the current measured value is adopted as the tare value.

Related commands: **TAR**, **TAT**, **TAV**

Amplifier type: **MC**

TAR?

230G-Tare Value Query
Output tare value

Syntax: TAR?(x)

Parameter: none

Response: *q1(y):* tare value in ADC units

Note:

You can use the following equation to convert the tare value to the relevant unit:

$$\text{Tare value (unit)} = \frac{\text{Full scale value (unit)} \times \text{tare value (ADC units)}}{7\,680\,000}$$

Conversion for thermocouples and Pt10, Pt100, Pt1000 (°C, °F, K):

$$\text{Value in degrees} = \frac{q1}{2560}$$

9.7 Analogue outputs

OPS

Output Path Select
Allocate analogue outputs

Syntax: OPS p1,p2(x)

Parameter:

p1	Output
1	Vo1
2	Vo2

p2	Signal
0	no connection
1	Gross (S1)
2	Net (S2)
3	Peak value store 1 (S3)
4	Peak value store 2 (S4)

Related commands: **OPS**, **OSP**, **SAO**

Amplifier type: **MC**

OPS?

Output Path Select Query
Query allocation of analogue outputs

Syntax: OPS? p1(x)

Parameter: p1: code number of the required output (see OPS command).

Response: *q1,q2(y)*

q1: code number of the output (see OPS command).

q2: code number of the allocated signal (see OPS command).

9.8 Peak store

PVS

Peak Value Select
Input the peak value store setup

Syntax: PVS p1,p2,p3,p4(x)

Parameter:

p1	Peak value store
1	Peak value 1 (S3)
2	Peak value 2 (S4)

p2	Peak value determination (always applies to both peak value stores)
0	Off
1	On

p3	Signal
–1	–Gross (–S1)
–2	–Net (–S2)
+1	+Gross (+S1)
+2	+Net (+S2)

p4	Envelope curves
0	Envelope function off
1..60000	Time constant in ms

Related commands: **CPV**, **HPV**, **MPV**, **PCM**, **PSM**, **PVS**
Amplifier type: **MC**

PVS?

Peak Value Select Query
Output the peak value store setup

Syntax: PVS?p1(x)

Parameter: p1: code number of the peak value store (see PVS command).

Response: *q1,q2,q3,q4(y)*

q1	Code number of the peak value store
q2	Peak value determination On/Off
q3	Code number of the signal with polarity
q4	Time constant for envelope function in ms

9.9 Limit value monitoring

LIV

231G-Limit Value
Input the limit value switching thresholds

Syntax: LIVp1,p2,p3,p4,p5(x)

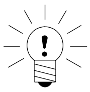
Parameter: p1: number of limit switches (1...4)

p2: operation (On=1 or Off=0)

p3	Input signal
1	Gross (S1)
2	Net (S2)
3	Peak value 1 (S3)
4	Peak value 2 (S4)
5	Combined peak value

p4: activation level in ADC units

p5: deactivation level in ADC units



Note:

Updating rate of limit switches: 1.2 kHz

Related commands: **LIV**, **LVD**, **LVL**, **LVS**, **SLC**, **SOP**

Amplifier type: **MC**

Example 1: LIV1,1,2,3840000,3072000(x)
0(y)
Limit switch 1 is set to limit value monitoring (On), net input signal (S2) and to switching points +5V (make contact) or +4V (break contact). Hysteresis 1V.

Example 2: for linearised measurement of thermocouples:
LIV1,1,2,512000,256000(x)
0(y)
Limit switch 1 is set to limit value monitoring (On), net input signal (S2) and to switching points 200 degrees (make contact) or 100 degrees (break contact). Hysteresis 100 degrees.

Note:

You can use the following equations to convert the switching points to ADC units

$$\text{Switching point (ADC units)} = \frac{7\,680\,000 \ \text{x switching point (unit)}}{\text{Full scale value (unit)}}$$

With linearised measurement of thermocouples and resistance thermometer (^{o}C, ^{o}F, K):

$$\text{Switching point (ADC units)} = \text{Switching point (Grad) x } 2560$$

LIV?

Limit Value Query
Output the limit value switching thresholds

Syntax: LIV? p1,p2(x)

Parameter:

p1	Limit switches (status)
0	Query the signal value of p2 (output in ADC units)
1...4	1...4

p2	Signal code number, if p1=0
1	Gross (S1)
2	Net (S2)
3	Peak value 1 (S3)
4	Peak value 2 (S4)
5	Combined peak value

Response: *q1,q2,q3,q4,q5(y)* or *q6(y)*

q1	Limit switches (status)
q2	Limit value monitoring On/Off
q3	Limit switch input signal
q4	Switching point 1 (make contact) in ADC units
q5	Switching point 2 (break contact) in ADC units
q6	Level value of signal p2 in ADC units

Note:

You can use the following equations to convert the switching points or the level values into the selected unit.

$$\text{Switching point (units)} = \frac{\text{Full scale value (unit)} \quad \text{x switching point (ADC units)} =}{7\ 680\ 000}$$

$$\text{Level value (unit)} = \frac{\text{Full scale value (unit)} \quad \text{x level value (ADC units)}}{7\ 680\ 000}$$

With linearised measurement of thermocouples and resistance thermometer (^{o}C, ^{o}F, K) the output value multiplied by 1/2560 corresponds to the level value in degrees.

9.10 Transferring the amplifier setup and the comment

MDD

Memory Device Data
Input the amplifier setup data

Syntax: MDD p1(x)

Parameter: p1

Amplifier setup data, retrieved from the amplifier with the command MDD?, as a hexadecimal string "___", (approx. 123 bytes = 246 characters).

Effect: the command is used to save and recall entire setups. If you want to change individual parameters, please use the relevant command (e.g. IMR). Amplifier setup parameters are recalled to the amplifier channel, this is then followed by a warm start. The amplifier channel number which is the intended destination for the setup data, is coded in the setup data.

Related commands: **MDD**, **TDD**, **UCC**
Communications processor: **CP12, CP32**

MDD?

Memory Device Data Query
Output the amplifier setup data

Syntax: MDD? p1(x)

Parameter: p1: channel number of amplifier (1-16)

Effect: outputs the amplifier setup parameters

Response: *"___hexadecimal string___"(y):* approx. 123 bytes =246 characters)

9.11 Remote control

RFP

Remote Function Programming
Assign the remote functions

Syntax: RFP " . . . . "(x)

Parameter: Pin 9 Pin2 Bu2 of the connection boards

Code number	Function Function	ML01	ML35	ML38	ML10/30/50/51/55 ML55S6	ML60
0	Autocalibration Off/On	ACAL	ACAL	ACAL	ACAL	ACAL
1	Initiate taring	TARE	TARE	TARE	TARE	TARE
2	Switch range 2/1	NOP?[3)]	NOP?	NOP?	NOP?	NOP?
3	Switch frequency 2/1	NOP?	NOP?	NOP?	NOP?	NOP?
4	Current value/peak value 1	CPV1	CPV1	CPV1	CPV1	CPV1
5	Peak value 1 (HOLD)	HLD1	HLD1	HLD1	HLD1	HLD1
6	Current value/Peak value 2	CPV2	CPV2	CPV2	CPV2	CPV2
7	Peak value 2 (HOLD)	HLD2	HLD2	HLD2	HLD2	HLD2
8	Zero amplifier	ZERO	ZERO	ZERO	ZERO	ZERO
9	REMOTE/LOCAL-switchover	REMT	REMT	REMT	REMT	REMT
A	Shunt on/off	NOP[1)]	NOP	SHNT	SHNT	SHNT[2)]
B	Print On/Off	PRNT	PRNT	PRNT	PRNT	PRNT
C	Switch on calibration signal	CAL	CAL	CAL	CAL	CAL
D	Activate zero signal	NULL	NULL	NULL	NULL	NULL
E	Operating sign change	NOP	INV	–	INV	INV
F	Trigger for measurement store	NOP	NOP	–	NOP	NOP

1) NOP = no function
2) Activates the calibration signal of the torque shaft
3) Functions no longer supported by MGC*plus*, are acknowledged with NOP?.

Note: the REMOTE-LOCAL switching option is always available even if the device has LOCAL status.

Related commands: **LOR**, **RFP**, **RIF**, **RIP?**, **DFL**
Amplifier type: **MC**

RFP?

Remote Function Programming Query
Query assignment of remote functions

Syntax: RFP? p1(x)

Parameter:

p1	
0	Output assignment of remote functions
1	Output table of available functions

Response: *" . . . . "(y):* maximum of 64 characters

9.12 Output format, measurement output

COF

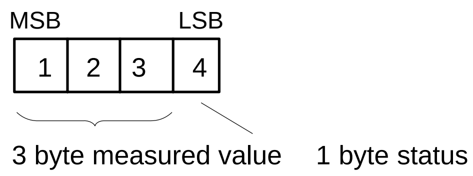
Change Output Format
Define the measured value output format (for MSV? command)

Syntax: COF p1(x)

Parameter:

p1	Measurement output format
0	Measured value, channel, status (ASCII format)
1	Measured value (ASCII format)
2	4 byte binary measurement output (MSB XX XX LSB)
3	4 byte binary measurement output (LSB XX XX MSB)
4	2 byte binary measurement output (MSB LSB)
5	2 byte binary measurement output (LSB MSB)

4 byte output:



7 680 000 ADU units = final scale value (unit)
In the case of thermocouples and the units °C,°F, K corresponds to the output value multiplied by 1 / 2560 of the measured value in degrees.

Bit:	7	6	5	4	3	2	1	0	Status byte

1	Limit switch 1 On
2	Limit switch 2 On
4	Limit switch 3 On
8	Limit switch 4 On
16	Gross overflow (S1)
32	Net overflow (S2)
64	Calibration error
128	Setup modification (e.g. using remote control signals), Value not available

2 byte output

MSB	LSB
1	2

2 byte measured value

30 000 = final scale value (unit)
In the case of thermocouples and the units °C,°F, K corresponds to the output value multiplied by 1 / 10 of the measured value in degrees.

Note:
This command always applies to all channels of a device.

Note:

With ASCII format, the measured values are output scaled, with binary format in ADC units.

Related commands: **CDW, CMR, COF, IMR, ISR, MSV?, RMV?, STP, TEX**

Communications processor: **CP12, CP32**

COF?

Measuring Value Format Query
Query measurement output format

Syntax:	COF?(x)
Parameter:	none
Response:	*q1(y):* code number of output format (see COF command).

MSV?

Measuring Signal Value Query

Output the measured value

Syntax: MSV? p1,p2,p3(x)

Parameter:

p1	Signal	
1	S1	Gross
2	S2	Net
3	S3	Peak value 1
4	S4	Peak value 2
5	Limit1	Make contact
6	Limit1	Break contact
7	Limit2	Make contact
8	Limit2	Break contact
9	Limit3	Make contact
10	Limit3	Break contact
11	Limit4	Make contact
12	Limit4	Break contact
13	S1	Gross, dynamic
14	S2	Net, dynamic

p1 = 1, 2: with display filtering

p1 = 13, 14: without display filtering

p2	Number of measured values
0	Continuous sending
1 ... 65 535	Default = 1

Parameter: p3

Sequence time in seconds 0.1s to 60.0s. Output time in seconds between measured values (only in the case of binary output).

Effect: the measured value of the required signal (p1) is output. The output format depends on the last COF and TEX command. You use the PCS command to specify from which channel the measured values are to be output.

Response: *measured value(y):*output format see COF command.

Example 1: **Output in full format ASCII**

PCS3,5(x) 'Select Channels 3 and 5
0(y)

TEX44,59(x) Separator ',' and ';'
0(y)

COF0(x) Full-format ASCII (measured value, channel, status)
0(y)

Fetch a gross measured value (S1).
MSV?1(x)
9.998,3,0,8.888,5,0(y)

Status byte
channel
measured value (e.g. 9.998kg)

Fetch 3 net measured values (3 measured values each from channels 3 and 5).
MSV?2,3(x)
9.998,3,0,8.888,5,0;9.999,3,0,8.889,5,0;9.998,3,0,8.888,5,0(y)

For thermocouples:
Fetch a measured value.
MSV?1(x)
100.00,3,0,200.00,5,0(y)

Status byte
channel
measured value (e.g. 100 degrees)

Example 2: **Output in short-format ASCII**

Short-format ASCII (measured value only), channels as in example 1. Fetch one gross measured value from channels 3 and 5.
COF1(x)
0(y)

MSV?1(x)
9.998,8.888(y)

Fetch 3 net measured values.
MSV?2,3(x)
9.998,8.888;9.998,8.888;9.998,8.888(y)

Send gross measured value continuously
MSV?1,0
9.998,8.888;9.998,8.888;9.998,8.888;.......

STP(x)Abort output

Example 3: **Output in 4 byte binary format**

Binary 4 byte format, the same channels as in example 1. One gross measured value from channels 3 and 5
COF2(x)
0(y)
MSV?1(x)

#18ffeedd00aabbcc00(y)

Binary format:
- 1 byte status from channel 5
- 3 byte measured value from channel 5
- 1 byte status from channel 3
- 3 byte measured value from channel 3

ASCII format:
- y
- x
- z

z: Identification for binary output
x: specifies the number of decimal places of y (x=0 for continuous output)
y: specifies the number of binary bytes to follow (not applicable to continuous output)

e.g.:	x = 1	y = 8	(outputs 8 binary bytes)
	x = 2	y = 16	(outputs 16 binary bytes)

Example 3.1: **Continuous output**

Gross measured value from channels 3 and 5 are output continuously.
MSV?1,0(x)
#0ffeedd00aabbcc00...
STP(x) Abort output

Example 3.2: **Continuous output in a fixed time frame**

Channel 1 selected
PCS1(x)
0(y)

Binary measurement output (4 byte)
COF2(x)
0(y)

Continuously output gross measured value with a 0.1s (100ms) repetition time. 10 measured value are output every second.
MSV?1,0,0.1(x)
#0ffeedd00ffeedd00...
STP(x) Abort output

Related commands: **CDW, CMR, COF, IMR, ISR, MSV?, RMV?, STP, TEX**

Communications processor: **CP12, CP32**

9.13 Display functions

Note:

The commands in this ”Display functions” chapter affect how the measured values are displayed on the AB22A/AB32 display and control panel.

ENU

Engineering Unit
Input of unit

Syntax: ENU p1,”p2”(x)

Parameter:

p1	Range for which the unit applies
1	Range 1
2	Range 1

p2	The required unit as a string
	4 characters

CAUTION

Only one more range available. Parameter p1=2 affects range 1.

Related commands: **ENU**, **IAD**, **ICS**, **ISS**, **‹SIS**, **STL**
Amplifier type: **MC**

ENU?

Engineering Unit Query
Output of unit

Syntax: ENU? p1(x)

Parameter:

p1	Unit
0	Unit of the range currently set
1	Unit of range 1
2	Unit of range 1
3	Table of possible units

Response: *q1,q2(y)*

q1: range no. (range 1/2)

q2: a string comprising a maximum of 4 characters. The string characters can be found in the table on the next page.

When p1=3, the string length corresponds to the table of possible units

Significance:

MV/V	=	mV/V	PSI	=	PSI
V	=	V	uM	=	µm
G	=	g	MM	=	mm
KG	=	kg	CM	=	cm
T	=	T	M	=	M
KT	=	kt	INCH	=	inch
TONS	=	tons	NM	=	Nm
LBS	=	lbs	FTLB	=	ftlb
N	=	N	INLB	=	inlb
KN	=	kN	uM/M	=	µm/m
BAR	=	bar	M/S	=	m/s
mBAR	=	mbar	M/SS	=	m/s^2
PA	=	PA	p/o	=	%
PAS	=	PAS	p/oo	=	%o
HPAS	=	HPAS	PPM	=	ppm
KPAS	=	KPAS	NC	=	nC
UMIN	=	U/min	PC	=	pC
*pC	=	Grad C	V_IN	=	V
*pF	=	Grad F	MW	=	MW
*K	=	Kelvin	KW	=	kW
*V	=	V	W	=	Watt
mV	=	mV	mA	=	mA
A	=	A			

* With thermocouples, only these marked units will be output for the table of possible units.

Possible units for the amplifier **ML01**:
”pC_ _pF_ _K_ _ _V_ _ _NC_ _mV_ _ A_ _ _mA_ _W_ _ _KW_ _ MW_ _G_ _
_KG_ _ T_ _ _KT_ _TONSLBS_ N_ _ _KN_ _ BAR_mBARPA_ _
PAS_HPASKPASPSI_UM_ _ MM_ _ CM_ _M_ _ _INCHNM_ _
FTLBINLBUM/MM/S_M/SSp/o_p/ooPPM_UMIN_ _ _ _V_IN_NC_ _ PC_ _”

Possible units for the amplifier **ML10:**
”MV/V_V_ _ _G_ _ _KG_ _T_ _ _KT_ _TONSLBS_N_ _ _KN_ _BAR_mBARPA_ _
PAS_HPASKPASPSI_uM_ _MM_ _CM_ _M_ _ _INCHNM_ _KNM_
FTLBINLBUM/MM/S_M/SSp/o_p/ooPPM_ _ _ _ _V_INNC_ _PC_ _”

Possible units for the amplifiers **ML30, ML38, ML50, ML51, ML55**:
”MV/VV_ _ _G_ _ _KG_ _T_ _ _KT_ _TONSLBS_N_ _ _KN_ _BAR_mBARPA_ _
PAS_HPASKPASPSI_uM_ _MM_ _CM_ _M_ _ _INCHNM_ _
FTLBINLBUM/MM/S_M/SSp/o_p/ooPPM_ _ _ _ _”

Possible units for the amplifier **ML35**:
”pC_ _pF_ _K_ _ _V_ _ _OHM_ KOHM_mV_ _ A_ _ _mA_ _W_ _ _KW_ _ MW_ _
G_ _ _KG_ _ T_ _ _KT_ _TONSLBS_ N_ _ _KN_ _ BAR_mBARPA_ _ PAS_
HPASKPASPSI_UM_ _ MM_ _ CM_ _M_ _ _INCHNM_ _
FTLBINLBUM/MM/S_M/SSp/o_p/ooPPM_UMIN_ _ _ _ _”

Possible units for the amplifier **ML60**:
”V_ _
_kHZ_HZ_1/S_RPM_UMINIMP_KIMPDEG_NM_KNM_FTLBINLB_M/S_KM/HM/SS
_G_ KPAS_ _KG_ _T_ _ _KT_ _TONSLBS_N_ _ _KN_ _BAR_mBARPA_ _
PAS_HPASPSI_UM_ _MM_ _CM_ _M_ _ _INCHuM/Mp/o_p/ooPPM_ _ _ _ _”

IAD

Indication Adaptation
Enter input range, decimal point, step

Syntax: IAD p1,p2,p3,p4(x)

Parameter:

p1	Range 1
p2	Input range without the decimal point
p3	Decimal point (number of decimal places)
p4	Step (minimum increment as a digit)

Step p4 can take the following values:

p4	Step
1	1
2	2
3	5
4	10
5	20
6	50
7	100
8	200
9	500
10	1000

CAUTION

Only one more range available. Parameter p1=2 affects range 1.

Related commands: **ENU**, **IAD**, **ICS**, **ISS**, **SIS**, **STL**
Amplifier type: **MC**

IAD?

Indication Adaptation Query
Output input range, decimal point, step

Syntax: IAD? p1(x)

Parameter:

p1	Status
1	of range 1
2	of range 1

Response: *q1,q2,q3,q4(y):* for parameters, see IAD command

CAUTION

Only one more range available. Parameter p1=2 affects range 1.

ISS

Indication Signal Select
Selection of channel and signal display on AB22A/AB32

Syntax: ISS p1,p2(x)

Parameter: p1: channel number (1...16)

p2: signal code number

p2	Signal that is to be displayed
0	no measured value
1	Gross (S1)
2	Net (S2)
3	Peak value 1 (S3)
4	Peak value 2 (S4)
5*	Limit switch 1 (Limit1)
7*	Limit switch 2 (Limit2)
9*	Limit switch 3 (Limit3)
11*	Limit switch 4 (Limit4)

*) Only if limit switches are on

Related commands: **ENU**, **IAD**, **ICS**, **ISS**, **SIS**, **STL**
Communications processor: **CP12, CP32**

ISS?

Indication Signal Select Query
Output the channel and the signal display on the AB22A/AB32

Syntax: ISS?(x)
Parameter: none
Response: *q1,q2(y):* channel number, signal code number

9.14 Print functions

PFS

232G-Print Format Select
Specify print format

Syntax: PFS p1(x)

Parameter:

p1	Signal that is to be printed
0	Display in the AB22A/AB32
1	Gross (S1)
2	Net (S2)
4	Peak value 1 (S3)
8	Peak value 2 (S4)
16	Limit switches (status)
64	Disable print function (PRNT OFF)

You can set any signal combination, by adding coding numbers.

Example 1: S1 and S2 are to be printed
p1= 1 + 2 = 3
PFS3(x)
0(y)

Example 2: Activation level of limit switch 1, channel 1 is to be printed.
ISS1,5(x)
0(y)
Show level in the AB22A/AB32 display (only if the limit switch is activated).
PFS0(x)
0(y)
Select the AB22A/AB32 display to print

Related commands: **LPR, PFS, PRF, PRT, SPP**

Amplifier type: **MC, MLxx**

PFS?

233G-Print Format Select Query
Query the print format

Syntax 1:	PFS?(x)
Parameter:	none
Response:	*q1(y):* signal or signal combination set with the PFS command. For coding, see PFS command.
Syntax 2:	PFS??(x)
Parameter:	none
Response:	*q1..qn(y):* any possible signal combination (0,1,2,3,4...31,64)
Example:	PFS?(x) *1(y):* signal 1 is to be printed

PRT

Print
Print values

Syntax: PRT p1(x)

Parameter:

p1	Print Signal
1	Gross (S1)
2	Net (S2)
4	Peak value 1 (S3)
8	Peak value 2 (S4)
16	Limit switches (status)

You can set any signal combination, by adding coding numbers.

If p1 is not sent:

The signal output is that set with the AB22A/AB32 or from the computer (PFS command).

Effect: sends the print data from selected channels to the interface specified by the SPP command.

Related commands: **LPR, PFS, PRF, PRT, SPP**

Communications processor: **CP12, CP32**

Example 1: An external computer is connected to the RS232C interface. The RS485 printer interface should be used for printing.

SPP1306(x)
0(y)

Select printer interface

CHS16(x)
0(y)

Select channel 5

PRT1(x)
0(y)

Prints S1 from channel 5

Example 2: PRT31(x)
0(y)

Prints S1, S2, S3, S4 and limit switch (status) from channel 5.

10 Profibus

EST?

Error Status Query
Read out identification string

Syntax:	EST? p1(x)
Parameter:	p1: (UINT16)
Effect:	the command is used to read out the identification string for a parameter value.
Response:	p1 = 0:
	q1,...qn: IDS value(s) of the error(s)
	p1=1:
	q1, q2
	q1 !=0: an error exists
	q2 !=0: a warning exists

Amplifier type: **ML77**

Related commands: **EST?, IDS?**

IDS?

Identifier Settings Query
Query the read out identification string

Syntax:	IDS? p1(x)
Parameter:	p1: (UINT16) value of the parameter
Effect:	the command is used to read out the identification string for a parameter value.
Response:	q1: identification string in the set language

Amplifier type: **ML77**

11 CAN-bus

NAS?

Neighbourhood active subchannel Query
Output neighbourhood subchannel

Syntax:	NAS?(x)	
Effect:	the command is addressed to a subchannel and returns the adjoining subchannels. If there is no lower or higher active subchannel available, 0 is downloaded for the corresponding parameter.	
Response:	p1(INT16):	next lowest active subchannel
	p2(INT16):	next lowest active subchannel

Related commands: **CPI?, CPO, DBA, DBE?, DBS?, FCL, FLO, FLW, NAS?, OSP, REV?, SAO**

Amplifier type: **ML71**

REV?

Revision Query
Output revision

Syntax: REV? p1(x)

Parameter: p1: 0:

0: revision ML hardware

1: revision AP hardware (partly readable)

Response: p1: (IEEEString) 4 Character Revision

Amplifier type: **ML71**

Related commands: **CPI?, CPO, DBA, DBE?, DBS?, FCL, FLO, FLW, NAS?, OSP, REV?, SAO**

11.1 File commands

FLO

File open
Open file

Syntax: FLOp1,p2(x)

Parameter: p1: (UNIT32) file size in bytes

p2: (UNIT16) type of file (IDS_FopenBin=1700,
IDS_FopenAsc=1701

Effect: the command opens a file in ML71. If this file is not available as yet, it is created.

Response: 0: command successfully executed

?: an error has occurred

Related commands: **CPI?**, **CPO**, **DBA**, **DBE?**, **DBS?**, **FCL**, **FLO**, **FLW**, **NAS?**, **OSP**, **REV?**, **SAO**

Amplifier type: **ML71**

FLW

File Write
File/Block download

Syntax: FLWp1,p2(x)

Parameter: if opened as a binary file:

p1: (INT16) block number

p2: (HEXASC) data block

if file opened in ASCII mode:

Effect: a file must be opened. In binary mode, a 128-byte block is downloaded with block number p2.

In ASCII mode, a line of a text file is downloaded.

Caution!

In ASCII mode, upper quotation marks (ASCII code 34) must be displayed by ASCII code 127.

Response: 0: command successfully executed

?: an error has occurred

Related commands: **CPI?, CPO, DBA, DBE?, DBS?, FCL, FLO, FLW, NAS?, OSP, REV?, SAO**

Amplifier type: **ML71**

FCL

File Close
Close file and delete

Syntax: FCL(x)

Parameter: none

Effect: a file must be opened. The open file is closed and deleted.

Response: 0: command successfully executed

?: an error has occurred

Related commands: **CPI?, CPO, DBA, DBE?, DBS?, FCL, FLO, FLW, NAS?, OSP, REV?, SAO**

Amplifier type: **ML71**

11.2 Parameter assignment for CAN data acquisition

DBE?

Data Base Error Query
Query Data base Errors

Syntax: DBE?(x)

Parameter: none

Effect: the command returns detailed information about errors that can occur when the CQT command is executed.

Response: *q1,q2(y)*

q1	q2	Meaning
=0	=0	no error occurred
>0	>0	q1= row and q2= column number of the DBC file where an error has occurred
–1	x	3
–1	x	4

Related commands: **CPI?**, **CPO**, **DBA**, **DBE?**, **DBS?**, **FCL, FLO**, **FLW**, **NAS?**, **OSP**, **REV?**, **SAO**

Amplifier type: **ML71**

DBA

Data Base
Create database

Syntax: DBAp1,p2(x)

Parameter: p1: (UINT16) database number
p2: (UINT16) IDS_NewDatabase, IDS_DeleteDatabase

Effect: a database file is read in and used to create a new database for internal use or to delete an existing database. When a new database is defined, this is stored in flash memory.

Related commands: **CPI?**, **CPO**, **DBA**, **DBE?**, **DBS?**, **FCL, FLO**, **FLW**, **NAS?**, **OSP**, **REV?**, **SAO**

Amplifier type: **ML71**

Response: 0: command successfully executed
?: an error has occurred

Amplifier type: **ML71**

DBA?

Data Base Query
Query the database

Syntax: DBA? p1(x)

Parameter: p1: (UINT16) database number

Effect: the command queries whether there is already an internal database under the given number.

Response: 0: database exists
1: no database exists

DBS?

Data Base Signal Query
Output parameters from the database

Syntax:	DBS? p1,p2(x)
Parameter:	p1: (UINT16) database number
	p2: (UINT16) signal number
Effect:	the parameters of a signal are output from the database.
Response:	p1: message name
	p2: signal name
	p3: CAN ID
	p4: mode
	p5: start bit
	p6: length
	p7: scaling
	p8: offset
	p9: unit
	p10: frame format
	p11: value format

Related commands: **CPI?**, **CPO**, **DBA**, **DBE?**, **DBS?**, **FCL**, **FLO**, **FLW**, **NAS?**, **OSP**, **REV?**, **SAO**

Amplifier type: **ML71**

11.3 Exchange of CAN messages

CPO

CAN Protocol Output
Output CAN protocol

Syntax:	CPOp1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p1 1(x)
Parameter:	p1: (UINT8) number of the CAN interface (1 or 2)
	p2: (UINT32) CAN identifier
	p3: (UINT8) length of the CAN message (0...8)
	p4: (UINT8) byte 1
	p5: (UINT8) byte 2
	p6: (UINT8) byte 3
	p7: (UINT8) byte 4
	p8: (UINT8) byte 5
	p9: (UINT8) byte 6
	p10: (UINT8) byte
	p11: (UINT8) byte 8
Effect:	output of a CAN protocol at one of the two interfaces.
Response:	0: command successfully executed
	?: an error has occurred

Related commands: **CPI?**, **CPO**, **DBA**, **DBE?**, **DBS?**, **FCL**, **FLO**, **FLW**, **NAS?**, **OSP**, **REV?**, **SAO**

Amplifier type: **ML71**

CPI?

CAN Protocol Input Query
Output CAN message

Syntax: CPOp1,p2,p3(x)

Parameter: p1: (UINT8) number of the CAN interface (1 or 2)

p2: (UINT32) CAN identifier of the receive message

Effect: output of any CAN message which has been received with the corresponding ID(p2). If no suitable message has been received, 0 is output. However, once messages have been received, you can only call them for a certain period of time (dependent on utilisation of bus capacity), in other words, only until the entry in the CAN input buffer is overwritten (buffer size 50).

Response: q1: (UINT8) length of the receive message

q2: (UINT8) byte 1

q3: (UINT8) byte 2

q4: (UINT8) byte 3

q5: (UINT8) byte 4

q6: (UINT8) byte 5

q7: (UINT8) byte 6

q8: (UINT8) byte 7

q9: (UINT8) byte 8

q10: (UINT8) timestamp[6]

or

?: error

0: no message received

Related commands: **CPI?**, **CPO**, **DBA**, **DBE?**, **DBS?**, **FCL**, **FLO**, **FLW**, **NAS?**, **OSP**, **REV?**, **SAO**

Amplifier type: **ML71**

11.4 Assigning analogue output parameters

OSP

Output Select Path
Allocate analogue outputs

Syntax: OSPp1,p2(x)
Parameter: p1: (UINT16) analogue output number (code number of the analogue output)
p2: signal source
200: no signal
219: signal set with SAO
220: signal of the selected subchannel
Effect: a signal is assigned to the analogue outputs.
Response: 0: command successfully executed
?: an error has occurred

Related commands: **CPI?, CPO, DBA, DBE?, DBS?, FCL, FLO, FLW, NAS?, OSP, REV?, SAO**

Amplifier type: **ML71**

OSP?

Output Select Path Query
Query allocation of analogue outputs

Syntax: OSP?p1(x)
Parameter: p1: (UINT16) signal source (see above)
p2: signal source
Effect: a signal is assigned to the analogue outputs.
Response: q1: assigned subchannel number (1...32)

SAO

Set Analogue Output
Output voltage

Syntax:	SAOp1,p2(x)
Parameter:	p1: (UINT16) analogue output
	p2: output voltage in V
Effect:	outputs the voltage applied to an analogue output by SAO
Response:	p1: analogue output
	p2: voltage

Related commands: **CPI?**, **CPO**, **DBA**, **DBE?**, **DBS?**, **FCL**, **FLO**, **FLW**, **NAS?**, **OSP**, **REV?**, **SAO**

Amplifier type: **ML71**

SAO?

Set Analog Output Query
Query the output voltage

Syntax:	SAO?p1(x)
Parameter:	p1: (UINT16) analogue output
Effect:	outputs the voltage applied to an analogue output by SAO
Response:	p1: analogue output
	p2: voltage

Specifying the output subchannel and the output characteristics

The subchannel output at the analogue output and the output characteristics are assigned parameters in dialogue box 3 ”Output Characteristics”.

The TDD command in ML71

Although ML71 is a multi-channel amplifier, all TDD commands **always apply to the entire module.** Even if a TDD is addressed to one subchannel, the settings will be saved or loaded for all subchannels.

For reasons of memory space, the device only has **one** parameter set, in contrast to the eight parameter sets of other modules.

11.5 Using function keys to send CAN messages

You can configure 10 function keys. Assign parameters to the message which is to be sent once or cyclically, by pressing the function key.

Sending the CAN message once

To make sure that only one message is only sent when a function key is pressed, the transmission period must be set to 0ms when parameters are assigned. Each time the function key is pressed, the CAN message is sent once.

Cyclic transmission of CAN messages

A transmission period of 0ms must be set here. The longest possible transmission period which can be set is 65535ms.

When the function key is pressed once, cyclic transmission of the CAN message starts with the period set. The next time you press the key, transmission is interrupted until it is restarted by pressing the key again.

12 Multi-channel I/O module ML78

12.1 File commands

FLO

File open

Syntax: FLOp1,p2

Parameters: p1: (UNIT32) File size in bytes

p2: (UNIT16) File type (IDS_FopenBin=1700, IDS_FopenAsc=1701)

Effect: The command opens an existing file in the ML78. If this file does not already exist, it is created.

Response: 0: Command has been executed

?: An error has occurred

Related commands: **FLW**, **FCL**, **DBA**, **DBE?**, **SAO**, **OSP**, **SOP**, **SPF?**, **RIP?**, **ROP?**, **TDD**, **TDD?**

FLW

File Write
File/block transfer (a file must be open before this command is used)

Syntax: FLWp1,p2(x)

Parameters: If binary file open:

p1: (INT16) Block number

p2: (HEXASC) Data block

If file in ASCII mode open:
p1: (IEESTR) Line (max. 245 characters)

Effect: In binary mode a 128–byte block with block number p2 is transferred.

In ASCII mode a line of a text file is transferred.

Caution!

In ASCII mode, high quotation marks (ASCII code 34) must be represented by ASCII code 127.

Response: 0: Command has been executed

?: An error has occurred

Related commands: **FLO**, **FCL**, **DBA**, **DBE?**, **SAO**, **OSP**, **SOP**, **SPF?**, **RIP?**, **ROP?**, **TDD**, **TDD?**

FCL

File Close
Close and delete file (a file must be open before this command is used)

Syntax:	FCL(x)
Parameters:	None
Effect:	The open file is closed and deleted.
Response:	0: Command has been executed
	?: An error has occurred

Related commands: **FLO**, **FLW**, **DBA**, **DBE?**, **SAO**, **OSP**, **SOP**, **SPF?**, **RIP?**, **ROP?**, **TDD**, **TDD?**

DBA

Data Base
Create or delete database

Syntax:	DBAp1, p2
Parameters:	p1: (UNIT16) Database number
	p2: (UINT16) IDS_NewDatabase (1720), IDS_DeleteDatabase (1721), IDS_NewDB_Parts(1722)
Effect:	A curve for internal use by the system is created for the function generator from an imported curve file or an existing curve is deleted. When a new curve is created it is saved to the Flash memory.
Response:	0: Command has been executed ?: An error has occurred

Related commands: **FLO**, **FLW**, **FCL**, **DBE?**, **SAO**, **OSP**, **SOP**, **SPF?**, **RIP?**, **ROP?**, **TDD**, **TDD?**

DBE?

Data Base Error Query
Error typeout

Syntax: DBE?p1

Parameters: No parameters for error query

p1 = 1 : Save the curve in parts (DBA<No>,1722)

Response:
0 : All parts saved
1 : DBE?1 must be sent again, since the database has not yet been saved in full

Effect: Output giving detailed information on errors that may have occurred when the DBA command was being executed.

Response: *q1,q2(y)*

q1	q2	Meaning
=0	=0	No error occurred
>0	>0	q1=line number and q2=column number of the DBC file in which an error occurred
–1	x	No file open
–2	x	A table with the specified number already exists
–3	x	The database to be deleted does not exist
–4	X	Insufficient memory space (Flash or RAM)

Related commands: **FLO**, **FLW**, **FCL**, **DBA**, **SAO**, **OSP**, **SOP**, **SPF?**, **RIP?**, **ROP?**, **TDD**, **TDD?**

12.2 Other commands

SAO

Set Analog Output

Syntax:	SAOp1,p2(x)
	p1 = Number of analogue output 1...10 1 = analogue output 1: BNC or VO1 AP78 or VO1 AP75 2 = analogue output 2: VO2 AP78 or VO2 AP75 3...10 = AO3 ... AO10 AP78
	p2: Voltage from –10V ... +10V
Effect:	Sets analogue output n to the required output voltage
Response:	0: Command has been executed
	?: An error has occurred
Requirement:	The output concerned must have been set up for an “External” source (see command OSP)
Example:	SAO3,–5.680 Analogue output AO3 is set to –5.68 V

Related commands: **FLO**, **FLW**, **FCL**, **DBA**, **DBE?**, **OSP**, **SOP**, **SPF?**, **RIP?**, **ROP?**, **TDD**, **TDD?**

OSP

Output Select Path
Assign analogue outputs

Syntax: OSPp1,p2(x)

p1 = Number of analogue output 1...10
1 = analogue output 1: BNC or VO1 AP78 or VO1 AP75
2 = analogue output 2: VO2/AP78 or VO2/AP75
3...10 = AO3 ... AO10/AP78

p2: Signal according to table

IDS constant	Value	Meaning
IDS_ANASRC_NULL	1908	Output fixed at 0V
IDS_ANASRC_EXT	1909	Output voltage adjustable using software command SAO
IDS_ANASRC_CHAN	1910	Output voltage linked to MGC signal
IDS_ANASRC_FUNC1	1911	Output linked to function generator 1
:	:	:
:	:	:
IDS_ANASRC_FUNC10	1920	Output linked to function generator 10

Effect: The signal source is assigned to analogue output x.

Response: 0: Command has been executed

?: An error has occurred

Example: OSP4, 1909
Signal source “External” is assigned to analogue output AO4.

Related commands: **FLO**, **FLW**, **FCL**, **DBA**, **DBE?**, **SAO**, **SOP**, **SPF?**, **RIP?**, **ROP?**, **TDD**, **TDD?**

SOP

Set Output
Set a logical level for the analog outputs

Syntax: SOPp1,p2(x)

p1 = Number of the analogue output it is intended to set
1 ... 8 output A.1 ... A.8 on the AP75 (slot at rear of ML78)
9 ... 16 output B.1 ... B.8 on the AP75 (slot alongside the ML78)

p2 = 0 = Off, 1= On

Effect: Sets output p1 to the required logical level

Response: 0: Command has been executed

?: An error has occurred

Requirement: In the “Output” menu on the AB22A or on the Assistant for the output concerned, the option “External” must have been selected with YES. If there is no intention to set up a link with any other MGC signals, no channel selection string must be specified in the other fields.

Example: SOP1, 1
Output A.1 is set to 1.

Related commands: **FLO**, **FLW**, **FCL**, **DBA**, **DBE?**, **SAO**, **OSP**, **SPF?**, **RIP?**, **ROP?**, **TDD**, **TDD?**

SPF?

Special Functions Query
Query the special properties of an amplifier

Syntax: SPF?

Parameters: None

Effect: To find out the special properties of amplifiers, it is possible to request a 32–bit value containing the relevant information in bit-coded form.

Response: 61

Bit0 Multimaster

Bit1 Asynchronous module

Bit2 Generally configurable

Bit3 Special handling with regard to saving a parameter set to hard disk or to the Assistant.

Bit4 TDD once only per module, not for each individual subchannel

Bit5 Synchronous link data

Related commands: **FLO**, **FLW**, **FCL**, **DBA**, **DBE?**, **SAO**, **OSP**, **SOP**, **RIP?**, **ROP?**, **TDD**, **TDD?**

RIP?

Read Input Query
Import the logical levels of the inputs

Syntax: RIP?
Parameters: None
Effect: Imports the logical level of the AP75 inputs from slots A and B. 0=0V, 1=24V
Response: q1: Integer value with the following contents

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Input	B.8	B.7	B.6	B.5	B.4	B.3	B.2	B.1	A.8	A.7	A.6	A.5	A.4	A.3	A.2	A.1

ROP?

Read Output Query
Import the logical levels of the outputs

Syntax: ROP?
Parameters: None
Effect: Imports the logical level of the AP75 outputs from slots A and B. 0=0V, 1=24V
Response: q1: Integer value with the following contents

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Output	B.8	B.7	B.6	B.5	B.4	B.3	B.2	B.1	A.8	A.7	A.6	A.5	A.4	A.3	A.2	A.1

Related commands: **FLO**, **FLW**, **FCL**, **DBA**, **DBE?**, **SAO**, **OSP**, **SOP**, **SPF?**, **TDD**, **TDD?**

TDD

Transmit Device Data
Save/import settings, comment

Syntax: TDDp1,p2

Parameters:

p1	p2	Amplifier settings
0	any	Load factory settings
1	any	Import parameter set
2	any	Save parameter set
5	1..10	Import comment
6	1..10	Save comment

Remarks: Although the ML78 is a multi-channel amplifier, all TDD commands always apply to the whole module. Even if a TDD is addressed to a subchannel, the settings for all subchannels are saved or imported. The ML78 has only one parameter set.

Caution: If curve data is imported into the ML78 and command TDD2 is then executed, the modified curve data is stored in the Flash memory. Saving curve data (256 Kbytes) takes about 30 sec. During this time, analogue output updates run at reduced speed. Command TDD?10 can be used to query whether the data save has finished.

Related commands: **FLO**, **FLW**, **FCL**, **DBA**, **DBE?**, **SAO**, **OSP**, **SOP**, **SPF?**, **RIP?**, **ROP?**, **TDD?**

TDD? Transmit Device Data Query

Syntax: TDD?p1

Parameters:

p1	Amplifier settings
0	Origin (upper, left corner) of the amplifier data
1	Active parameter set
3	Test for presence of XM001
10	Test whether curve data has now been fully saved

Response:
q1 when p1=0: as for standard MGC
q1 when p1=1: as for standard MGC (when ML78 always 1)
q1 when p1=3: as for standard MGC (when ML78 always –1)
q1 when p1 = 10:

q1	Test whether curve data has been saved
0	Save complete
1	Data currently being saved

Related commands: **FLO**, **FLW**, **FCL**, **DBA**, **DBE?**, **SAO**, **OSP**, **SOP**, **SPF?**, **RIP?**, **ROP?**, **TDD**

I Global tables (parameter list)

See the below tables for the major parameter values. The MGCplus system CD (directory D/file Ids-kon.pdf) provides a complete list of parameters.

Status	Value	Remarks
On	1	
Off	0	
Yes	3	
No	2	

Tab.1: General

Status	Value	Remarks	Command
No supply	10		SAD
1V	11		
1.25V	12		
2.5V	13		
5V	14		
10V	15		
2–20mA	16	AP18	
0.2V	17	AP810/AP814	
0.5V	18	AP810/AP814	

Tab.2: Bridge excitation voltage

Status	Value	Remarks	Command
5V	21	ML60	SAD
100mV	22	ML60	

Tab.3: Input amplitude

Status	Value	Remarks	Command
Three–wire connection	25	ML35	SAD
Four–wire connection	26	ML35	

Tab.4: Input circuit

Status	Value	Remarks	Command
Short	31	AP08	SAD
Medium	32	AP08	
Long	33	AP08	

Tab.5: Decay time

Status	Value	Remarks	Command
Zero signal	40		AIS
Calibration signal	41		
Measuring signal	42		
Benchmark measuring point	43	ML01, AP09	

Tab.6: Input signal

Status	Value	Remarks	Command
No function	50		RIF
Autocal	51		
Tare	52		
Current/peak value 1	53		
Hold peak value 1	54		
Current/peak value 2	55		
Hold peak value 2	56		
Zeroing the amplifier	57		
Shunt	58		
Print	59		
Calibration signal	60		
Zero signal	61		
Operating sign change	62		
Load parameter set 1	63		
Load parameter set 2	64		
Load parameter set 4	65		
Remote/Local	66		
Start/Stop Integration	67		

Tab.7: Remote function

Status	Value	Remarks	Command
1	110		ICS
2	111		
5	112		
10	113		
20	114		
50	115		
100	116		
200	117		
500	118		
1000	119		

Tab.8: Display step

Status	Value	Remarks	Command
Above limit	130		LVS
Below limit	131		

Tab.9: Operating direction

Status	Value	Remarks	Command
Positive logic	135		LVS
Negative logic	136		

Tab.10: Output logic

Status	Value	Remarks	Command
No filter	140	–	SFC
Butterworth	141		
Bessel	142		
Aperiodic	143	–	

Tab.11: Filter characteristics

Status	Value	Remarks	Command
No combination	150	–	SLC, TRD
AND	151		
OR	152		
EXOR	153		
NAND	154		
NOR	155		
EXNOR	156		

Tab.12: Combination type

Status	Value	Remarks	Command
Minimum gross	180		PSM, PCM
Minimum net	181		
Maximum gross	182		
Maximum net	183		
Peak–to–peak gross	184	–	
Peak–to–peak net	185	–	
Comb.: Peak/Peak	186		
Comb.: Mean value	187		
Integration: gross	188	mean–value calculation	
Integration: net	189	mean–value calculation	

Tab.13: Mode peak store

Status	Value	Remarks	Command
No signal	200		LVS, SIS, PRF, LPR, MSS
Signal on display	201		
Gross, display filtered	202		
Net, display filtered	203		
Peak value 1	204		
Peak value 2	205		
Limit1 closing level	206		
Limit1 opening level	207	–	
Limit2 closing level	208		
Limit2 opening level	209	–	
Limit3 closing level	210		
Limit3 opening level	211	–	
Limit4 closing level	212		
Limit4 opening level	213	–	
Gross, dynamic	214		
Net, dynamic	215		
Limit value stati	216	On/Off	
Comb. Peak value	217		
User def. signal	219		

Tab.14: Signal type

Status	Value	Remarks	Command
Does not go into comb. on	300		SLC
Limit value status 1	301		
Limit value status 1 inverse	302		
Limit value status 2	303		
Limit value status 2 inverse	304		
Limit value status 3	305		
Limit value status 3 inverse	306		
Limit value status 4	307		
Limit value status 4 inverse	308		
Input 1	309		
Input 1 inverse	310		
Input 2	311		
Input 2 inverse	312		
Input 3	313		
Input 3 inverse	314		
Input 4	315		
Input 4 inverse	316		

Tab.15: Logical input signal

Status	Value	Remarks	Command
Full bridge	350		SAD
Half bridge	351		
Quarter bridge	352		
Strain gauge full bridge	353	ML30/38/55	
Strain gauge half bridge	354	ML55	
Strain gauge quarter bridge	355		
Inductive full bridge	356	ML50/51/55	
Inductive half bridge	357	ML50/5155	
Full bridge low level	358	ML10	
Half bridge low level	359	ML10	
Full bridge high level	360	ML10	
Half bridge high level	361	ML10	
Strain gauge full bridge 120 ohm	362	AP14	
Strain gauge full bridge 350 ohm	363	AP14	
Strain gauge full bridge 700 ohm	364	AP14	
Strain gauge half bridge 120 ohm	365	AP14	
Strain gauge half bridge 350 ohm	366	AP14	
Strain gauge half bridge 700 ohm	367	AP14	
Quarter bridge 120 ohm 4L	368	AP14	
Quarter bridge 350 ohm 4L	369	AP14	
Quarter bridge 700 ohm 4L	370	AP14	
Quarter bridge 120 ohm 3L	371	AP14	
Quarter bridge 350 ohm 3L	372	AP14	
Quarter bridge 700 ohm 3L	373	AP14	
DC V	420		SAD
DC A	421		
DC 75mV	425	ML01	
DC 10V	426	ML01	
DC 20mA	427	ML01	

Tab.16: Transducer type

Continued: Tab. 16 transducer types

Status	Value	Remarks	Command
Type J	450	ML01	
Type K	451	ML01	
Type T	452	ML01	
Type S	453	ML01	
Type B	454	ML01, ML01B and ML801 with AP09 and AP08	
Type E	455	ML01, ML01B and ML801 with AP09 and AP08	
Type R	456	ML01, ML01B and ML801 with AP09 and AP08	
Resistance	475		
500 ohm	476	ML35	
5000 ohm	477	ML35	
Pt10	500	ML35	
Pt100	501	ML35	
Pt500	502		
Pt1000	503	ML35	
Internal temperature	510		
Frequency	520		
Period	521		
Pulse counting	525	ML60	
2 kHz	530	ML60	
20 kHz	531	ML60	
200 kHz	532	ML60	
2MHz	537	ML60	
Deltatron^{TM} 0.1V	550	AP18	
Deltatron^{TM} 1V	551	AP18	
Deltatron^{TM} 10V	552	AP18	
Charge	570		
0.1nC	571	AP08	
1nC	572	AP08	
10nC	573	AP08	
100nC	574	AP08	

Status	Value	Remarks	Command
0.010 Hz	901		SFC
0.012	902		
0.015	903		
0.020	904		
0.025	905		
0.030	906	Bessel, ML38	
0.040	907		
0.050	908	Bessel, ML01, ML10, ML30, ML35, ML38, ML50/51/55, ML60	
0.060	909		
0.070	910		
0.075	911		
0.080	912		
0.090	913		
0.100	914	Bessel, ML01, ML10, ML30, ML35, ML38, ML50/51/55, ML60	
0.120	915		
0.150	916		
0.200	917	Bessel, ML01, ML10, ML30, ML35, ML38, ML50/51/55, ML60	
0.250	918		
0.300	919		
0.400	920		
0.500	921	Bessel, ML01, ML10, ML30, ML35, ML38, ML50/51/55, ML60	
0.600	922		
0.700	923		
0.750	924		
0.800	925		
0.900	926	Bessel ML38	
1.000	927	Butterworth ML38	
1.200	928		
1.250	929	Bessel, ML01, ML10, ML30, ML35, ML50/51/55, ML60	
1.500	930	Bessel, Butterworth ML38	
2.000	931		
2.500	932	Bessel, ML01, ML10, ML30, ML35, ML50/51/55, ML60, Butterworth ML38	
3.000	933	Butterworth ML38	

Tab.17: Low–pass filter frequencies

Continued: Tab. 17 Low–pass filter frequencies

Status	Value	Remarks	Command
4.000Hz	934		SFC
5.000	935	Bessel, ML01, ML10, ML30, ML35, ML50/51/55, ML60, Butterworth ML01, ML10, ML30, ML35, ML38, ML60	
6.000	936	Butterworth ML38	
7.000	937		
7.500	938		
8.000	939		
9.000	940	Butterworth ML38	
10.00	941	Bessel, Butterworth, ML01, ML10, ML30, ML35, ML38, ML50/51/55, ML60	
12.00	942		
15.00	943		
17.50	944		
20.00	945	Bessel, Butterworth, ML01, ML10, ML30, ML35, ML50/51/55, ML60	
25.00	946		
30.00	947		
40.00	948	Bessel, Butterworth, ML01, ML10, ML30, ML50/51/55, ML60	
50.00	949		
60.00	950		
70.00	951		
75.00	952		
80.00	953	Butterworth, ML01, ML10, ML30, ML50/51/55, ML60	
90.00	954		
100.0	955	Bessel, ML01, ML10, ML30, ML50/51/55, ML60	
120.0	956		
150.0	957		
200.0	958	Bessel, ML01, ML10, ML50/51/55, ML60, Butterworth ML30	
250.0	959	Butterworth, ML01, ML10, ML50/51/55, ML60	
300.0	960		

Continued: Tab. 17 Low–pass filter frequencies

Status	Value	Remarks	Command
400.0Hz	961	Bessel, ML01, ML10, ML50/51/55, ML60	SFC
500.0	962	Butterworth, ML01, ML10, ML50/51/55, ML60	
600.0	964		
700.0	965		
750.0	966		
800.0	967		
900.0	968	Bessel, ML50/51/55/60	
1000.	969	Butterworth, ML01, ML10, ML50/51/55, ML60, Bessel ML10	
1100.	970	Bessel ML01	
1200.	971		
1500.	972	Butterworth, ML50/51/55	
2000.	973	Butterworth, ML01, ML10	
2400.	974	Butterworth, ML01	
2500.	975		
3000.	976		
4000.	977		
5000.	978		
6000.	979		
7000.	980		
7500.	981		
8000.	982		
9000.	983		
10,000	1100	Bessel, ML10	
50,000	1140	Bessel, ML10	
100,000	1150	Bessel, ML10 (special version with 100kHz)	
–	1199	without filter	

Status	Value	Remarks	Command
HP Off	1200		HPS
HP 0.1 Hz	1201		
HP 1.0 Hz	1202		
HP 10 Hz	1203		

Tab.18: High pass filter frequencies

Status	Value	Remarks	Command
ASCII measured value, channel, status	1250		MVF, MBF
ASCII measured value	1251		
4 bytes binary MSB... LSB	1252		
4 bytes binary LSB..MSB	1253		
2 bytes binary MSB LSB	1254		
2 bytes binary LSB MSB	1255		
4 bytes float MSB LSB	1256		
4 bytes float LSB MSB	1257		

Tab.19: Measurement output format

Status	Value	Remarks	Command
current interface	1300		SBR, SPP, ADS
RS–232	1301		
RS–485	1302		
IEEE488	1303		
IEEE1284	1304		
Ethernet	1305		
Printer	1306		

Tab.20: Interface

Status	Value	Remarks	Command
No parity	1350		SBR
Even parity	1351		
Odd parity	1352		

Tab.21: Parity

Status	Value	Remarks	Command
75 Bd	1400		SBR
110 Bd	1401		
150 Bd	1402		
300 Bd	1403		
600 Bd	1404		
1200 Bd	1405		
2400 Bd	1406	CP12, CP32	
4800 Bd	1407		
9600 Bd	1408		
10000 Bd	1409		
19200 Bd	1410	CP12, CP32	
20000 Bd	1411		
38400 Bd	1412	CP32	
50000 Bd	1413		
57600 Bd	1414	CP32	
76800 Bd	1415		
115200 Bd	1416		
125000 Bd	1417		
153600 Bd	1418		
250000 Bd	1419		
307200 Bd	1420		
500000 Bd	1421		
614400 Bd	1422		
800000 Bd	1423		
1000000 Bd	1424		
1288000 Bd	1425		
2457600 Bd	1426		

Tab.22: Baud rate

Status	Value	Remarks	Command
ML30	5000		AMT?
ML50	5001		
ML01	5002		
ML55	5003		
ML60	5004		
ML35	5005		
ML10	5006		
ML51	5007		
ML55S6	5008		
ML38	5009		
ML801	5015		

Tab.23: Amplifier type

Status	Value	Remarks	Command
AP01, AP03, AP04, AP07, AP11, AP12, AP13	5500	Standard connection boards	CBT?
AP05, AP06	5501	Connection board with Zener barrier	
AP08	5502	Connection board for charge amplifier	
AP14	5503	Connection board with quarter bridge extension	
AP18	5504	Connection board for piezoelectric transducer	
AP09	5505	Connection board with reference point	
AP801	5506	for ML801	
AP809	5507	for ML801	
AP835	5508	for ML801	
AP409	5509	for ML801	
AP4092	5510	for ML801 (2x mounted AP409)	
AP810	5511	for ML801	
AP814	5512	for ML801	

Tab.24: Connection board type

Status	Value	Remarks	Command
Measured value ≥ Trigger level	6000	–	TRE
Measured value ≤ Trigger level	6001	–	
Positive edge	6002		
Negative edge	6003		
Static positive (≥ or within)	6004		
Static negative (≥ or outside)	6005		
Threshold	6006		
Window	6007		
Limit value1	6008		
Limit value2	6009		
Limit value3	6010		
Limit value4	6011		

Tab.25: Trigger mode

Status	Value	Remarks	Command
immediately	6100		TRD, TRE
TRG command	6101	–	
External trigger	6102		
Internal trigger	6103	–	

Tab.26: Trigger start mode

Status	Value	Remarks	Command
1 Hz	6300		ICR
2	6301		
5	6302		
10	6303		
25	6304		
50	6305		
60	6306		
75	6307		
100	6308		
150	6309		
200	6310		
300	6311		
400	6312		
600	6313		
800	6314		
1200	6315		
1600	6316	–	
2400	6317	Default	
3200	6318	–	
4800	6319		
9600	6320		

Tab.27: Sampling rates

Continued: Tab. 27 sampling rates

3 Hz	6321		ICR
4	6322		
6	6323		
8	6324		
15	6325		
20	6326		
30	6327		
48	6328		
64	6329	–	
80	6330		
96	6331	–	
120	6332		
160	6333		
192	6334	–	
240	6335		
320	6336	–	
384	6337	–	
480	6338		
640	6339	–	
960	6340	–	
1920	6341	–	
2s	6350		
5	6351		
10	6352		
20	6353		
50	6354		
100	6355		
200	6356		
500	6357		
1000	6358		
2000	6359		
5000	6360		
10000	6361		

Status	Value	Remarks	Command
Start measured value memory	6400		OMP, RMB?
Start measurement run	6401		
Start next measurement run	6402		
Start next measurement run, release old one	6403		
Trigger point	6404		
End of measurement run	6405		
Current position	6406		
Current position, release before	6407		
Current recording position	6408		
Current position, release inclusive	6409		

Tab.28: Reader pointer positioning in recorded measurement

Status	Value	Remarks	Command
Start trigger	6500		TRD
Stop trigger	6501		
Sampling rate trigger	6502		

Tab.29: Trigger type

Status	Value	Remarks	Command
No error	0		EST?
FACTORY CAL ERR 1	15001		
FACTORY CAL ERR 2	15002		
FACTORY CAL ERR 3	15003		
FACTORY CAL ERR 4	15004		
FACTORY CAL ERR 5	15005		
FACTORY CAL ERR 6	15006		
FACTORY CAL ERR 7	15007		
AP WRONG	15010	Connection board not suitable for amplifier; Remedy:load factory setup	
APXX ERR	15011	Parameters not suitable for AP; Remedy:load factory setup	
PARAM NOT VALID	15012	Parameters not consistent; Remedy:load factory setup	
TRANSDUCER ERR	15015	Error in six wire circuit	
CALIBRATION ERR	15020	Calibration lines have not produced a valid setting	
Hardware underflow	15030	Error in six wire circuit	
Hardware overflow	15031	Error in six wire circuit	
Gross overflow	15040	Error in six wire circuit	
Net overflow	15041	Error in six wire circuit	
SYNC ERR	15050	Excitation voltage not synchronous, fast measurement acquisition not possible; Remedy: synchronise device to MASTER or devices with cables	
Additional errors	15100	–	
PARAM.CHANGE	20000	Table setting has changed	
AO1 OVERFL	20020	Analogue output 1 overflow	
AO2 OVERFL	20021	Analogue output 2 overflow	

Tab.30: Error query

J Assignment command – hardware

Command	CP12	CP32B	MC	MLxx	MLxxB	ML801	ML71	ML77	ML78
ACL			x	x		x	x[1)]	x[1)]	x[1)]
ADR?	x	x							
ADS	x	x							
AFS			x			x			
AID?			x			x	x	x	x
AIS			x			x			
AMT?			x			x	x	x	x
ASA			x						
ASF			x						
ASS			x			x			
BDR	x	x				x			
CAL			x	x		x	x[1)]	x[1)]	x[1)]
CAN				x		x	x[1)]	x[1)]	x[1)]
CAV					x				
CAP				x		x			
CBT?				x		x	x	x	x
CDT				x		x			
CDV				x		x			
CDW			x						
CFS		x							
CGP		x							
CHS	x	x							
*CLS	x	x							

1) Command will be accepted, however, it will not have any effect

Command	CP12	CP32	MC	MLxx	MLxxB	ML801	ML71	ML77	ML78
CMF		x							
CMR			x			x			
COF	x	x							
CPI?							x		
CPO							x		
CPV			x	x		x	x[1)]	x[1)]	x[1)]
DBA							DBA		DBA
DBE?							DBE?		DBE?
DBS?							x		
DCL	x	x							
DEL		x							
DFL					x	x			
DGM			x			x			
DID?	x	x							
DIR		x							
DSD	x	x					x	x	x
EES?		x							
ENU			x			x			
*ESE	x	x							
ESR?	x	x							
EST?					x			x	x
EUN				x		x	x		

Command	CP12	CP32B	MC	MLxx	MLxxB	ML801	ML71	ML77	ML78
FCL							FCL		FCL
FDA									
FDE									
FDH									
FDI?									
FDR?									
FDS									
FDW									
FLO							x		FLO
FLW							x		FLW
GFV			x			x[1)]			
HCF?		x							
HPS				x	x	x[2)]			
HPV				x		x[2)]			
IAD			x						
IBY?	x								
ICR		x							
ICS				x		x	x		
*IDN?	x	x							

[1)] AP814 only [2)] Command will be accepted, however, it will not have any effect

Command	CP12	CP32B	MC	MLxx	MLxxB	ML801	ML71	ML77	ML78
IDS?				x		x		x	x
IHD		x							
IMR			x						
IPA		x							
ISR			x			x			
ISS	x	x							
IST?	x	x							
ISV?				x		x	x		
IUN				x		x	x		
LIV			x						
LOR			x	x		x[2)]	x[2)]	x[2)]	x[2)]
LPR		x							
LSS?					x				
LVD				x		x			
LVL				x		x			
LVS				x		x	x[2)]	x[2)]	
MBF		x							
MCS		x							
MDD	x	x							
MPV				x		x[2)]			
MRG		x							
MSS		x							
MSV?	x	x					x	x[2)]	x[2)]

2) Command will be accepted, however, it will not have any effect

Command	CP12	CP32B	MC	MLxx	MLxxB	ML801	ML71	ML77	ML78	ML85
MVA?				x		x				
MVC				x		x[2)]				
MVF		x					x		x	
NAS?							x			
OCP				x			x			
OMP		x								
OPS			x			x[2)]				
OSP				x		x[2)]	**OSP**		**OSP**	
PCM				x		x[2)]				
PCS		x								
PFS			x	x		x				
PHD		x								
PPM	x	x								
***PRE**	x	x								
PRF				x		x[1)]	x			
PRQ										x
PRT	x	x								
PSC										
PSM				x		x				
PVS			x			x				

1) not PkPk
2) Command will be accepted, however, it will not have any effect

Command	CP12	CP32B	MC	MLxx	MLxxB	ML801	ML71	ML77	ML78
RES	x	x				x			
REV?							x		
RFP			x						
RHD?		x							
RIF				x		x[2]			
RIP?				x		x[2]			RIP?
RMB?		x							
RMS?		x							
RMV?		x							
ROP?									x
*RST	x	x							
SAD				x		x			
SAF				x					
SAO					SAO		SAO		SAO
SBR		x							
SCL			x	x		x[1]			
SCM		x							
SFC				x		x			
SHD		x							
SIS		x							

[1] AP810 and AP814 only

[2] Command will be accepted, however, it will not have any effect

Com-mand	CP12	CP32B	MC	MLxx	MLxxB	ML801	ML71	ML77	ML78
SLC				x		x[2)]			
SML		x							
SMS		x							
SNO?		x							
SOP				x		x[2)]			SOP
SPF							x		SPF?
SPP	x	x							
SPS		x							
SRB	x	x							
*SRE	x	x							
SRP		x							
STB?	x	x							
STD		x							
STL			x	x		x[2)]	x		
STP	x	x							
Sxx	x	x							
TAR			x						
TAT				x		x			
TAV				x		x	x[2)]	x[2)]	x[2)]
TCD		x							
TDD			x	x		x	x	x	TDD

2) Command will be accepted, however, it will not have any effect

Command	CP12	CP32B	MC	MLxx	MLxxB	ML801	ML71	ML77	ML78
TEX	x	x							
TRD		x							
TRE		x							
TRG		x							
TRR		x							
TSV		x							
UCC			x	x		x	x	x	x

K Alphabetical command list

G

H

I

L

M

N

O

P

L Command list by function

Abbreviation		Page
• **Identification**		**H–38**
AID?	Output the amplifier identification	H–145, H–38
AMT?	Output the amplifier type	H–38
CBT?	Output the connection board type	H–40
DID?	Output the amplifier identification	H–41
EES?	Queries internal error status	H–43
***IDN?**	Output the device identification	H–40
IDS?	Identification string for the parameter value Read out	H–39, H–253
SML	Select message language	H–42
SML?	Query the language setting	H–42
SNO?	Read out serial number of CP32B	H–44
• **Time and date**		**H–45**
STD	Set up the system time and date	H–45
STD?	Output the system time and date	H–45
CGP	Set GPIB Write timeout	H–46
CGP?	Establish GPIB Write timeout	H–46

Amplifier set–ups

Abbreviation		Page
• **Amplifier input**		**H–47**
AIS	Select the amplifier input signal	H–47
AIS?	Output the amplifier input signal	H–47
ASA	Input the bridge excitation voltage and transducer type	H–202
ASA?	Output the bridge excitation voltage and transducer type	H–205
ASS	Select the amplifier input signal	H–209
ASS?	Output the input signal type	H–210
HPS	Select the high pass cut–off frequency	H–48
HPS?	Output the high pass cut–off frequency	H–48
SAD	Select the input characteristic adjustment	H–49
SAD?	Output the input characteristic adjustment	H–49
SAF	Transducer frequency adaptation	H–50
DSD	Input of amplifier settings	H–146, H–52
DSD?	Output of amplifier settings	H–52, H–149
• **Filter setting**		**H–211**
AFS	Filter changeover (fc 1/2)	H–211
AFS?	Output the filter setting	H–211
ASF	Input the cut–off frequency and filter characteristics	H–212
ASF?	Output the cut–off frequency and filter characteristics	H–213
SFC	Input the cut–off frequency and filter characteristics	H–53
SFC?	Output the cut–off frequency and filter characteristics	H–53
SPS	Subchannel selection mask	H–56
SPS?	Subchannel selection mask output	H–56

B 31.MGCPR.40 en

Hottinger Baldwin Messtechnik GmbH

Postfach 10 01 51, D-64201 Darmstadt
Im Tiefen See 45, D-64293 Darmstadt
Tel.: +49/61 51/8 03-0; Fax: +49/61 51/ 8039100
E–mail: support@hbm.com www.hbm.com