DIGIFORCE® 9310

Interface Commands
RS232 / 485 / Ethernet

valid as of software version: **V2006.01**
Status: 28. February 2007

Manufacturer:
burster präzisionsmeßtechnik gmbh & co kg
Talstraße 1-5 P.O. Box 1432
76593 Gernsbach 76587 Gernsbach
☎+49(0)7224-645-0

Note:
The following information may be amended without prior notice. No part of this document may be reproduced or processed using electronic system without prior consent in writing.

burster provides no warranty of any kind respect to this material, including the implied warranty of merchantable quality and fitness for purpose.
burster is not liable under any circumstances for errors, incidental damage or consequential loss sustained in connection with the function or use of this material.

DIGIFORCE® 9310

External Interface Commands
RS232 / 485 / Ethernet

Table of Content

1 Revision History

Status	description
28.02.2007 / KS	- EthernetUDP protocol included
18.08.2006 / KS	- envelope commands inserted - chapter „Seral communication“ inserted - chapter „ Testmode“ inserted - new commands to manage the ethernet interface inserted

2 Serial communication

The PC-Software Digicontrol 9310 for device configuration, backup and data recording is available (Ordering code 9310-P10x). The software is available in two versions. The P101 version is for device configuration and backup. The P100 version supports an additional automatized data recording.

2.1 Serial communication during measurement access

During measurement of the DIGIFORCE 9310 there is no communication possible. The device does not answer to a commad.

2.2 Interface parameter of the serial RS232/485 interface

The settings have to be defines in the general settings menue

Baudrate:	300, 600, 1200, 2400, 4800, 9600(*), 19200, 38400, 56000, 57600
Data bits:	7 or 8(*)
Stop bits:	1(*) or 2
Parity:	no(*), even, odd
Blockcheck:	OFF(*), ON

no hardware handshake

(*) → default setting after initialisation

2.3 Communication protocol

Kontrollzeichen:	<STX>	0x02	=> start of Text
	<ETX>	0x03	=> end of Text
	<ENQ>	0x05	=> anquiry
	<ACK>	0x06	=> acknowlege
	<S>	0x20	=> space
	<NAK>	0x15	=> not Acknowlege
	<LF>	0x0A	=> line feed
	<EOT>	0x04	=> end of transmission
	<NUL>	0x00	=> NUL sign

ANSII standard X3.28-1976 Subcategory 2.5, A4 is used as the communications protocol. This standard is used in systems in which a number of secondary stations exist in a non-switched multipoint connection, and all commands are sent by a control station. Only one transmitter (master) and one receiver (slave) are ever active on the bus at one time. One station is the control station. The control station is given master status and sends commands to a selected slave station, or relinquishes its master status to a secondary station and assumes slave status to receive data. A connection between two secondary stations is not allowed. The control station monitors the connection continuously.

2.3.1 Establishing a connection

Before a connection is established, the control station has master status and none of the secondary stations have slave status. The connection can be established in two different ways:

(1) „Selection with response“
In this case, device addressing and command sending do not take place in the same communications step. This method is useful when you want to send several commands to the same device and then retrieve the responses to these commands in one go. (See communication example at the end of this chapter.)

or

(2) „Fast Selection“
In this case addressing is combined with the command. This saves a communications step if you want to exchange data with several devices (via RS485) (see communications example at the end of this chapter)

When establishing a connection, the control station can either:

(1) specify a slave station
in order to set up a connection i.e. send a command to the addressed slave

or

(2) carry out polling,
to grant master status to a secondary station, i.e. request a response to a previously sent command and hence assign the transmit right to the slave.

2.3.2 Selection with response

The control station sends a "selection supervisory sequence". The selection supervisory sequence is used to initialize the 9310 as a slave so that it is subsequently possible to send commands to it. The prefix calls up a single secondary station. **<ENQ>** defines the end of the selection supervisory sequence.

The selection supervisory sequence of the 9310 has the following format.

<Address>sr<ENQ>

- **<Address>** device address (decimally 0..99)
- **sr** ASCII-string "s" und "r"
- **<ENQ>** ASCII code ENQ

A secondary station that recognizes its selection supervisory sequence assumes slave status and sends one of two responses:

(1) If the station is ready to receive data,
it sends **<ACK>**. On receiving this response,
the master station initiates data transfer.

(2) If the station is not ready to receive data,
it sends **<NAK>**. On receiving this response,
the master station can attempt call up the same station again.

If the master station receives an invalid response or none at all, it can attempt to address the same station again or end the transmission.

2.3.3 Fast Selection

Instead of "selection with response", the master station can send a selection supervisory sequence without **<ENQ>**. This will designate a secondary station as the slave station. It then initiates data transfer directly without waiting for the acknowledge response from the secondary station.

The fast selection supervisory sequence of the 9310 has the following format.

<Address>sr<STX>command<ETX><BCC>

- **<Address>** device address (decimally, 0..99)
- **sr** ASCII-string "s" und "r"
- **<STX>** ASCII code STX
- **Befehl** command sequence
- **<ETX>** ASCII code ETX
- **<BCC>** **optionally** Blockcheck (see Chapter 2.3.5 Datenübertragung)

2.3.4 Polling

The control station sends a "polling supervisory sequence". The polling supervisory sequence is used to retrieve requested data from the 9310. The prefix selects a single station. **<ENQ>** defines the end of the “Polling supervisory sequence”:

The Polling Supervisory Sequence of DIGIFORCE 9310 has the following structure:

<Address>po<ENQ>

- **<Address>** hexadezimal device address (0..99)
- **po** ASCII-string "p" und "o"
- **<ENQ>** ASCII code ENQ

A secondary station that recognizes its polling supervisory sequence responds using one of two options:

(1) If the station has data ready to send, it starts the data transfer. The control station assumes slave status.

(2) If the station has no data ready to send, it sends **<EOT>**, which terminates its master status. The master status returns to the control station.

If the control station receives an invalid response or none at all, it terminates the connection by sending **<EOT>**.

2.3.5 Data transfer

After establishing the connection, data is transferred in accordance with the rules of subcategory A4. The master station begins the transfer with **<STX>**. Then the corresponding data is sent. The data block is terminated with **<ETX>**. The **<ETX>** character is followed by the optional block check character **<BCC>**. This is formed from all the bytes that come after **<STX>**, **including <ETX>**. The **<BCC>** is generated by performing an exclusive-OR operation on all these bytes. 80hex is also XOR'ed with the result of this operation in order to prevent any possible mix up with control characters.

The slave station sends one of two possible responses after detecting the **<BCC>**:

- If the data has been accepted and the station is ready to receive new data, it sends **<ACK>.** On receiving this, the master station either sends new data or terminates the data transfer.

- If the data was not accepted and the slave station is ready to receive new data, it sends **<NAK>**. On receiving this, the master station may send other data or terminate the connection.

2.3.6 Terminating a connection

The master station sends **<EOT>** to indicate that it has no more data to transfer. **<EOT>** returns the master status to the control station.

2.3.7 Command structure

2.3.7.1 Commands without parameters

aaaaB<LF>

aaaa	commad name, 4 ASCII-characters
B	command type, ´?´ read out, ´!´ set
<LF>	Line Feed, 0x0A

2.3.7.2 Commands with parameters

aaaaB<S>P1,P2,...,Px<LF>

aaaa	commad name, 4 ASCII-characters
B	command type, ´?´ read out, ´!´ set
<S>	space, 0x20
P1,P2,...,Px	Parameter 1.....x, separated by comma
<LF>	Line Feed, 0x0A

2.3.7.3 Strukture of an answer on a aaaa? command

Command **aaaa?<LF>**

aaaa	commad name, 4 ASCII-characters
B	command type, ´?´ read out, ´!´ set
<LF>	Line Feed, 0x0A

DIGIFORCE 9310 answers (example with three parameters):

P1<NUL>,P2<NUL>,P3<NUL><LF>

Px	Parameter x
<NUL>	NUL sign, ASCII code 0x00
<LF>	Line Feed, 0x0A

Note:
One exception forms the KURV? Command, with no <NUL>-sign within the answer.

2.3.8 Examples of the communication sequence

The following sequence illustrates the DIGIFORCE 9310 communicating with a host controller in the two communications modes "selection with response" and "fast selection". In the example, the INFO query command is used, the 9310 has address 00, and block check is disabled (in one example the block check is also shown for the given command / the given data).

2.3.8.1 Communication “Selection with response” (sample)

Host sends: <EOT>

to make sure that any open connections are terminated and the 9310 receive buffer is cleared.

Host sends: 00sr<ENQ>
Selection: DIGIFORCE 9310 with address 0 is addressed

9310 answers: <ACK>
Acknowledge from 9310

Host sends (Blockcheck OFF): <STX>info?<LF><ETX>
Host sends (Blockcheck ON): <STX>info?<LF><ETX><BCC> (here BCC = B8hex)

9310 answers: <ACK>
Acknowledge from 9310

Host sends: <EOT>
The host controller unaddresses the device in order to start a polling sequence immediately.

Host sends: 00po<ENQ>
The 9310 with address 0 is requested to send all existing responses

If block check is OFF, 9310 answers:
<STX>V200101<NUL>,SN123456<NUL>,09.03.2001<NUL><LF><ETX>

If block check is ON, 9310 answers:
<STX>V200101<NUL>,SN123456<NUL>,09.03.2001<NUL><LF><ETX><BCC> (here BCC = CEhex)

This is the correct response to the info? command

Host sends: <ACK>
The controller has received the response and accepted it. Does the 9310 have other queries saved for which a response can now be sent?

9310 answers: <EOT>
No. This ends the communication sequence and the 9310 has unaddressed itself.

2.3.8.2 Communication with “Fast Selection” (sample)

Host sends: <EOT>
to make sure that any existing connections are terminated and the 9310 receive buffer is cleared.

Host sends: 00sr<STX>info?<LF><ETX>
Command sequence: The 9310 with address 0 is to be addressed and the info? command is to be executed

9310 answers: <ACK>
The 9310 signals that it accepts the addressing and recognizes and has understood the info? command

Host sends: <EOT>
The host controller unaddresses the device in order to start a polling sequence immediately.

Host sends: 00po<ENQ>
The 9310 with address 0 is requested to send all existing responses

9310 answers: <STX>V200101<NUL>,SN123456<NUL>,09.03.2001<NUL><LF><ETX>
This is the correct response to the info? command

Host sends: <ACK>
The controller has received the response and accepted it. Does the 9310 have other queries saved for which a response can now be sent?

9310 replies with: <EOT>
No. This ends the communication sequence and the 9310 has unaddressed itself.

2.3.9 Qbasic sample

Read out of the DIGIFROCE 9310 Info-String (Compiler Version: Quick-Basic 4.5)

2.3.9.1 Communication with „Selection with response“

```
REM ************************************************************************
REM **                                                                    **
REM **        9310_1.bas                  Developped by:MN,Li             **
REM **        Communication               Prog. language: Qbasic 1.1      **
REM **                    exe-File created with QB 4.5                    **
REM **        with selection with                                         **
REM **        response                    date:  13.03.2000, 05.05.2003   **
REM **        example: ask for ID-string                                  **
REM **                                                                    **
REM ************************************************************************

REM (1) Definition of ASCII-Control Characters

REM STX Start of text:  0x02
STX$ = CHR$(2)

REM ETX End of text:  0x03
ETX$ = CHR$(3)

REM EOT End of transmission:  0x04
EOT$ = CHR$(4)

REM ENQ Enquiry:  0x05
ENQ$ = CHR$(5)

REM ACK Acknowledge:  0x06
ACK$ = CHR$(6)

REM LF line feed:  0x0a
LF$ = CHR$(10)

REM CR carriage return:  0x0d
CRE$ = CHR$(13)

REM NAK not acknowledge:  0x15
NAK$ = CHR$(21)

REM+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
REM Dialog:  Selection and opening/initialisation of PC-Interface
REM+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

CLS
INPUT "Which interface do you want to use? (1 -> COM1, 2 -> COM2)"; a
 IF ((a <> 1) AND (a <> 2)) THEN PRINT "illegal Interface": END
 IF (a = 1) THEN com$ = "COM1"
 IF (a = 2) THEN com$ = "COM2"
openstr$ = com$ + ":9600,N,8,1"
PRINT

REM ** rs232 initialisation
OPEN openstr$ FOR RANDOM AS #3


REM+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
REM Ask Device (adr 0) for ID-String with Mode "selection with response"
REM (one of the two communication modes)
REM+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

PRINT "-------->>>>> Connecting Device with adress 1...."

REM ** Sending "selection supervisory sequence" and pick up answer   send EOT first to end
other (probably unanswered) enquiries
PRINT #3, EOT$ + "00" + "sr" + ENQ$
REM clear answer string
ant$ = ""
REM read characters from serial interface
ant$ = INPUT$(1, #3)

REM new char should be an ACK
```

```
IF ant$ <> ACK$ THEN PRINT "Comunication error, not (ACK) received but:"; ant$
PRINT "selection supervisory string sent"
REM press 'enter' to proceed
INPUT "ENTER TO GO ON"; a$: a$ = ""

REM ** Sending command "INFO?" to 9310 (enclosed with STX and ETX)
PRINT #3, STX$ + "INFO?" + ETX$

REM clear answer string
ant$ = ""
REM read characters from serial interface
ant$ = INPUT$(1, #3)

REM new char should be an ACK
IF ant$ <> ACK$ THEN PRINT "Comunication error, not (ACK) received but:"; ant$

REM !!IMPORTANT!! de-adress before start polling
PRINT #3, EOT$

PRINT "ID-Enquiry sent"
REM press 'enter' to proceed
INPUT "ENTER TO GO ON"; a$: a$ = ""

REM 9310 wants to answer now and waits for polling

REM start polling
PRINT #3, "00" + "po" + ENQ$

REM clear answer string
ant$ = ""

REM initialize variable char$ to anything but ETX
char$ = STX$
REM read from serial interface until ETX and add to answer-string
WHILE (char$ <> ETX$)
        char$ = INPUT$(1, #3)
        ant$ = ant$ + char$
WEND

REM ID-string received, send ACK
PRINT #3, ACK$

REM Printing "Dev 0 INFO:" on PC-sreen:
PRINT "DEVICE 0 answers: ", ant$

REM Reading EOT from 9310
ant$ = ""
ant$ = INPUT$(1, #3)

REM new char should be an EOT
IF ant$ <> EOT$ THEN PRINT "Comunication error, not (EOT) received but:"; ant$

PRINT "Program has ended successfully"

END
```

2.3.9.2 Communication with „Fast Selection“

```
REM ************************************************************************
REM **                                                                    **
REM **        9310_2.bas               Developped by:MN,Li                **
REM **                         Prog. language: Qbasic 4.5                 **
REM **        Communication              exe-File created with QB 4.5     **
REM **        with fast selection        date:  13.03.2000                **
REM **        example: ask for ID-string with fast selection              **
Rem ************************************************************************

REM Definition of ASCII-Control Characters

REM STX Start of text:  0x02
STX$ = CHR$(2)

REM ETX End of text:  0x03
ETX$ = CHR$(3)

REM EOT End of transmission:  0x04
EOT$ = CHR$(4)

REM ENQ Enquiry:  0x05
ENQ$ = CHR$(5)

REM ACK Acknowledge:  0x06
ACK$ = CHR$(6)

REM LF line feed:  0x0a
LF$ = CHR$(10)

REM CR carriage return:  0x0d
CRE$ = CHR$(13)

REM NAK not acknowledge:  0x15
NAK$ = CHR$(21)

REM+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
REM Dialog:  Selection and opening/initialisation of PC-Interface
REM+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

CLS
INPUT "Which interface do you want to use? (1 -> COM1, 2 -> COM2)"; a
 IF ((a <> 1) AND (a <> 2)) THEN PRINT "illegal Interface": END
 IF (a = 1) THEN com$ = "COM1"
 IF (a = 2) THEN com$ = "COM2"
openstr$ = com$ + ":9600,N,8,1"
PRINT

REM ** rs232 initialisation
OPEN openstr$ FOR RANDOM AS #3

PRINT "Please set up the 9310 with:"
PRINT "          baudrate = 9600, Data bits = 8,"
PRINT "          Stopp bits = 1, No parity, no blockcheck"
PRINT "          adress 0"
PRINT

REM+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
REM Ask Device (adr 0) for ID-String with Mode "fast selection"
REM (one of the two communication modes)
REM All commands in the user manual are described in this mode
REM+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

PRINT "-------->>>>> Connecting Device with adress 0...."

REM send EOT first to end other (probably un-answered) enquiries (strongly recommended)
PRINT #3, EOT$

REM Create and send command
PRINT #3, "00" + "sr" + STX$ + "INFO?" + ETX$

REM clear answer string
ant$ = ""
REM read characters from serial interface
ant$ = INPUT$(1, #3)
```

```
REM new char should be an ACK
IF ant$ <> ACK$ THEN PRINT "Comunication error, not (ACK) received but:"; ant$

REM press 'enter' to proceed
INPUT "ENTER TO GO ON"; a$: a$ = ""

REM !!IMPORTANT!! de-adress before start polling
PRINT #3, EOT$

REM 9310 wants to answer now and waits for polling

REM start polling
PRINT #3, "00" + "po" + ENQ$

REM clear answer string
ant$ = ""

REM initialize variable char$ to anything but ETX
char$ = STX$
REM read from serial interface until ETX and add to answer-string
WHILE (char$ <> ETX$)
        char$ = INPUT$(1, #3)
        ant$ = ant$ + char$
WEND

REM ID-string received, send ACK
PRINT #3, ACK$

REM Printing "INFO" on PC-sreen:
PRINT "Device (0) answers: ", ant$

REM Reading EOT from 9310
ant$ = ""
ant$ = INPUT$(1, #3)

REM new char should be an EOT
IF ant$ <> EOT$ THEN PRINT "Comunication error, not (EOT) received but:"; ant$

PRINT "Program has ended successfully"

END
```

2.3.10 Construction and fragmentation of Ethernet UDP frames

With a data set of more than 7500 byte the data packet must become fragmented (divided) - see example. The DIGIFORCE 9310 supports coded (more burster internally) and unencrypted UDP frames, this can be configured in the equipment (default is: coded and not coded).

Telegram format from 9310 to host

Format with data **(answer to xxxx? command)**

<STX>Key, Identifier, Status, Number, Data <End sign> Block-Check

Format without data **(error or no xxxx? command)**

<STX>Key, Identifier, Status, Number,<End Sign>Block-Check

STX:	0x02
Key:	0: message is not coded 1: message is coded
Identifier:	continuous number 1..999 (ASCII), 9310 answers with the same number as received
Status:	0: OK 1: NAK 2: not used 3: Timeout on serial interface 4: STX not identified 5: Identifier not valid 6: ETX not identified 7: Checksum error 8: no answer 9: unknown error A: measurement active B: illegal Host-IP-address (static) C: Not coded message received D: Not guilty key number E: Device is blocked by another master (MAST! command)
Number:	0: No fragmentation or count of the first fragmentation >0: identity number of fragmentation
Data:	User data in ASCII format
End sign:	ETX (0x03) or ENQ (0x05) with fragmentation
Block-Check:	1 Byte Checksum, all Bytes after STX including ETX/ENQ are XOR-combined

No fragmentation necessary (data packet <=7500 Byte)

<STX>Key, Identifier, Status, 0, Data <ETX><Block-Check>

Number 0 means no fragmentation. End sign is ETX (no fragmentation).

With fragmentation (example: data packet 18000 Byte)

1st fragment
<STX>Key, Identifier, Status, **0**, Data 1…7500<**ENQ**><Block-Check>
2nd fragment
<STX>Key, Identifier, Status, **1**, Data 7501…15000<**ENQ**><Block-Check>
3rd fragment
<STX>Key, Identifier, Status, **2**, Data 1…7500<**ETX**><Block-Check>

UDP frame format from HOST to DIGIFORCE 9310

<STX>Key, Identifier, Data,<End Sign>Block-Check

STX:	0x02
Key:	0: message is not coded 1: message is coded
Identifier:	continuous number 1..999 (ASCII)
Data:	User data in ASCII format
End sign:	ETX (0x03) or ENQ (0x05) with fragmentation
Block-Check:	1 Byte Checksum, all Bytes after STX including ETX/ENQ are XOR-combined

Example:
Host sends INFO?-command to DIGIFORCE 9310

Host sends:
<STX>0,1,INFO?<ETX><179>

DIGIFORCE 9310 answers:
<2>0,1,0,0,V200606 ,298043 ,15.11.2006<3><242>

2.4 General instructions

2.4.1 Interface watchdog timer

2.4.1.1 Timer A (response timer)

Timer A is used by the DIGIFORCE 9310 to protect itself from an invalid response or no response.

- **Start**: Timer A is started after data transfer has been terminated with <ETX>. The instrument waits for an acknowledgement by the master.
- **Stop**: Timer A is stopped if a valid response <ACK> has been received.
- **Timeout**: If a timeout occurs, the DIGIFORCE 9310 sends an <EOT> and returns to its initial state (ready for a new command).

The timeout for Timer A is set to 5 seconds.

2.4.1.2 Timer B (receive timer)

Timer B is used by the receive station, to protect itself against non-recognition of the <ETX> character.

- **Start**: Timer B is started after receiving the <STX> character.
- **Restart**: Timer B is restarted as long as data is being received in order to allow variable data block lengths to be received.
- **Stop**: Timer B is stopped when the <ETX> character has been received.
- **Timeout**: If a timeout occurs, the received data (command) is discarded. The instrument enters the initial state and waits for new commands.

The timeout for Timer B is set to 5 seconds.

2.4.2 Information about the command descriptions

In the following command examples, only connection establishment with “fast selection” is described. "Selection with response” can obviously also be used.

It is recommended that before each command sequence an **<EOT>** is sent to the 9310. This clears the receive buffers so that no previously sent characters (e.g. sent during interface initialization) can cause an erroneous command interpretation.

Meaning of the symbol syntax used in the following examples:

<Address> => device address (00 .. 99)
<STX> 0x02 => Start of Text
<ETX> 0x03 => End of Text
<ENQ> 0x05 => Enquiry
<ACK> 0x06 => Acknowlege
<NAK> 0x15 => Not Acknowlege
<LF> 0x0A => Line Feed
<EOT> 0x04 => End Of Transmission
<NUL> 0x00 => NUL sign
<S> 0x20 => Space

⚠ Caution. *Important notes*:

- **Only the commands described in this operating manual should be used. Use of undocumented commands can cause incorrect unit operation.**
- **No commas can occur within a parameter.**
- **A point ‘.’ is used in floating-point numbers**
- **The number of parameters must always be adhered to**
- **Command transfer must take exclusively in upper or lower case**

3 Interface commands

3.1 Comman Comands

3.1.1 Default Calibration

3.1.1.1 LGRK **Load default calibration**

With the command LGRK! the default calibration from EEPROM is loaded into the 9310 buffered RAM.

Host sends:	<Address>sr<STX>LGRK!<ETX>[<BCC>]
DIGIFORCE answers:	<ACK>
Host sends:	<EOT>

3.1.2 Informationen for PC-software DIGICONTROL

3.1.2.1 DIGI **DIGICONTROL – specific device properties**

With the command DIGI? the host can read out specific information about the device software.

Host sends: <Address>sr<STX>DIGI?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE antswers: <STX>P1,P2,P3,P3,P5,P6,P7<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	**Meaning**	**Value**
P1	Version of the sensor hardware	Bit0 = 0 DMS (standard) Bit0 = 1 Piezo (hex, unsigned short)
P2	Version of the communikation hardware	0x0: standard (RS232/485) Bit0: 1 PROFIBUS available Bit1: 0 = RS485 Bit1: 1 = Ethernet Bit2: 0 = display device Bit2: 1 = Blackbox unsigned short
P3	Counter an software changes with influence on serial communication	On every change of the 9310 software with influence on serial communication this value is incremented. The value starts on version V200208 with 0 (decimally, unsigned short)
P4	Identifier of specific software version (first Byte)	Bit0: = 0 standard; = 1 specific software Bit1: PC software totally locked Bit2: *reserved* Bit3: *reserved* Bit4: Upload locked Bit5: Download locked Bit6: *reserved* Bit7: Laboratory operation mode locked (hex, unsigned short)
P5	Identifier of specific software version (second Byte)	Bit0: General settings locked Bit1: Access authorization locked Bit2: Program access locked Bit3: *reserved* Bit4: *reserved* Bit5: *reserved* Bit6: *reserved* Bit7: *reserved* (hex, unsigned short)
P6	Identifier of specific software version (third Byte)	Bit0: Channel settings locked Bit1: Measuring mode locked Bit2: Sendor test locked Bit3: Evaluation locked Bit4: Switch points locked Bit5: Darstellung gesperrt Bit6: Envelope locked Bit7: *reserved* (hex, unsigned short)

P7	Identifier of specific software version (fourth Byte)	Bit0: *reserved* Bit1: *reserved* Bit2: *reserved* Bit3: *reserved* Bit4: *reserved* Bit5: *reserved* Bit6: *reserved* Bit7: *reserved* (hex, unsigned short)

3.1.2.2 IDEN? Serial number and device information

Readout of the present value

With the command IDEN? device specific settings and informations can be read out. The PC-software Digicontrol 9310 use this command while searching 9310 devices on Ethernet network. To differentiate several devices with the same IP address (e.g. the default address) the Digicontrol 9310 softwaree uses the device serial number.

Host sends: <Address>sr<STX> IDEN?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,P4,P5,P6 ,P7,P8,P9,P10,P11,P12,P13,P14,P15,P16,P17 ,P18,P19,P20,P21,P22,P23<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	value
P1	Device type	String „9310“
P2	Version of device software	String
P3	Serial number	String
P4	Calibration date (dd.mm.yy)	String
P5	Station number	String
P6	9310 Device options	Bit0: 0 = DMS (standard) Bit0: 1 = Piezo Bit1: 0 = Anzeigegerät Bit1: 1= Blackbox (hex, unsigned short)
P7	9310 device communication interfaces	0x0: Standard (RS232/485) Bit0: 1 = PROFIBUS available Bit1: 0 = RS485 Bit1: 1 = Ethernet (hex, unsigned short)
P8	Counter an software changes with influence on serial communication	On every change of the 9310 software with influence on serial communication this value is incremented. The value starts on version V200208 with 0 (decimally, unsigned short)
P9	Identifier of specific software version (first Byte)	Bit0: = 0 standard; = 1 specific software Bit1: PC software totally locked Bit2: *reserved* Bit3: *reserved* Bit4: Upload locked Bit5: Download locked Bit6: *reserved* Bit7: Laboratory operation mode locked (hex, unsigned short)
P10	Identifier of specific software version (second Byte)	Bit0: General settings locked Bit1: Access authorization locked Bit2: Program access locked Bit3: *reserved*

		Bit4: *reserved* Bit5: *reserved* Bit6: *reserved* Bit7: *reserved* (hex, unsigned short)
P11	Identifier of specific software version (third Byte)	Bit0: Channel settings locked Bit1: Measuring mode locked Bit2: Sendor test locked Bit3: Evaluation locked Bit4: Switch points locked Bit5: Display locked Bit6: Envelope locked Bit7: *reserved* (hex, unsigned short)
P12	Identifier of specific software version (fourth Byte)	Bit0: *reserved* Bit1: *reserved* Bit2: *reserved* Bit3: *reserved* Bit4: *reserved* Bit5: *reserved* Bit6: *reserved* Bit7: *reserved* (hex, unsigned short)
P13	Subnet mask	Long (32 Bit) HEXFORMAT
P14	Gateway address	Long (32 Bit) HEXFORMAT
P15	Version of the ethernet modul	String
P16	Address of the serial interface	00 .. 99
P17	MAC address	String e.g. 11-22-33-44-55-66
P18	IP 1 access restriction	0 -> access not possible 1 -> access possible
P19	Host IP address 1	Long (32 Bit) HEXFORMAT
P20	IP 2 access restriction	0 -> access not possible 1 -> access possible
P21	Host IP address 2	Long (32 Bit) HEXFORMAT
P22	IP 3 access restriction	0 -> access not possible 1 -> access possible
P23	Host IP address 3	Long (32 Bit) HEXFORMAT

3.2 General settings

3.2.1 Access authorization

3.2.1.1 MPAS Master password

Input of a new value

With the command MPAS! ia new master password can be set.

Host sends: <Address>sr<STX>MPAS! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Master password	0..9999

Readout of the present value

With the command MPAS? the present master password can be read out.

Host sends: <Address>sr<STX>MPAS?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Master password	0..9999

3.2.1.2 UPAS User password

Input of a new value

With the command UPAS! you can set a new user password.

Host sends: <Address>sr<STX>UPAS! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	User password	0..9999

Readout the present value

With the command UPAS? you can read out the present user password.

Host sends: <Address>sr<STX>UPAS?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	User password	0..9999

3.2.1.3 PASP Activate password

Input of a new value

With the command PASP! the password protection can be activated/deactivated.

Host sends: <Address>sr<STX>PASP! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Password protection	0 →deactivate password protection 1 → activate password protection

Readout of the present value

With the command PASP? the present status of password protection can be read out.

Host sends: <Address>sr<STX>PASP?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Password protection	0 →deactivate password protection 1 → activate password protection

3.2.1.4 ZUGR Access levels

Input of a new value

With the command ZUGR! the access to the configuration menues with an activated password protection can be defined.

Host sends: <Address>sr<STX>ZUGR! P1,P2<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Access levels	GRUND → Basic menu MESSP → Measurement program STARE → Statistics reset KANAL → Channel settings MESSV → Measurement mode BEWER → Evaluation SCHAL → Switch points EINRI → Teach-in mode
P2	Access for User	0 → Access denied 1 → Access allowed

Readout of the present value

With the command ZUGR? the access settings on the configuration menues with an activated password protection can be read out. There are two possibilities – the direct request on the access for the different menues or a general request for access settings.

Direct request on the access of a menu

Host sends: <Address>sr<STX>ZUGR? P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P2<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Access levels	GRUND → Basic menu MESSP → Measurement program STARE → Statistics reset KANAL → Channel settings MESSV → Measurement mode BEWER → Evaluation SCHAL → Switch points EINRI → Teach-in mode
P2	Access for User	0 → Access denied 1 → Access allowed

General request for Access authorization

Host sends: <Address>sr<STX>ZUGR?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,P4,P5,P6,P7,P8<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	**Meaning**	**Wert**
P1	Access on menu General Settings for User	0 → Access denied 1 → Access allowed
P2	Access on menu Measurement program for User	0 → Access denied 1 → Access allowed
P3	Access on menu Statics Reset for User	0 → Access denied 1 → Access allowed
P4	Access on menu Cannel Settings for User	0 → Access denied 1 → Access allowed
P5	Access on menu Measurement Mode for User	0 → Access denied 1 → Access allowed
P6	Access on menu Evaluation for User	0 → Access denied 1 → Access allowed
P7	Access on menu Switch points for User	0 → Access denied 1 → Access allowed
P8	Access on configuration mode for User	0 → Access denied 1 → Access allowed

3.2.2 INFO

3.2.2.1 INFO Info line

With the command INFO? the info line of the device can be read out. It contains the number of the software version, the Serial number and the last date of adjustment

Host sends: <Address>sr<STX>INFO?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Info line	e.g.: „V200101,SN123456, 09.03.2001“

3.2.2.2 STAN Station number

Input of a new value

With the command STAN! a station number is transferred and saved for identifying the instrument in the interface mode.

Host sends: <Address>sr<STX>STAN! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Station number	Figure chain with length = 10 (e.g. 1234567890)

Readout of present value

With the command STAN? the station number of the instrument can be read out.

Host sends: <Address>sr<STX>STAN?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Station number	Figure chain with length = 10 (e.g. 1234567890)

3.2.3 LCD contrast

3.2.3.1 LCDK LCD contrast

Input of a new value

With the command LCDK! a new value for the LCD contrast can be set.

Host sends: <Address>sr<STX>LCDK! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	LCD contrast	Number from 0 (min.) to 10 (max.)

Readout of present value

With the command LCDK? the present value of the LCD contrast can be read out.

Host sends: <Address>sr<STX>LCDK?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	LCD contrast	Number from 0 (min.) to 10 (max.)

3.2.4 Menu language

3.2.4.1 SPRA Menu language

Input of a new value

With the command SPRA! a new menu language can be chosen.

Host sends: <Address>sr<STX>SPRA! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Menu language	DEUTSCH → Menu language German ENGLISCH → Menu language English FRANZOESISCH → Menu language French

Readout of the present value

With the command SPRA? the present menu language can be read out.

Host sends: <Address>sr<STX>SPRA?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Menu language	DEUTSCH → Menu language German ENGLISCH → Menu language English FRANZOESISCH → Menu language French

3.2.5 RS232 interface

Settings of the RS232-interface cannot be done via the RS232-interface

3.2.6 Ethernet Interface

3.2.6.1 IPEX IP address, Gateway address, subnet mask and UDP Port

Input of a new value

With the command IPEX! IP address, Gateway address, subnet mask and UDP port can be set.
The settings are only valid if the srial number of the commad is the same as the device.

Host sends: <Address>sr<STX>IPEX! P1,P2,P3,P4,P5<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Serial number of the DIGIFORCE 9310	Figure chain with length = 10 (e.g. 1234567890)
P2	IP address	Long (32 Bit) HEXFORMAT
P3	Subnet mask	Long (32 Bit) HEXFORMAT
P4	Gateway address	Long (32 Bit) HEXFORMAT
P5	Portnumber	Integer (16bit) decimally

Readout of the present value

With the command IPEX! IP address, Gateway address, subnet mask and UDP port can be read out.

Host sends: <Address>sr<STX> IPEX?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE anwers: <STX>P1,P2,P3,P4<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	value
P1	IP Adaress	Long (32 Bit) HEXFORMAT
P2	Subnet mask	Long (32 Bit) HEXFORMAT
P3	Gateway Address	Long (32 Bit) HEXFORMAT
P4	UDP-Port number	Integer (16bit) decimally

3.2.6.2 PORT UDP port number

Input of a new value

With the command PORT! the UDP port number can be set.

Host sends: <Address>sr<STX>PORT! P1 <ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	meaning	value
P1	Port number	Short (16 Bit)

Readout of the present value

With the command PORT? the UDP port number can be read out.

Host sends: <Address>sr<STX> PORT?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1 <ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	meaning	value
P1	Port number	Short (16 Bit)

3.2.6.3 IPVE Software version of the Ethernet modul

Readout of the present value

With the command IPVE? the software version of the Ethernet modul can be read.

Host sends: <Address>sr<STX> IPVE?<ETX>[<BCC>]
DIGIFORCE antswers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1 <ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	**meaning**	**value**
P1	Software version Ethernet modul	String

3.2.6.4 COMM coded/uncoded Ethernet communication

Input of a new value

With the command COMM! the UDP communication is set coded or uncoded

Host sends: <Address>sr<STX>COMM! P1 <ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	meaning	value
P1	Code style of UDP communication	0 → only coded 1 → coded and uncoded

Readout of the present value

With the command COMM! the UDP communication style (coded or coded and uncoded) can be read.

Host sends: <Address>sr<STX> COMM?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1 <ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	**meaning**	**value**
P1	Code style of UDP communication	0 → only coded 1 → coded and uncoded

3.2.6.5 HOST Host IP addresses access restriction

Input of a new value

With the command HOST! It is possible to set up to three Host IP addresses which are only allowed to communicate with the DIGIFORCE 9310. Ift he restriction of all addresses is set to 0 (access not possible) there is no restriction active.

Host sends: <Address>sr<STX>HOST! P1,P2,P3,P4,P5,P6<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	**meaning**	**value**
P1	IP 1 access restriction	0 -> access not possible 1 -> access possible
P2	Host IP address 1	Long (32 Bit) HEXFORMAT
P3	IP 2 access restriction	0 -> access not possible 1 -> access possible
P4	Host IP address 2	Long (32 Bit) HEXFORMAT
P5	IP 3 access restriction	0 -> access not possible 1 -> access possible
P6	Host IP address 3	Long (32 Bit) HEXFORMAT

Readout of the present value

With the command HOST? The status of the Host IP restriction can be read out.

Host sends: <Address>sr<STX> HOST?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,P4,P5,P6<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	**meaning**	**value**
P1	IP 1 access restriction	0 -> access not possible 1 -> access possible
P2	Host IP address 1	Long (32 Bit) HEXFORMAT
P3	IP 2 access restriction	0 -> access not possible 1 -> access possible
P4	Host IP address 2	Long (32 Bit) HEXFORMAT
P5	IP 3 access restriction	0 -> access not possible 1 -> access possible
P6	Host IP address 3	Long (32 Bit) HEXFORMAT

3.2.6.6 BUID? 9310 device identifiers

Readout of the present value

With the command BUID? Device specific identifiers can be read out.

Host sends: <Address>sr<STX>BUID?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,P4,P5,P6,P7,P8,P9,P10,P11,P12,P13,P14, P15,P16,P17,P18<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	value
P1	Device ID	ASCII String: „9310“
P2	Serial number	ASCII String
P3	Device name	ASCII String
P4	Status if the device is reserved from a host	0 → device not reserved <Address> → IP Address of the host (long (32Bit) HEXFORMAT)
P5	Counter on software changes of the Ethernet modul	On every change of the Ethernet modul software this value is incremented. The value start at version V200604 with 0 (decimally, unsigned short)
P6	MAC address	String e.g. 11-22-33-44-55-66
P7	IP address	Long (32 Bit) HEXFORMAT
P8	Subnet mask	Long (32 Bit) HEXFORMAT
P9	Gateway address	Long (32 Bit) HEXFORMAT
P10	Version of the Ethernet modul	String
P11	UDP Portnummer	Integer (16bit) decimally
P12	Version of the sensor hardware	Bit0 = 0 DMS (standard) Bit0 = 1 Piezo (hex, unsigned short)
P13	Version of the communikation hardware	0x0: standard (RS232/485) Bit0: 1 PROFIBUS available Bit1: 0 = RS485 Bit1: 1 = Ethernet Bit2: 0 = display device Bit2: 1 = Blackbox unsigned short
P14	Counter an software changes with influence on serial communication	On every change of the 9310 software with influence on serial communication this value is incremented. The value starts on version V200208 with 0 (decimally, unsigned short)
P15	Identifier of specific software version (first Byte)	Bit0: = 0 standard; = 1 specific software Bit1: PC software totally locked Bit2: *reserved* Bit3: *reserved* Bit4: Upload locked Bit5: Download locked Bit6: *reserved* Bit7: Laboratory operation mode locked (hex, unsigned short)
P16	Identifier of specific software version (second Byte)	Bit0: General settings locked Bit1: Access authorization locked Bit2: Program access locked Bit3: *reserved* Bit4: *reserved* Bit5: *reserved*

		Bit6: *reserved* Bit7: *reserved* (hex, unsigned short)
P17	Identifier of specific software version (third Byte)	Bit0: Channel settings locked Bit1: Measuring mode locked Bit2: Sendor test locked Bit3: Evaluation locked Bit4: Switch points locked Bit5: Display locked Bit6: Envelope locked Bit7: *reserved* (hex, unsigned short)
P18	Identifier of specific software version (fourth Byte)	Bit0: *reserved* Bit1: *reserved* Bit2: *reserved* Bit3: *reserved* Bit4: *reserved* Bit5: *reserved* Bit6: *reserved* Bit7: *reserved* (hex, unsigned short)

3.2.6.7 MAST! Ethernet host can reserve a DIGIFORCE 9310

This command can also used durin a running measurement.

Input of a new value

With the command MUST! The Ethernet host can temporary reserve an DIGIFORCE 9310 to block communicatin with other hosts.

Host sends: <Address>sr<STX> MAST! P1 <ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Time units in 100ms to reserve the 9310	0 -> device not reserved >0 -> time units in 100ms (unsigned long) e.g. a value of 10 reserve the device 1 second

Note:
Within the running time the host have to send a new MAST! command to ensure that the device is blocked for other hosts.

3.2.7 PROFIBUS-interface

3.2.7.1 PBAD PROFIBUS-address

This command only exists in the instrument if the option PROFIBUS DP is installed. If this is not the case, the request will be answered by NAK.

Input of a new value

With the command PBAD! the new PROFIBUS-address can be entered.

Host sends: <Address>sr<STX>PBAD! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Address for PROFIBUS-interface	0 ... 127

Readout of present value

With the command PBAD? the present PROFIBUS-address can be read out.

Host sends: <Address>sr<STX>PBAD?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Address for PROFIBUS-interface	0 ... 127

3.2.7.2 PBUE PROFIBUS-Control

This command only exists in the instrument if the option PROFIBUS DP is installed. If this is not the case, the request will be answered by NAK.

Input of a new value

With the command PBAD? the present PROFIBUS-control can be activated.

Host sends: <Address>sr<STX>PBUE! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Internal PROFIBUS-control	0 → Control deactivated 1 → Control activated

Readout of the present value

With the command PBUE? the status of the PROFIBUS-control can be read out.

Host sends: <Address>sr<STX>PBUE?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Internal PROFIBUS-control	0 → Control deactivated 1 → Control activated

3.2.7.3 PBIN Digital Inputs

This command only exists in the instrument if the option PROFIBUS DP is installed. If this is not the case, the request will be answered by NAK.

Input of a new value

With the command PBIN! it can be determined if the control signals are read in via the PLC-inputs or the PROFIBUS-inputs.

Host sends: <Adress>sr<STX>PBIN! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Source of control signals	SPS → use PLC-input signals PRO → use PROFIBUS-signals

Readout of the present value

With the command PBIN? the present source of control signals can be read out.

Host sends: <Address>sr<STX>PBIN?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Source of control signals	SPS → use PLC-input signals PRO → use PROFIBUS-signals

3.2.7.4 PMEM **Einstellen der Messmenü Anwahl bei Profibusverkehr**

The command PMEM is only active if the Pfofibus hardware is available (device option). Otherwise DIGIFORCE answers with NAK.

Input of a new value

With the command PMEM! can be specified, how the 9310 goes into measurement mode during Profibus access.

Host sends: <Address>sr<STX>PMEM! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Activation of measurement menu during Proifibus access	AUTO → 9310 goes into measurement mode if the AUTO bit is set STD → 9310 goes into measurement mode as soon as Profibus access is active

Readout of the present value

With the command PMEM? the status, how the 9310 goes into measurement mode during Profibus access can be read out.

Host sends: <Address>sr<STX>PMEM?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Activation of measurement menu during Proifibus access	AUTO → 9310 goes into measurement mode if the AUTO bit is set STD → 9310 goes into measurement mode as soon as Profibus access is active.

3.2.8 Reset

3.2.8.1 RSET Start Reset

With the command RSET! the Statistics-Reset can be started.

Host sends: <Address>sr<STX>RSET!<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

3.2.9 Block Measurement

3.2.9.1 MEFR Block Measurement

Input of a new value

With the command MEFR! the start of a new measurement can be blocked.

Host sends: <Address>sr<STX>MEFR! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Block Measurement	0 → Measurement blocked 1 → Measurement released

Readout of the present value

With the command MEFR? the present status of the block on measurement can be read out.

Host sends: <Address>sr<STX>MEFR?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Block Measurement	0 → Measurement blocked 1 → Measurement released

3.2.10 READY-Mode

3.2.10.1 RDYM Activate READY-Mode

Input of a new value

With the command RDYM! the Ready-Mode can be switched on and off. In READY-Mode PC-controlled, the device waits for the PC to release the signal before setting the READY-signal after a measurement.

Host sends: <Address>sr<STX>RDYM! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Ready-Mode	0 → READY-Mode normal 1 → READY-Mode PC-controlled

Readout of present value

With the command RDYM? the present READY-Mode can be read out.

Host sends: <Address>sr<STX>RDYM?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Ready-Mode	0 → READY-Mode normal 1 → READY-Mode PC-controlled

3.2.10.2 REDY Release READY-Signal

With the command REDY! the READY-Signal can be released. This command makes sense only when the READY-Mode PC-controlled is active.

Host sends: <Address>sr<STX>REDY?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

3.2.11 Administration of Measurement programs

3.2.11.1 DEFA Default-Initializing of all Measurement programs

With the command DEFA! all seven Measurement programs are initialized with Default-values. All entries are erased.

Host sends: <Address>sr<STX>DEFA! <ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

3.2.11.2 DEFP Default-Initializing of discrete Measurement programs

With the command DEFP! a discrete Measurement program is initialized with es Meßprogramm mit Default-Werten initialisiert. Alle Eingaben in diesem Programm werden gelöscht.

Host sends: <Address>sr<STX>DEFP! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Program number	0 ... 7

3.2.11.3 CMPR Copy a Measurement program

With the command CMPR! a single Measurement program (source program number) can be copied to a group of Measurement program memory cells (limited by TARGET-START program number and TARGET-END program number). The necessary condition is that the TARGET-END program number is greater or equal to the TARGET-START program number.

Host sends: <Address>sr<STX>CMPR! P1,P2,P3<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Source program number	0 ... 7
P2	Target-START program number	0 .. Target-END program number
P3	Target-END program number	Target-START program number ...7

3.2.12 OK/NOK graphical display

3.2.12.1 INGR Selection OK/NOK graphical display

Input of a new value

With the command INGR! the symbol for display of an OK or NOK measurement in the relative Measurement menu can be chosen.

Host sends: <Address>sr<STX>INGR! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Choose the graphical symbol	TEXT → display OK/NOK-text SMILEY → display a round Smiley

Readout of the present value

With the command INGR? the presently chosen graphical symbol can be read out.

Host sends: <Address>sr<STX>INGR?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Choose the graphical symbol	TEXT → display OK/NOK-text SMILEY → display a round Smiley

3.2.13 Choose Measurement menu

3.2.13.1 MENU Choose Measurement menu

With the command MENU! the Measurement menu can be chosen.

Host sends: <Address>sr<STX>MENU! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Measurement display curve	M1
	Meas. display single window evaluation	M2
	Meas. display NOK-statistics of windows	M3
	Meas. display total result	M4
	Meas. display general curve data	M5

With the command MENU? the presently configured Measurement menu can be read out.

Host sends: <Address>sr<STX>MENU?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Measurement display curve	M1
	Meas. display single window evaluation	M2
	Meas. display NOK-statistics of windows	M3
	Meas. display total result	M4
	Meas. display general curve data	M5

3.2.14 LCD update

3.2.14.1 UPDA Control LCD update

Input of a new value

With the command UPDA! the LCD update can be switched on and off. After power up of the device the update is switched on automatically.

Host sends: <Address>sr<STX>UPDA! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	LCD update	1 → Switch on update 0 → Switch off update

Readout of the present value

With the command UPDA? the present status of LCD update can be read out.

Host sends: <Address>sr<STX>UPDA?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	LCD update	1 → Switch on update 0 → Switch off update

3.2.15 Burster Logo activation

3.2.15.1 LOGO **Burster Logo**

Input of a new value

With the command LOGO! the burster Logo in the start up menue can bei switched ON/OFF

Host sends: <Address>sr<STX>LOGO! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	**Meaning**	**value**
P1	Burster Logo during start up menue	1 → Logo appears 0 → Logo does not appear

Readout of the present value

With the command LOGO! the status of the burster Logo during start up menue can bei read out.

Host sends: <Address>sr<STX>LOGO?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE antswers: <EOT>

Meaning of parameter Pn

Parameter	**Meaning**	**value**
P1	Burster Logo during start up menue	1 → Logo appears 0 → Logo does not appear

3.3 Program specific setup

3.3.1 General programm settings

3.3.1.1 PRNR Program number

Input of a new value

With the command PRNR! a new Measurement program can be chosen.

Host sends: <Address>sr<STX>PRNR! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Measurement program number	0 ... 7

Readout of the present value

With the command PRNR? the number of the presently configured Measurement programs can be read out.

Host sends: <Address>sr<STX>PRNR?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Measurement program number	0 ... 7

3.3.1.2 PNAM Program name

Input of a new value

With the command PNAM! the name for the present Measurement program can be given. Choosing a name does not select a new Measurement program. The same name could be given to multiple programs.

Host sends: <Address>sr<STX>PNAM! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of paramter Pn

Parameter	Meaning	Value
P1	Measurement program name	Character chain of max. 12 figures

Readout of the present value

With the command PNAM? the name of the present Measurement program can be read out.

Host sends: <Address>sr<STX>PNAM?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of paramter Pn

Parameter	Meaning	Value
P1	Measurement program name	Character chain of max. 12 figures

3.3.1.3 NAME **Program name (on each measurement program)**

Input of a new value

With the command NAME! a name for any measurement program can be set. The command does not choise a measure program.

Host sends: <Address>sr<STX>NAME! P1,P2 <ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of paramter Pn

Parameter	**Meaning**	**value**
P1	Measurement prog. number	0 ... 7
P2	Measurement prog. name	String max. 12 character

Readout of the present value

With the command NAME! the name for any measurement program can be read out.

Host sends: <Address>sr<STX>NAME? P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P2<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of paramter Pn

Parameter	**Meaning**	**value**
P1	Measurement prog. number	0 ... 7
P2	Measurement prog. name	String max. 12 character

3.3.1.4 BDEX User-defined unit X-channel

Input of a new value

With the command BDEX! a string as user-defined unit for Measurement channel X can be configured.
This input is valid only for the present Measurement program.

Host sends: <Address>sr<STX>BDEX! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	User-defined unit X-channel	Character chain of max. 4 figures

Readout of present value

With the command BDEX? the user-defined unit of the X-channel can be read out.

Host sends: <Address>sr<STX>BDEX?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	User-defined unit X-channel	Character chain of max. 4 figures

3.3.1.5 BDEY User-defined unit Y-channel

Input of a new value

With the command BDEY! a string as user-defined unit for Measurement channel Y can be configured.
This input is valid only for the present Measurement program.

Host sends: <Address>sr<STX>BDEY! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	User-defined unit Y-channel	Character chain of max. 4 figures

Readout of the present value

With the command BDEY? the user-defined unit of the Y-channel can be read out.

Host sends: <Address>sr<STX>BDEY?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	User-defined unit Y-channel	Character chain of max. 4 figures

3.3.1.6 AUSC Switch on/off of Auto-Scaling

Input of a new value

With the command AUSC! the Auto-Scaling of the graphical display can be switched on and off. In case of a switched on Auto-Scaling the device will look always for the optimal scale range. When this feature is switched off the scaling is done by the manually configurable scale data.
This input is valid only for the present Measurement program.

Host sends: <Address>sr<STX>AUSC! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Auto-Scaling	1 → Switch on Auto-Scaling 0 → Switch off Auto-Scaling

Readout of the present value

With the command AUSC? the present status of Auto-Scaling can be read out.

Host sends: <Address>sr<STX>AUSC?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Auto-Scaling	1 → Switch on Auto-Scaling 0 → Switch off Auto-Scaling

3.3.1.7 SCAL Manual Scaling of Measurement curve

Input of a new value

With the command SCAL! the graphical display limitations can be set manually. This command can only be sent when Auto-Scaling is switched off.
This input is valid only for the present Measurement program.

Host sends: <Address>sr<STX>SCAL! P1,P2,P3,P4<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Lower display limit X	Floating point number without unit
P2	Upper display limit X	Floating point number without unit
P3	Lower display limit Y	Floating point number without unit
P4	Upper display limit Y	Floating point number without unit

Readout of the present value

With the command SCAL? the present manual scale values can be read out.

Host sends: <Address>sr<STX>SCAL?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,P4<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Lower display limit X	Floating point number without unit
P2	Upper display limit X	Floating point number without unit
P3	Lower display limit Y	Floating point number without unit
P4	Upper display limit Y	Floating point number without unit

3.3.1.8 NIOA NOK-display in percent/absolute

Input of a new value

With the command NIOA! the display of NOK-measurements can be switched to show the absolute NOK-quantity or the percentage of NOK-parts in relation to the total number of measurements in the M4-measurement menu. This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>NIOA! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	NOK-display	PROZ → Display in percentage ABS → Display in absolute figures

Readout of the present value

With the command NIOA? the present status of the NOK-display can be read out.

Host sends: <Address>sr<STX>NIOA?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	NOK-display	PROZ → Display in percentage ABS → Display in absolute figures

3.3.2 Measurement menu release

3.3.2.1 MFRE Measurement menues

Input of a new value

With the command MFRE! the display of the separate measurement menues can be activated.
This setting depends on the measurement program.

Host sends: <Address>sr<STX>MFRE! P1,P2<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Measurement menu	MESS1 → Menu 1: Curve display MESS2 → Menu 2: Window evaluation MESS3 → Menu 3: Window statistics MESS4 → Menu 4: Total evaluation MESS5 → Menu 5: General curve data
P2	Display	0 → do not show menu 1 → show menu

Readout of the present value

With the command MFRE? the display of the separate measurement menues can be read out.
There are two possibilities, the direct request for release of a single menu or the general request for menu release.

Direct request for release of a single menu

Host sends: <Address>sr<STX>MFRE? P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P2<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Measurement menu	MESS1 → Menu 1: Curve display MESS2 → Menu 2: Window evaluation MESS3 → Menu 3: Window statistics MESS4 → Menu 4: Total evaluation MESS5 → Menu 5: General curve data
P2	Display	0 → do not show menu 1 → show menu

General request for menu release

Host sends: <Address>sr<STX>MFRE?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,P4,P5<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	**Meaning**	**Wert**
P1	Display Measurement menu 1	0 → do not show menu 1 → show menu
P2	Display Measurement menu 2	0 → do not show menu 1 → show menu
P3	Display Measurement menu 3	0 → do not show menu 1 → show menu
P4	Display Measurement menu 4	0 → do not show menu 1 → show menu
P5	Display Measurement menu 5	0 → do not show menu 1 → show menu

3.3.3 Connection of sensors

3.3.3.1 EINX Unit X

Input of a new value

With the command EINX! the unit for Measurement channel X can be entered.
This setting is valid only for the present measurement program.

Host sends: <Address>sr<STX>EINX! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Unit X-channel	0 → user-defined 1 → ´mm´ 2 → ´????´

Readout of the present value

With the command EINX? the configured unit for X-channel can be read out.

Host sends: <Address>sr<STX>EINX?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Unit X-channel	0 → user-defined 1 → ´mm´ 2 → ´????´

3.3.3.2 EINY Unit Y

Input of a new value

With the command EINY! the unit for Measurement channel Y can be entered.
This input is valid only for the present Measurement program.

Host sends: <Address>sr<STX>EINY! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Unit Y-channel	0 → user-defined 1 → ´ N´ 2 → ´ kN´ 3 → ´ Nm´

Readout of present value

With the command EINY? the configured unit for Y-channel can be read out.

Host sends: <Address>sr<STX>EINY?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Unit Y-channel	0 → user-defined 1 → ´ N´ 2 → ´ kN´ 3 → ´ Nm´

3.3.3.3 XPOT Channel setting X Potentiometer

Input of a new value

With the command **XPOT**! the sensor channel X can be set to potentiometer.
This input is valid only in the present measurement program.

Host sends: <Address>sr<STX>**XPOT**! <ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Readout of the present value

With the command **XPOT**? the present channel setting of X-channel resp. potentiometer can be read out.

Host sends: <Address>sr<STX>**XPOT**?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Potentiometer (yes or no)	1 → Potentiometer 0 → no Potentiometer but standard signal

3.3.3.4 XNOR Channel setting X standard signal

Input of a new value

With the command **XNOR**! the sensor channel X can be set to standard signal.
This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>**XNOR**! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Input range	„5V" → 5V input range „10V"→ 10V input range

Readout of the present value

With the command **XNOR**? the present channel setting of the X-channel resp. standard signal can be read out.

Host sends: <Address>sr<STX>**XNOR**?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Potentiometer (yes or no)	1 → Potentiometer 0 → no Potentiometer but standard signal

Parameter	Meaning	Value
P2	Input range	„5V" → 5V input range „10V"→ 10V input range

3.3.3.5 YNOR Channel setting Y standard signal

Input of a new value

With the command **YNOR**! the sensor channel Y can be set to standard signal.
This input is valid only for the present measurement program.

Host sends: <Address>sr<STX> **YNOR**! <ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Readout of the present value

With the command **YNOR**? the present channel setting of the Y-channel resp. standard signal can be read out.

Host sends: <Address>sr<STX>**YNOR**?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	**Meaning**	**Value**
P1	Standard signal (yes or no)	1 → Standard signal 0 → not a standard signal but potentiometer

3.3.3.6 YDMS Channel setting Y strain gage

Input of a new value

With the command **YDMS!** the sensor channel Y can be set to strain gage input.
This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>**YDMS**! P1,P2,P3,P4 <ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

P1	Input terminal voltage	2.5V → 2,5V-input voltage 5V →5V-input voltage
P2	Sensitivity in [mV/V]	**Positive** Floating-point numbers, e.g. 1.498
P3	End value of sensor	Floating-point numbers, e.g. 150.0
P4	Used range of sensor, how great are the maximum sensor forces that occur in the given application?	Floating-point numbers, e.g. 125.0

Readout of present value

With the command **YDMS**? the present channel setting for channel Y resp. strain gage signal can be read out.

Host sends: <Address>sr<STX>**YDMS**?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,P4,P5<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	**Meaning**	**Value**
P1	Strain gage (yes or no)	1 → Strain gage 0 → no strain gage but standard signal
P2	Input terminal voltage	2.5V → 2,5V-input voltage 5V →5V-input voltage
P3	Sensitivity in [mV/V]	Floating-point numbers, e.g. 1.498
P4	End value of sensor	Floating-point numbers, e.g. 150.0
P5	Used range of sensor, how great are the maximum sensor forces that occur in the given application?	Floating-point numbers, e.g. 125.0

3.3.3.7 YPIE Set the range for Piezo sensor

Setting of the present value

With the command YPIE! the input range of the PIEZO-input can be selected.
This command is available only with the PIEZO option.

Host sends: <Address>sr<STX>YPIE! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Input range	1NC → Input range 1 nC 2NC → Input range 2 nC 5NC → Input range 5 nC 10NC → Input range 10 nC 20NC → Input range 20 nC 50NC → Input range 50 nC 100NC → Input range 100 nC 200NC → Input range 200 nC 400NC → Input range 400 nC

Readout of the present value

With the command YPIE? the present input range can be read out.

Host sends: <Address>sr<STX>YPIE?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Input range	1NC → Input range 1 nC 2NC → Input range 2 nC 5NC → Input range 5 nC 10NC → Input range 10 nC 20NC → Input range 20 nC 50NC → Input range 50 nC 100NC → Input range 100 nC 200NC → Input range 200 nC 400NC → Input range 400 nC

3.3.3.8 PIKZ! Short circuit for the Piezo amplifier (close and open)

Input of a new value

The command PIKZ! is valid only for the piezo version of the DIGIFORCE. PIKZ! value ´1´ must be send to unload the charging amplifier and close the Piezo-input after every measurement. Before you START the next measurement you have to send the PIKZ! command with the value ´0´ (open).

Host sends: <Adress>sr<STX> PIKZ! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Piezo input (charging amplifier)	1 → closed 0 → open (only while the measurement is running)

Readout the present status of the charging amplifier (open or close)

With the command PIKZ? you can readout the present status (open or close) of the charging amplifier. The command PIKZ? is valid only for the piezo version of the DIGIFORCE.

Host sends: <Address>sr<STX> PIKZ?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Piezo input (charging amplifier)	1 → closed (unloading) 0 → open (only while the measurement is running)

3.3.3.9 FILX Filter X

Input of a new value

With the command FILX! the filter on the X-channel can be activated.
This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>FILX! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Critical frequency of the X-channel filter	´AUS´ → Filter off ´5Hz´ → Filter with f_g=5Hz ´10Hz´ → Filter with f_g=10Hz ´25Hz´ → Filter with f_g=25Hz ´50Hz´ → Filter with f_g=50Hz ´100Hz´ → Filter with f_g=100Hz ´200Hz´ → Filter with f_g=200Hz ´400Hz´ → Filter with f_g=400Hz

Readout of the present value

With the command FILX? the present settings for channel X can be read out.

Host sends: <Address>sr<STX>FILX?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Critical frequency of the X-channel filter	´AUS´ → Filter off ´5Hz´ → Filter with f_g=5Hz ´10Hz´ → Filter with f_g=10Hz ´25Hz´ → Filter with f_g=25Hz ´50Hz´ → Filter with f_g=50Hz ´100Hz´ → Filter with f_g=100Hz ´200Hz´ → Filter with f_g=200Hz ´400Hz´ → Filter with f_g=400Hz

3.3.3.10 FILY Filter Y

Input of a new value

With the command FILY! the filter on the Y channel can be activated.
This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>FILY! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Critical frequency of the Y-channel filter	´AUS´ → Filter off ´5Hz´ → Filter with f_g=5Hz ´10Hz´ → Filter with f_g=10Hz ´25Hz´ → Filter with f_g=25Hz ´50Hz´ → Filter with f_g=50Hz ´100Hz´ → Filter with f_g=100Hz ´200Hz´ → Filter with f_g=200Hz ´400Hz´ → Filter with f_g=400Hz

Readout of the present value

With the command FILY? the present setting for the filter on channel Y can be read out.

Host sends: <Address>sr<STX>FILY?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Critical frequency of the Y-channel filter	´AUS´ → Filter off ´5Hz´ → Filter with f_g=5Hz ´10Hz´ → Filter with f_g=10Hz ´25Hz´ → Filter with f_g=25Hz ´50Hz´ → Filter with f_g=50Hz ´100Hz´ → Filter with f_g=100Hz ´200Hz´ → Filter with f_g=200Hz ´400Hz´ → Filter with f_g=400Hz

3.3.3.11 INVX Inverting X-channel

Input of a new value

With the command INVX! the inverting of the X-channel can be switched on and off. **If the device is readjusted or if the adjustment values are loaded from the EEPROM the inverting is switched off automatically. This is valid also for a default adjustment in case of loss of memory.** This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>INVX! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Inverting X-channel	1 → Inverting switched on 0 → Inverting switched off

Readout of the present value

With the command INVX? the present status of the X-channel inverting can be read out.

Host sends: <Address>sr<STX>INVX?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Inverting X-channel	1 → Inverting switched on 0 → Inverting switched off

3.3.3.12 INVY Inverting Y-channel

Input of a new value

With the command INVY! the inverting of the Y-channel can be switched on and off. **If the device is readjusted or if the adjustment values are loaded from the EEPROM the inverting is switched off automatically. This is valid also for a default adjustment in case of loss of memory.** This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>INVY! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Inverting Y-channel	1 → Inverting switched on 0 → Inverting switched off

Readout of the present value

With the command INVY? the present status of Y-channel inverting can be read out.

Host sends: <Address>sr<STX>INVY?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Inverting Y-channel	1 → Inverting switched on 0 → Inverting switched off

3.3.3.13 SKAX Scalar values X

Input of a new value

With the command SKAX! the scalar values for the X-channel can be transferred.
The entered values, however, are only taken over after a new calibration for this channel has been calculated (DOKX/DOKY)! This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>SKAX! P1,P2<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	lower scalar value X-channel	Floating-point number
P2	upper scalar value X-channel	Floating-point number

Readout of the present value

With the command SKAX? the present scalar values can be read out.

Host sends: <Address>sr<STX>SKAX?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	lower scalar value X-channel	Floating-point number
P2	upper scalar value X-channel	Floating-point number

3.3.3.14 SKAY Scalar value Y

Input of a new value

With the command SKAY! the scalar values for the Y-channel can be transferred.
The entered values, however, are only taken over after a new calibration for this channel has been calculated (DOKX/DOKY)! This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>SKAY! P1,P2<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	lower scalar value Y-channel	Floating-point number
P2	upper scalar value Y-channel	Floating-point number

Readout of the present value

With the command SKAY? the present scalar values for the Y-channel can be read out.

Host sends: <Address>sr<STX>SKAY?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	lower scalar value Y-channel	Floating-point number
P2	upper scalar value Y-channel	Floating-point number

3.3.3.15 KALX Calibration values X

Input of a new value

With the command KALX! the calibration values for the X-channel can be transferred.
The entered values, however, are only taken over after a new calibration for this channel has been calculated (DOKX/DOKY)! This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>KALX! P1,P2<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	lower calibration value X-channel	Floating-point number
P2	upper calibration value X-channel	Floating-point number

Readout of the present value

With the command KALX? the present calibration values can be read out.

Host sends: <Address>sr<STX>KALX?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	lower calibration value X-channel	Floating-point number
P2	upper calibration value X-channel	Floating-point number

3.3.3.16 KALY Calibration values Y

Input of a new value

With the command KALY! the calibration values for the Y-channel can be transferred.
The entered values, however, are only taken over after a new calibration for this channel has been calculated (DOKX/DOKY)! This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>KALY! P1,P2<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	lower calibration value Y-channel	Floating-point number
P2	upper calibration value Y-channel	Floating-point number

Readout of the present value

With the command KALY? the present calibration values can be read out.

Host sends: <Address>sr<STX>KALY?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	lower calibration value Y-channel	Floating-point number
P2	upper calibration value Y-channel	Floating-point number

3.3.3.17 MKLX Measure calibration value X-channel

With the command MKLX! the calibration values for the X-channel can be measured. It must be indicated which calibration value is to be measured.
The entered values, however, are only taken over after a new calibration for this channel has been calculated (DOKX/DOKY)! This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>MKLX! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Type of calibration value	´UNTEN´ → Measure lower cal-val ´OBEN´ → Measure upper cal-val

3.3.3.18 MKLY Measure calibration value Y-channel

With the command MKLY! the calibration values for the Y-channel can be measured. It must be indicated which calibration value is to be measured.
The entered values, however, are only taken over after a new calibration for this channel has been calculated (DOKX/DOKY)! This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>MKLY! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Type of calibration value	´UNTEN´ → Measure lower cal-val ´OBEN´ → Measure upper cal-val

3.3.3.19 DOKX Execute calibration on X-channel

With the command DOKX! a calculation of calibration data for the X-channel is started. Only then the calibration data is taken over. During calculation the data is being checked. If this check has a negative outcome the device sends a NAK and does not take over the calibration. The command has no further parameters. This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>DOKX!<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

3.3.3.20 DOHX Execute calibration on X-channel with envelope correction

With the command DOHX! a calculation of calibration data for the X-channel is started. Only then the calibration data is taken over. During calculation the data is being checked. If this check has a negative outcome the device sends a NAK and does not take over the calibration. The command has no further parameters. An envelope will be corrected with the new calibration value if available.
This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>DOHX!<ETX>[<BCC>]
DIGIFORCE answers <ACK>
Host sends: <EOT>

3.3.3.21 DOKY Execute calibration on Y-channel

With the command DOKY! a calculation of calibration data for the Y-channel is started. Only then the calibration data is taken over. During calculation the data is being checked. If this check has a negative outcome the device sends a NAK and does not take over the calibration. The command has no further parameters.
This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>DOKY!<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

3.3.3.22 DOHY Execute calibration on Y-channel with envelope correction

With the command DOHY! a calculation of calibration data for the Y-channel is started. Only then the calibration data is taken over. During calculation the data is being checked. If this check has a negative outcome the device sends a NAK and does not take over the calibration. The command has no further parameters. An envelope will be corrected with the new calibration value if available.
This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>DOHY!<ETX>[<BCC>]
DIGIFORCE anwers: <ACK>
Host sends: <EOT>

3.3.4 Zero signal

3.3.4.1 XORG? Originally zero signal X channel

Readout of the present value

With the command XORG? the originally zero signal of channel X can be read out.

Host sends: <Address>sr<STX>XORG?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	value
P1	Zero signal channel X	Float value

3.3.4.2 YORG? Originally zero signal Y channel

Readout of the present value

With the command YORG? the originally zero signal of channel Y can be read out.

Host sends: <Address>sr<STX>YORG?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Bedeutung	Wert
P1	Zero signal channel Y	Float value

3.3.5 Sensor test

3.3.5.1 SETX Teach-in sensor test X

Input of a new value

With the command SETX! the value for the X-channel is measured which later on functions as a reference value for the sensor test X. During the sensor test this value must be met within the indicated tolerances.
This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>SETX!<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Readout of the present value

With the command SETX? the presently measured sensor test value for X-channel can be read out.

Host sends: <Address>sr<STX>SETX?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Value for sensor test X	Floating-point number with unit, e.g. ´1.2345mm´

3.3.5.2 SETY Teach-in sensor test Y

Input of a new value

With the command SETY! the value for the Y-channel is measured which later on functions as a reference value for the sensor test Y. During the sensor test this value must be met within the indicated tolerances.
This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>SETY!<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Readout of the present value

With the command SETY? the presently measured sensor test value for Y-channel can be read out.

Host sends: <Address>sr<STX>SETY?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	**Meaning**	**Value**
P1	Value for sensor test Y	Floating-point number with unit, e.g. ´1.2345mm´

3.3.5.3 STTX Tolerance sensor test X

Input of a new value

With the command STTX! the tolerance value for sensor test X can be entered. The value is given in the present X-unit. This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>STTX! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Tolerance value for sensor test X	Floating-point numbers without unit

Readout of the present value

With the command STTX? the present tolerance value for sensor test X can be read out.

Host sends: <Address>sr<STX>STTX?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Tolerance value for sensor test X	Floating-point numbers without unit

3.3.5.4 STTY Tolerance sensor test Y

Input of a new value

With the command STTX! the tolerance value for sensor test X can be entered. The value is given in the present X-unit. This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>STTY! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Tolerance value for sensor test Y	Floating-point numbers without unit

Readout of the present value

With the command STTY? the present tolerance value for sensor test Y can be read out.

Host sends: <Address>sr<STX>STTY?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Tolerance value for sensor test Y	Floating-point numbers without unit

3.3.5.5 STST Execute sensor test

With the command STST? the sensor test is executed and the result is transferred.

Host sends: <Address>sr<STX>STST?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Result of sensor test	´IO´ → Sensor is within the specs. ´NIOX´ → X-Sensor is out of tolerance ´NIOY´ → Y-Sensor is out of tolerance ´NIOXY´ → X- und Y-Sensor are out of tolerance

3.3.5.6 STWX Reference value for sensor test channel X

Input of a new value

With the command STWX! the reference value for sensor test X can be entered. The value is given in the present X-unit. This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>STWX! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Reference value for sensor test X	Floating-point numbers without unit

Readout of the present value

With the command STWX? the present reference value for sensor test X can be read out.

Host sends: <Address>sr<STX>STWX?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Reference value for sensor test X	Floating-point numbers without unit

3.3.5.7 STWY Reference value for sensor test channel Y

Input of a new value

With the command STWY! the reference value for the sensor test Y can be entered. The value is given in the present X-unit. This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>STWY! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Reference value for sensor test Y	Floating-point numbers without unit

Readout of the present value

With the command STWX? the present reference value for sensor test Y can be read out.

Host sends: <Address>sr<STX>STWY?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Reference value for sensor test Y	Floating-point numbers without unit

3.3.6 Setting of measurement function

3.3.6.1 MFKT Measurement function

Input of a new value

With the command MFKT! the measurement function for the present program can be transferred.

Host sends: <Address>sr<STX>MFKT! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Measurement function	´Y=F(X)´ → Meas. funct. $y=f_{(x)}$ ´Y=F(XT)´ → Meas. funct. $y=f_{(x,t)}$ ´Y=F(T)´ → Meas. funct. $y=f_{(t)}$

Readout of the present value

With the command MFKT? the present measurement function can be read out.

Host sends: <Address>sr<STX>MFKT?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Measurement function	´Y=F(X)´ → Meas. funct. $y=f_{(x)}$ ´Y=F(XT)´ → Meas. funct. $y=f_{(x,t)}$ ´Y=F(T)´ → Meas. funct. $y=f_{(t)}$

3.3.6.2 RAST **Sampling rate**

Input of a new value

With the command RAST! the value for the sampling rate can be entered. It needs to be decided if it is a measurement function depending on displacement or time. As a result of this difference the value limits for the parameters change and in case these limits are passed the device answers with NAK.
This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>RAST! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

for measurement function = ´Y=F(X)´

Parameter	Meaning	Value
P1	Sampling rate	0.001 ... 1000.0

for measurement function = ´Y=F(XT)´ oder ´Y=F(T)´

Parameter	Meaning	Value
P1	Sampling grid [ms]	0.2 ... 500.0

Readout of the present value

With the command RAST? the present value of the sampling raster can be read out.

Host sends: <Address>sr<STX>RAST?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Sampling grid	Floating-point numbers with unit, e.g. ´1.23ms´

3.3.6.3 BZUG **Reference**

Input of a new value

With the command BZUG! the curve reference can be selected.
This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>BZUG! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Curve reference	´ABS´ → Absolute reference ´TRI´ → Trigger reference ´END´ → Block reference ‘ BLF’ → Block window

Readout of the present value

With the command RDYM? the present curve reference can be read out.

Host sends: <Address>sr<STX>BZUG?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Curve reference	´ABS´ → Absolute reference ´TRI´ → Trigger reference ´END´ → Block reference ‘ BLF’ → Block window

3.3.6.4 TRGP **Trigger point**

Input of a new value

With the command TRGP! the trigger point can be entered. This command makes sense only if the trigger reference has been chosen. If this is not the case then the entered value has no effect.
This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>TRGP! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	trigger point	Floating-point numbers with unit, e.g. ´1.234N´

Readout of the present value

With the command TRGP? the present trigger point can be read out.

Host sends: <Address>sr<STX>TRGP?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	trigger point	Floating-point numbers with unit, e.g. ´1.234N´

3.3.6.5 UKPT **Turning Point**

Input of a new value

With the command UKPT! the turning point can be defined. Up to this point the curve is displayed and evaluated. This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>UKPT! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Turning point	´XMAX´ → Turning point at maximum displacement ´YMAX´ → Turning point at maximum force

Readout of the present value

With the command UKPT? the present turning point can be read out.

Host sends: <Address>sr<STX>UKPT?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Turning point	´XMAX´ → Turning point at maximum displacement ´YMAX´ → Turning point at maximum force

3.3.6.6 STMD **Start mode**

Input of a new value

With the command STMD! the Start/Stop-mode can be defined. It is determined if the measurement is controlled externally or internally after reaching a certain value limit.
This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>STMD! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Start mode	´EXTERN´ → Start/Stop controlled by PLC / Field bus ´INTERNX´ → Start/Stop controlled by internal threshold value X-channel

Readout of the present value

With the command STMD? the present Start/Stop-mode can be read out.

Host sends: <Address>sr<STX>STMD?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Start mode	´EXTERN´ → Start/Stop controlled by PLC / Field bus ´INTERNX´ → Start/Stop controlled by internal threshold value X-channel

3.3.6.7 STSP **Start/Stop internal**

Input of a new value

With the command STSP! the Start/Stop-values for internal start X can be entered.
This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>STSP! P1,P2<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Internal start X	Floating-point numbers without unit
P2	Internal stop X	Floating-point numbers without unit

Readout of the present value

With the command STSP? the present Start/Stop-values can be read out.

Host sends: <Address>sr<STX>STSP?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Internal start X	Floating-point numbers without unit
P2	Internal stop X	Floating-point numbers without unit

3.3.6.8 STAR PLC Tare function

With the command STAR the PLC tare function can be configured or read out.

Configure PLC tare function:

Host sends: <Address>sr<STX> STAR! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	PLC tare function	X → Only tare X is started Y → Only tare Y is started X+Y -> Tare X and Y are started

Read out an already configured PLC tare function:

Host sends: <Address>sr<STX>STAR??<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1 <ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	PLC tare function	X → Only tare X is started Y → Only tare Y is started X+Y -> Tare X and Y are started

3.3.6.9 TARA Start Function via interface

With the command TARA! the tare X or Y can be started via the serial interface.

Host sends: <Address>sr<STX> TARA! P1,P2 <ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Tare X or Y	X → Tare X Y → Tare Y
P2	Start or take back tare	1 → Start tare function 0 → Take back tare function

3.3.7 Evaluation

3.3.7.1 Window

3.3.7.1.1 **FTYP** Type of window

Input of a new value

With the command FTYP! the type of evaluation window can be defined. The input makes sense only if window evaluation is set as type of evaluation. There can be only one window of the types block and online each. If a window of the same type already exists the device answers with NAK.
This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>FTYP! P1,P2<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Window number	1 ... 3
P2	Window type	´AUS´ → Window is switched off ´DURCH´ → Window is a pass through type ´BLOCK´ → Window is a block type ´ONLINE´ → Window is an online type

Readout of the present value

With the command FTYP? the window type can be read out.

Host sends: <Address>sr<STX>FTYP? P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P2<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Window number	1 ... 3
P2	Window type	´AUS´ → Window is switched off ´DURCH´ → Window is a pass through type ´BLOCK´ → Window is a block type ´ONLINE´ → Window is an online type

3.3.7.1.2 **FGRZ** Window borders

Input of a new value

With the command FGRZ! the borders resp. the position of the evaluation windows can be transferred. The upper window borders have to be greater than the lower borders.
This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>FGRZ! P1,P2,P3,P4,P5<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Window number	1 ... 3
P2	Window border Xmin	Floating-point number without unit
P2	Window border Xmax	Floating-point number without unit
P2	Window border Ymin	Floating-point number without unit
P2	Window border Ymax	Floating-point number without unit

Readout of the present value

With the command FGRZ? the present window borders can be read out.

Host sends: <Address>sr<STX>FGRZ? P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P2,P3,P4,P5<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Window number	1 ... 3
P2	Window border Xmin	Floating-point number without unit
P2	Window border Xmax	Floating-point number without unit
P2	Window border Ymin	Floating-point number without unit
P2	Window border Ymax	Floating-point number without unit

3.3.7.1.3 **FEIN** Window entry

Input of a new value

With the command FEIN! the side for window entry can be defined. The entry for an Online window is always on the left side, no matter how the setup is done.
This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>FEIN! P1,P2<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Window number	1 ... 3
P2	Side of entry	´LINKS´ → Entry on left ´RECHTS´ → Entry on right ´OBEN´ → Entry on top ´UNTEN´ → Entry on bottom ´EGAL´ → Entry does not matter

Readout of the present value

With the command FEIN? the side of entry to the window can be read out.

Host sends: <Address>sr<STX>FEIN? P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P2<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Window number	1 ... 3
P2	Side of entry	´LINKS´ → Entry on left ´RECHTS´ → Entry on right ´OBEN´ → Entry on top ´UNTEN´ → Entry on bottom ´EGAL´ → Entry side does not matter

3.3.7.1.4 **FAUS** Window exit

Input of a new value

With the command FAUS! the side for window entry can be defined. The exit is always on the right for an Online window, no matter how the setup is done. For the block window type the exit is not relevant.
This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>FAUS! P1,P2<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value	
P1	Window number	1 ... 3	
P2	Side of exit	´LINKS´	→ Exit to left
		´RECHTS´	→ Exit to right
		´OBEN´	→ Exit to top
		´UNTEN´	→ Exit to bottom
		´EGAL´	→ Exit side does not matter

Readout of the present value

With the command FAUS? the side set for window exit can be read out.

Host sends: <Address>sr<STX>FAUS? P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P2<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value	
P1	Window number	1 ... 3	
P2	Side of exit	´LINKS´	→ Exit to left
		´RECHTS´	→ Exit to right
		´OBEN´	→ Exit to top
		´UNTEN´	→ Exit to bottom
		´EGAL´	→ Exit side does not matter

3.3.7.2 Switch points

3.3.7.2.1 **SCHA** Switch point 1

Input of a new value

With the command SCHA! the first Online-switch point can be defined.
In case of Absolute reference or Block reference of the measurement curve, the switch point refers to the absolute zero point, with an active Trigger reference it refers to the trigger value.
This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>SCHA! P1,P2,P3<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Channel	´X´ → X-channel ´Y´ → Y-channel
P2	Switch point value	Floating-point number without unit
P3	Reference	´ABS´ → Reference: Absolute ´TRIG´ → Reference: Trigger (Only channel X and if Trigger is active)

Readout of the present value

With the command SCHA? the settings for the first switch point can be read out.

Host sends: <Address>sr<STX>SCHA?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Channel	´X´ → X-channel ´Y´ → Y-channel
P2	Switch point value	Floating-point number without unit
P3	Reference	´ABS´ → Reference: Absolute ´TRIG´ → Reference: Trigger (Only channel X and if Trigger is active)

3.3.7.2.2 **SCHB** Switch point 2

Input of a new value

With the command SCHB! the second Online switch point can be defined.
In case of Absolute reference or Block reference of the measurement curve, the switch point refers to the absolute zero point, with an active Trigger reference it refers to the trigger value.
This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>SCHB! P1,P2,P3<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Channel	´X´ → X-channel ´Y´ → Y-channel
P2	Switch point value	Floating-point number without unit
P3	Reference	´ABS´ → Reference: Absolute ´TRIG´ → Reference: Trigger (Only channel X and if Trigger is active)

Readout of present value

With the command SCHB? the settings for the second switch point can be read out.

Host sends: <Address>sr<STX>SCHB?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Channel	´X´ → X-channel ´Y´ → Y-channel
P2	Switch point value	Floating-point number without unit
P3	Reference	´ABS´ → Reference: Absolute ´TRIG´ → Reference: Trigger (Only channel X and if Trigger is active)

3.3.8 During Measurement

3.3.8.1 MSTA Status of Measurement

With the command MSTA? the present Measurement status can be read out.
This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>MSTA?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Status of Measurement	´0´ → Since last reset no measurement was started ´1´ → Present measurement, results have been read out ´2´ → Present measurement, no results have yet been read out

3.3.9 Output of Measurement results

3.3.9.1 Short form

3.3.9.1.1 **MERG** Measurement result

Input of a new value

With the command MERG! The pieces counter and NOK-counter can be set (for backup)

Host sends: <Address>sr<STX>MERG! P1,P2 <ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaing	Value
P1	Pieces counter	0 ... 2^{32}
P2	NOK-counter	0 ... 2^{32}

Readout of present value

With the command MERG? the short form of the evaluation result can be read out. If no measurement is present (pieces counter = 0), the device answers with the respective counter number but gives out a NOK signal. If one input channel is overdriven the evaluation result is NOK (see OVER? command). This input is valid only for the present measurement program.

Host sends: <Address>sr<STX>MERG?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Pieces counter	0 ... 2^{32}
P2	NOK-counter	0 ... 2^{32}
P3	Total result	´NOK´ → Evaluation result: not as per specs. ´OK´ → Evaluation result: in good order ´NOT´ → Evaluation result: NOK as result of passing the trend limiting (with actice clinch point monitoring);

3.3.9.1.2 **OVER** Overrange of channel X/Y

With the command OVER? The overrange status of both channel can be read out. If one of the channel is in overrange the evaluation of the vurve is set NOK.

Host sends: <Address>sr<STX>OVER?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	**Meaning**	**value**
P1	X-channel overrange	0 -> X-channel not in overrange 1 -> X-channel in overrange
P2	Y-channel overrange	0 -> Y-channel not in overrange 1 -> Y-channel in overrange

3.3.9.1.3 **AKRV** General curve data

With the command AKRV? the general curve data can be read out.
There are two ways to read out the curve data: first of all, all general curve data can be requested at once, secondly, the general curve data can be requested separately.

Readout of general curve data

Here the command AKRV? is called without parameter.

Host sends: <Address>sr<STX>AKRV?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,P4,P5,P6,P7,P8,P9,P10,P11,P12<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Smallest force, X-coordinate	Floating-point number with unit
P2	Smallest force, Y-coordinate	Floating-point number with unit
P3	Greatest force, X-coordinate	Floating-point number with unit
P4	Greatest force, Y-coordinate	Floating-point number with unit
P5	Smallest displacement, X-coordinate	Floating-point number with unit
P6	Smallest displacement, Y-coordinate	Floating-point number with unit
P7	Greatest displacement, X-coordinate	Floating-point number with unit
P8	Greatest displacement, Y-coordinate	Floating-point number with unit
P9	Last point, X-coordinate	Floating-point number with unit
P10	Last point, Y-coordinate	Floating-point number with unit
P11	First point, X-coordinate	Floating-point number with unit
P12	First point, Y-coordinate	Floating-point number with unit

Readout of single general curve data

With the command AKRV? the general curve data is read out with request parameters.

Host sends: <Address>sr<STX>AKRV? P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P2,P3<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Desired value	´YMIN´ → Smallest force ´YMAX´ → Greatest force ´XMIN´ → Smallest displacement ´XMAX´ → Greatest displacement ´LAST´ → Last point ´FIRST´ → First point
P2	Requested value, X-coordinate	Floating-point number with unit
P3	Requested value, Y-coordinate	Floating-point number with unit

3.3.9.1.4 MALL? Common curve data

With the command MALL? a summery of all curve and evaluation results can be read out. Single parts can be ead with the commands MERG?, AKRV? and KRVA?

Host sends: <Address>sr<STX>MALL?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,...,P25 <ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	**Meaning**	**value**
P1	Unit X-Axis	Character chain with 4 figures
P2	Unit Y-Axis	Character chain with 4 figures
P3	M X axis (zero point of X values)	float
P4	M Y axis (zero point of Y values)	float
P5	K X axis (gradient of X values)	float
P6	K Y axis (gradient of Y values)	float
P7	Number of measurement values	0 ... 4000
P8	Pieces counter	0 ... 2^{32}
P9	NOK-counter	0 ... 2^{32}
P10	Total result	´NOK´ → Evaluation result: not as per specs. ´OK´ → Evaluation result: in good order ´NOT´ → Evaluation result: NOK as result of passing the trend limiting (with actice clinch point monitoring);
P11	Smallest force, X-coordinate	Floating-point number with unit
P12	Smallest force, Y-coordinate	Floating-point number with unit
P13	Greatest force, X-coordinate	Floating-point number with unit
P14	Greatest force, Y-coordinate	Floating-point number with unit
P15	Smallest displacement, X-coordinate	Floating-point number with unit
P16	Smallest displacement, Y-coordinate	Floating-point number with unit
P17	Greatest displacement, X-coordinate	Floating-point number with unit
P18	Greatest displacement, Y-coordinate	Floating-point number with unit
P19	Last point, X-coordinate	Floating-point number with unit
P20	Last point, Y-coordinate	Floating-point number with unit
P21	First point, X-coordinate	Floating-point number with unit
P22	First point, Y-coordinate	Floating-point number with unit
P23	X-channel overrange	0 -> X-channel not in overrange 1 -> X-channel in overrange
P24	Y-channel overrange	0 -> Y-channel not in overrange 1 -> Y-channel in overrange
P25	Status: max. number of values reached	0 -> Max. number of values not reached 1 -> Max. number of values is reached

3.3.9.2 Window results

3.3.9.2.1 **FERG** Window result

With the command FERG? the results of the single windows can be requested. The data refers to the last measurement. If no measurement has been done since last reset then all active windows are displayed with NOK-result. If no measurement values have been recorded, the NAK is returned and the bit 0x0400 is set in the error status register.

Host sends: <Address>sr<STX>FERG?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Result window 1	´NOK´ → Window was not in specs. ´OK´ → Window was in specs. ´OFF´ → Window was not activated
P2	Result window 2	´NOK´ → Window was not in specs. ´OK´ → Window was in specs. ´OFF´ → Window was not activated
P3	Result window 3	´NOK´ → Window was not in specs. ´OK´ → Window was in specs. ´OFF´ → Window was not activated

3.3.9.2.2 FCNT Window NOK counter

Input of a new value

With the command FCNT! the window NOK counter can be set

Host sends: <Address>sr<STX> FCNT! P1,P2,P3<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	value
P1	Window 1 NOK counter	Counter value
P2	Window 2 NOK counter	Counter value
P2	Window 3 NOK counter	Counter value

Readout of present value

With the command FCNT! the window NOK counter can be read out.

Host sends: <Address>sr<STX>FCNT?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	value
P1	Window 1 NOK counter	Counter value
P2	Window 2 NOK counter	Counter value
P2	Window 3 NOK counter	Counter value

3.3.9.2.3 **FEAU** Window-Entry/-Exit

With the command FEAU? the coordinates for window entry/exit of any one window can be read out for the last measurement. If no measurement has been effected (pieces counter = 0) the device answers with NAK.

Host sends: <Address>sr<STX>FEAU? P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P2,P3,P4,P5<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	**Meaning**	**Value**
P1	Window number	1 ... 3
P2	Window entry, X-coordinate	Floating-point number with unit
P3	Window entry, Y-coordinate	Floating-point number with unit
P4	Window exit, X-coordinate (for block window: block value X)	Floating-point number with unit
P5	Window exit, Y-coordinate (for block window: block value Y)	Floating-point number with unit

3.3.9.2.4 FALL? Read out of all window evaluation values

With the command FALL? the summery of the single commands FERG?, FSTX? and FEAU? Can be read out with one command.

Host sends: <Address>sr<STX>FALL?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,..., P20<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	**Meaning**	**value**
P1	X unit	String
P2	Y unit	String
P3	Reslt window 1	´NOK´ → Window was not in specs. ´OK´ → Window was in specs. ´OFF´ → Window was not activated
P4	Statistics window 1	Floating-point number with unit (%)
P5	Window 1 entry, X-coordinate	Floating-point number with unit
P6	Window 1 entry, Y-coordinate	Floating-point number with unit
P7	Window 1 exit, X-coordinate (for block window: block value X)	Floating-point number with unit
P8	Window 1 exit, Y-coordinate (for block window: block value Y)	Floating-point number with unit
P9	Reslt window 2	´NOK´ → Window was not in specs. ´OK´ → Window was in specs. ´OFF´ → Window was not activated
P10	Statistics window 2	Floating-point number with unit (%)
P11	Window 2 entry, X-coordinate	Floating-point number with unit
P12	Window 2 entry, Y-coordinate	Floating-point number with unit
P13	Window 2 exit, X-coordinate (for block window: block value X)	Floating-point number with unit
P14	Window 2 exit, Y-coordinate (for block window: block value Y)	Floating-point number with unit
P15	Reslt window 3	´NOK´ → Window was not in specs. ´OK´ → Window was in specs. ´OFF´ → Window was not activated
P16	Statistics window 3	Floating-point number with unit (%)
P17	Window 3 entry, X-coordinate	Floating-point number with unit
P18	Window 3 entry, Y-coordinate	Floating-point number with unit
P19	Window 3 exit, X-coordinate (for block window: block value X)	Floating-point number with unit
P20	Window 3 exit, Y-coordinate (for block window: block value Y)	Floating-point number with unit

3.3.9.3 Statistics

3.3.9.3.1 **FSTX** Window statistics

With the command FSTX? the window statistics can be read out. The window statistics informs how strong each window is engaged in a total NOK-result. All data is given in percent of the NOK-counter.
For Window type = OFF and no measurement since last reset 0% is transferred.

Host sends: <Address>sr<STX>FSTX?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	**Meaning**	**Value**
P1	Statistics window 1	Floating-point number with unit (%)
P2	Statistics window 2	Floating-point number with unit (%)
P3	Statistics window 3	Floating-point number with unit (%)

3.3.9.4 Measurement curve

3.3.9.4.1 **KRVA** General Measurement curve data

With the command KRVA? all general information that are necessary for reading and interpreting the curve can be read out. With the help of the M- and K- values, the numbers transferred with the curve can be converted into floating-point values.

Host sends: <Address>sr<STX>KRVA?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,P4,P5,P6,P7,P8 <ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	**Meaning**	**Value**
P1	Unit X-Axis	Character chain with 4 figures
P2	Unit Y-Axis	Character chain with 4 figures
P3	M X-Axis	Floating-point number
P4	M Y-Axis	Floating-point number
P5	K X-Axis	Floating-point number
P6	K Y-Axis	Floating-point number
P7	Number of measurement values	0 ... 4000
P8	Status: Max. number of values reached	0 → max. numeber not reached 1 → max. number is reached

3.3.9.4.2 KURV Measurement curve

With the command **KURV?** the complete Measurement curve can be read out.
The Measurement data are transferred in hexadecimal data format and without 0-termination (for time reasons).
Parameter separator between the discrete values is the comma.
10 value pairs are always transferred, then an LF follows.
After confirmation of receipt of the host with ACK the next 10 value pairs are being transferred, etc.
The readout repeats again and again until no more measurement values are available. The transfer always consists of a group of 10 value pairs. If, at the end, there are less than 10 value pairs left, the transfer of the last value pair is repeated until the block of 10 is filled up.
The measurement value arise from e.g.: ((Yn-MY)*KY) [unit Y-Axis]. This information need to be read out first of all with the KRVA command.

Host sends: <Address>sr<STX>KURV?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>X_1,Y_1,X_2,Y_2,X_3,Y_3,X_4,Y_4,X_5,Y_5, X_6,Y_6,X_7,Y_7,X_8,Y_8,X_9,Y_9,X_{10},Y_{10}, LF<ETX>[<BCC>]
Host sends: <ACK>

DIGIFORCE answers: <STX>X_{11},Y_{11}, ... ,X_{20},Y_{20}, LF<ETX>[<BCC>]
Host sends: <ACK>

.....

DIGIFORCE answers: <EOT>

Whereas (X1|Y1) are the coordinates for the first point of the curve.

With the command KURV! data in the output buffer, arosen from an interrupted KURV? command, could be erased.

3.3.10 Envelope

3.3.10.1 Evaluation result

3.3.10.1.1 HERG Envelope evaluation result

With the command HERG? The evaluation result of the envelope can be read out.

Host sends: <Address>sr<STX>HERG?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1 <ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Envelope evaluation result	´NOK´ → envelope was violated ´OK´ → data within envelope curve band ´OFF´ → envelope not active

3.3.10.1.2 HCNT Envelope NOK counter

Input of a new value

With the command HCNT! the NOK counter of the envelope evaluation can be set

Host sends: <Address>sr<STX>HCNT! P1 <ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Envelope NOK counter	Counter value

Readout of present value

With the command HCNT! the NOK counter of the envelope evaluation can be read out. The command is only valid if the envelope is active.

Host sends: <Address>sr<STX>HCNT?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Envelope NOK counter	Counter value

3.3.10.1.3 TCNT Ternd limit violated NOK counter

Input of a new value

With the command TCNT! the NOK counter of the trend limit can be set.

Host sends: <Address>sr<STX>TCNT! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Trend limit violated NOK counter	Counter value

Readout of present value

With the command TCNT! the NOK counter of the trend limit can be read out. The command is only valid if the envelope is active.

Host sends: <Address>sr<STX>TCNT?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1 <ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Trend limit violated NOK counter	Counter value

3.3.10.1.4 HSTX Envelope NOK statistic

With the HSTX? command, the envelope NOK statistic can be queried. This relates to the NOK fraction (in percent) of the envelope relative to the total NOK evaluations.

Host sends: <Address>sr<STX>HSTX?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	NOK percentage of the envelope.	Positive integer with % sign

3.3.10.1.5 HNIO Envelope NOK X-/Y- Coordinate

With the HNIO? command, the X/Y coordinate can be retrieved, at which, in the NOK case, the envelope has been crossed. The command is only valid if the envelope is activated.

Host sends: <Address>sr<STX>HNIO?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	X coordinate of the point where the envelope has been crossed.	Floating-point number with X unit
P2	Y coordinate of the point where the envelope has been crossed.	Floating-point number with Y unit

3.3.10.1.6 TGRZ Trend limit reached

With the TGRZ? command, the result of trend tracking can be retrieved. Has the trend limit been reached and where? Through the envelope, the online window or the block window.
The command in only allowed if trend tracking is activated. If the online or the block window is off, then a 0 is returned.

Host sends: <Address>sr<STX>TGRZ?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Envelope trend limit reached	0 -> Trend limit of the envelope has not been reached 1 -> Trend limit of the envelope has not been reached
P2	Online window trend limit reached	0 -> Trend limit is not reached or online window is off. 1 -> Trend limit of the online window has not been reached
P3	Block window trend limit reached	0 -> Trend limit has not been reached or block window is off. 1 -> Trend limit of the block window has been reached

3.3.10.1.7 TSTX Trend limit reaches NOK statistic

With the TSTX? command, the 'trend limit reached' NOK statistic can be queried. This relates to the NOK fraction (in percent) of the 'trend limit reached' relative to the total NOK evaluations. The trend limit reached statistic only functions if clinch function is activated and trend is switched on.

Host sends: <Address>sr<STX>TSTX?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	NOK percentage if trend limit reached.	Positive integer with % sign

3.3.10.1.8 TNIO trend limit reached NOK X /Y coordinate

With the TNIO? command, the X/Y coordinate can be retrieved, at which, in the NOK case, the trend limit has been crossed. The command is only valid if the envelope is activated, clinch function is activated and trend switched on.

Host sends: <Address>sr<STX>TNIO?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	X coordinate of the point where the trend limit has been crossed.	Floating-point number with X unit
P2	Y coordinate of the point where the trend limit has been crossed.	Floating-point number with Y unit

3.3.10.2 Configuration of the envelope

3.3.10.2.1 HKRV Envelope switch ON / OFF

Input of a new value

With the HKRV! command, the envelope can be switched on or off.

Host sends: <Address>sr<STX>HKRV! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Switch envelope on/off	0 -> Envelope is switched off 1 -> Envelope is switched on

Readout of present value

With the HKRV? command, it can be queried whether envelope is switched on or off.

Host sends: <Address>sr<STX>HKRV?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Switch envelope on/off	0 -> Envelope is switched off 1 -> Envelope is switched on

3.3.10.2.2 HDEY Envelope delta Y value

Input of a new value

With the HDEY! command, the delta Y value of the envelope can be set.

Host sends: <Address>sr<STX>HDEY! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Envelope delta Y value	Floating-point number without unit

Readout of present value

With the HDEY? command, the delta Y value of the envelope can be queried.

Host sends: <Address>sr<STX>HDEY?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Current delta Y value of the envelope	Floating-point number with Y unit

3.3.10.2.3 HXGR Envelope X limits

Input of a new value

With the HXGR! command, Xmin and Xmax values of the envelope can be set.

Host sends: <Address>sr<STX>HXGR! P1,P2<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Xmin limit of the envelope	Floating-point number without unit
P2	Xmax limit of the envelope	Floating-point number without unit

Query value

With the HXGR? command, the delta Y value of the envelope can be queried.

Host sends: <Address>sr<STX>HXGR?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,P4<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Xmin limit of the envelope	Floating-point value with X unit
P2	Xmax limit of the envelope	Floating-point value with X unit
P3	Permitted minimum Xmin limit of the envelope	Floating-point value with X unit
P4	Permitted maximum Xmax limit of the envelope	Floating-point value with X unit

3.3.10.2.4 TREN switch trend on/off

Input of a new value

With the TREN! command, the envelope can be switched on or off.

Host sends: <Address>sr<STX>TREN! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Switch trend on/off	0 -> Trend env. c. off, trend online window off, trend block window off 1 -> Trend env. c. on, trend online window on, trend block window on 2 -> Trend env. c. on, trend online window on, trend block window off 3 -> Trend env. c. on, trend online window off, trend block window on 4 -> Trend env. c. on, trend online window off, trend block window off

Readout of present value

With the TREN? command, it can be queried whether trend is switched on or off.

Host sends: <Address>sr<STX>TREN?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Switch trend on/off	0 -> Trend env. c. off, trend online window off, trend block window off 1 -> Trend env. c. on, trend online window on, trend block window on 2 -> Trend env. c. on, trend online window on, trend block window off 3 -> Trend env. c. on, trend online window off, trend block window on 4 -> Trend env. c. on, trend online window off, trend block window off

3.3.10.2.5 TDEY trend delta Y value

Input of a new value

With the TDEY! command, the delta Y value of the trend can be set.

Host sends: <Address>sr<STX>TDEY! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Trend delta Y value	Floating-point number without unit

Readout of present value

With the TDEY? command, the delta Y value of the trend can be queried.

Host sends: <Address>sr<STX>TDEY?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Current delta Y value of the trend	Floating-point number with Y unit

3.3.10.2.6 TGEW trend weighting

Input of a new value

With the TGEW! command, the trend weighting can be set.

Host sends: <Address>sr<STX>TGEW! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Weighting for trend tracking	Valid range: 1 to 20

Readout of present value

With the TGEW? command, the trend weighting can be queried.

Host sends: <Address>sr<STX>TGEW?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Weighting for trend tracking	Valid range: 1 to 20

3.3.10.2.7 HSTA envelope available or not

With the HSTA? command, it can be queried whether an envelope is available or not.

Host sends: <Address>sr<STX>HSTA?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Envelopes available status	0 -> there is no envelope 1 -> an envelope is available

3.3.10.2.8 RESH reset envelope

With the RESH! command, the envelope can be reset to its teach-in value when trend tracking is switched on. The command is only valid if the clinch function is switched on

Host sends: <Address>sr<STX>RESH!<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

No parameter, no query form

3.3.10.3 Envelope setup

3.3.10.3.1 HEIN Setup envelope (new / modify)

Input of a new value

With the HEIN! command, the operating mode for setting up the envelope can be set (new / modify).

Host sends: <Address>sr<STX>HEIN! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Type of envelope setup	NEW -> new envelope is set up MODIFY -> envelope is modified

Readout of present value

With the HEIN? command, the operating mode for setting up the envelope can be queried (new / modify).

Host sends: <Address>sr<STX>HEIN?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Type of envelope setup	NEW -> new envelope is set up MODIFY -> envelope is modified

3.3.10.3.2 HEBE Type of confirmation when setting up the envelope (always/once)

Input of a new value

With the HEBE! command, the confirmation when setting up the envelope can be set.
(once / always)

Host sends: <Address>sr<STX>HEBE! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Confirmation type when setting up the envelope.	ALWAYS -> Confirm after each curve ONCE -> confirm once at end

Readout of present value

With the HEBE? command, the confirmation when setting up the envelope can be queried.
(once / always)

Host sends: <Address>sr<STX>HEBE?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Confirmation type when setting up the envelope.	ALWAYS -> Confirm after each curve ONCE -> confirm once at end

3.3.10.3.3 HARE number of reference curves when setting up the envelope

Input of a new value

With the HARE! command, the number of reference curves used when setting up the envelope can be set.

Host sends: <Address>sr<STX>HARE! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Number of reference curves when setting up the envelope	Valid range: 1 to 99

Readout of present value

With the HARE? command, the number of reference curves used when setting up the envelope can be queried.

Host sends: <Address>sr<STX>HARE?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Number of reference curves when setting up the envelope	Valid range: 1 to 99

3.3.10.3.4 RANZ number or reference curve points

Input of a new value

With the RANZ! command, the number of reference curve points can be set.

Host sends: <Address>sr<STX>RANZ! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Number of reference curve points	Valid range: 1 to 4000 1 means no reference curve

Readout of present value

With the RANZ? command, the number of reference curve points can be queried.

Host sends: <Address>sr<STX>RANZ?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Number of reference curve points	Valid range: 2 to 4000

3.3.10.3.5 RSTA reference curve status

Readout of present value

With the RSTA? command, the different states of the reference curve can be queried.

Host sends: <Address>sr<STX>RSTA?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,P4,P5,P6,P7,P8<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Status of Y values of the reference curve for creating the envelope	0 -> no reference curve 1 -> values already retrieved 2 -> new values
P2	Status of X values of the reference curve	0 -> no reference curve 1 -> values already retrieved 2 -> new values
P3	Status of Ymin values of the reference curve	0 -> no reference curve 1 -> values already retrieved 2 -> new values
P4	Status of Ymax values of the reference curve	0 -> no reference curve 1 -> values already retrieved 2 -> new values
P5	Status of Y values of the reference curve for creating the trend curve	0 -> no reference curve 1 -> values already retrieved 2 -> new values
P6	Status of Y cumulative values of the reference curve for creating the trend curve	0 -> no reference curve 1 -> values already retrieved 2 -> new values
P7	Reserved	
P8	Reserved	

3.3.10.3.6 RYWE Y mean values of the reference curve (tracked values)

If trend is activated, this relates to the tracked values.

RYWE! Values from host to unit

With this command, the Y mean values (normalized values: values corrected by the zero point) of the reference curve can be transferred from the host to the unit.
The Y mean values are transferred in blocks of up to 20 values (parameters 2 to 21). Parameter 1 is the block number. With a maximum of 4000 values there is thus a maximum of 200 blocks. The block number therefore gives the number of blocks, which each contain 20 values. Full 20 value containing blocks must always be transferred. However, the unit only considers values up until the number of values has been reached. In the last block, any of the 20 values, which remain unfilled can be padded with dummies. The number of values must therefore be transferred prior to this command using the command RANZ!.

Host sends: <Address>sr<STX>RYWE! P1,P2,....,P21<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Block number	Integer value: 1 to 200
P2-P21	Y mean values of the reference curve	Hex values without 0 termination

RYWE? Values from unit to host

With this command, the Y mean values (normalized values: values corrected by the zero point) of the reference curve can be transferred from the unit to the host.
40 values are always transferred at the same time. After polling, these first 40 values are transferred.
Once the host has confirmed receipt with <ACK> , the next 40 values are transferred. This continues until all values have been transferred. When all values have been transferred, the unit answers with <EOT> following the <ACK>. Full 40 value containing blocks are always transferred. The last block of 40 is padded with dummies. There may be up to 4000 values. Thus there may be up to 100 blocks. The host must previously have retrieved the number of values using the RANZ? command.

Host sends: <Address>sr<STX>RYWE?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,....P40<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <STX>P41,P42,P43,....P80<ETX>[<BCC>]
Host sends: <ACK>
.
.
.
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1-P4000	Y mean values of the reference curve	Hex values without 0 termination

3.3.10.3.7 RYSU Y cumulative values of the reference curve

RYSU! Values from host to unit

With this command, the Y cumulative values (normalized values: values corrected by the zero point) of the reference curve can be transferred from the host to the unit.
The Y cumulative values are transferred in blocks of up to 20 values (parameters 2 to 21). Parameter 1 is the block number. With a maximum of 4000 values there is thus a maximum of 200 blocks. The block number therefore gives the number of blocks, which each contain 20 values. Full 20 value containing blocks must always be transferred. However, the unit only considers values up until the number of values has been reached. In the last block, any of the 20 values, which remain unfilled can be padded with dummies. The number of values must therefore be transferred prior to this command using the command RANZ!.

Host sends: <Address>sr<STX>RYSU! P1,P2,....,P21<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Block number	Integer value: 1 to 200
P2-P21	Y cumulative values of the reference curve	Hex values without 0 termination

RYSU? Values from unit to host

With this command, the average Y values (normalized values: values corrected by the zero point) of the reference curve can be transferred from the unit to the host.
20 values are always transferred at the same time. After polling, these first 20 values are transferred.
Once the host has confirmed receipt with <ACK> , the next 20 values are transferred. This continues until all values have been transferred. When all values have been transferred, the unit answers with <EOT> following the <ACK>. Full 20 value blocks are always transferred. The last block of 20 is padded with dummies.
There may be up to 4000 values. Thus there may be up to 200 blocks. The host must previously have retrieved the number of values using the RANZ? command.

Host sends: <Address>sr<STX>RYSU?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,....P20<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <STX>P21,P22,P23,....P40<ETX>[<BCC>]
Host sends: <ACK>
.
.
.
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1-P4000	Y cumulative values of the reference curve	Hex values without 0 termination

3.3.10.3.8 RXWE X values of the reference curve

RXWE! Values from host to unit

With this command, the X values (***normalized values***: ***values corrected by the zero point***) of the reference curve can be transferred from the host to the unit.
The X values are transferred in blocks of up to 20 values (parameters 2 to 21). Parameter 1 is the block number. With a maximum of 4000 values there is thus a maximum of 200 blocks. The block number therefore gives the number of blocks, which each contain 20 values. Full 20 value containing blocks must always be transferred. However, the unit only considers values up until the number of values has been reached. In the last block, any of the 20 values, which remain unfilled can be padded with dummies. **The number of values must therefore be transferred prior to this command using the command RANZ!.**

Host sends: <Address>sr<STX>RXWE! P1,P2,....,P21<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Block number	Integer value: 1 to 200
P2-P21	X values of the reference curve	Hex values without 0 termination

RXWE? Values from unit to host

With this command, the X values (***normalized values***: ***values corrected by the zero point***) of the reference curve can be transferred from the unit to the host.
40 values are always transferred at the same time. After polling, these first 40 values are transferred.
Once the host has confirmed receipt with <ACK> , the next 40 values are transferred. This continues until all values have been transferred. When all values have been transferred, the unit answers with <EOT> following the <ACK>. Full 40 value containing blocks are always transferred. The last block of 40 is padded with dummies. There may be up to 4000 values. Thus there may be up to 100 blocks. **The host must previously have retrieved the number of values using the RANZ? command.**

Host sends: <Address>sr<STX>RXWE?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,....P40<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <STX>P41,P42,P43,....P80<ETX>[<BCC>]
Host sends: <ACK>
.
.
.
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1-P4000	X values of the reference curve	Hex values without 0 termination

3.3.10.3.9 RYMI delta Y minimum values of the reference curve

RYMI! Values from host to unit

With this command, the delta Y minimum values of the reference curve are transferred from the host to the unit. The delta Y minimum values are transferred in blocks of up to 20 values (parameters 2 to 21). Parameter 1 is the block number. With a maximum of 4000 values there is thus a maximum of 200 blocks. The block number therefore gives the number of blocks, which each contain 20 values. Full 20 value containing blocks must always be transferred. However, the unit only considers values up until the number of values has been reached. In the last block, any of the 20 values, which remain unfilled can be padded with dummies. **The number of values must therefore be transferred prior to this command using the command RANZ!.**

Host sends: <Address>sr<STX>RYMI! P1,P2,....,P21<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Block number	Integer value: 1 to 200
P2-P21	Delta Y minimum values of the reference curve	Hex values without 0 termination

RYMI? Values from unit to host

With this command, the delta Y minimum values of the reference curve are transferred from the unit to the host. 40 values are always transferred at the same time. After polling, these first 40 values are transferred.
Once the host has confirmed receipt with <ACK> , the next 40 values are transferred. This continues until all values have been transferred. When all values have been transferred, the unit answers with <EOT> following the <ACK>. Full 40 value containing blocks are always transferred. The last block of 40 is padded with dummies. There may be up to 4000 values. Thus there may be up to 100 blocks. **The host must previously have retrieved the number of values using the RANZ? command.**

Host sends: <Address>sr<STX>RYMI?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,....P40<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <STX>P41,P42,P43,....P80<ETX>[<BCC>]
Host sends: <ACK>
.
.
.
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1-P4000	Delta Y minimum values of the reference curve	Hex values without 0 termination

3.3.10.3.10 RYMA delta Y maximum value of the reference curve

RYMA! Values from host to unit

With this command, the delta Y maximum values of the reference curve are transferred from the host to the unit. The delta Y maximum values are transferred in blocks of up to 20 values (parameters 2 to 21). Parameter 1 is the block number. With a maximum of 4000 values there is thus a maximum of 200 blocks. The block number therefore gives the number of blocks, which each contain 20 values. Full 20 value containing blocks must always be transferred. However, the unit only considers values up until the number of values has been reached. In the last block, any of the 20 values, which remain unfilled can be padded with dummies. **The number of values must therefore be transferred prior to this command using the command RANZ!.**

Host sends: <Address>sr<STX>RYMA! P1,P2,....,P21<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Block number	Integer value: 1 to 200
P2-P21	Delta Y maximum values of the reference curve	Hex values without 0 termination

RYMA? Values from unit to host

With this command, the delta Y maximum values of the reference curve are transferred from the unit to the host. 40 values are always transferred at the same time. After polling, these first 40 values are transferred.
Once the host has confirmed receipt with <ACK> , the next 40 values are transferred. This continues until all values have been transferred. When all values have been transferred, the unit answers with <EOT> following the <ACK>. Full 40 value containing blocks are always transferred. The last block of 40 is padded with dummies. There may be up to 4000 values. Thus there may be up to 100 blocks. **The host must previously have retrieved the number of values using the RANZ? command.**

Host sends: <Address>sr<STX>RYMA?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,....P40<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <STX>P41,P42,P43,....P80<ETX>[<BCC>]
Host sends: <ACK>
.
.
.
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1-P4000	Delta Y maximum values of the reference curve	Hex values without 0 termination

3.3.10.3.11 RYTR Y mean values of the reference curve (non-tracked values)

If trend is activated, this relates to the non-tracked values.

RYTR! Values from host to unit

With this command, the Y mean values (***normalized value: values corrected by the zero point***) of the reference curve can be transferred from the host to the unit.
The Y mean values are transferred in blocks of up to 20 values (parameters 2 to 21). Parameter 1 is the block number. With a maximum of 4000 values there is thus a maximum of 200 blocks. The block number therefore gives the number of blocks, which each contain 20 values. Full 20 value containing blocks must always be transferred. However, the unit only considers values up until the number of values has been reached. In the last block, any of the 20 values, which remain unfilled can be padded with dummies. **The number of values must therefore be transferred prior to this command using the command RANZ!.**

Host sends: <Address>sr<STX>RYTR! P1,P2,....,P21<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Block number	Integer value: 1 to 200
P2-P21	Y mean values of the reference curve	Hex values without 0 termination

RYTR? Values from unit to host

With this command, the Y mean values (***normalized value: values corrected by the zero point***) of the reference curve can be transferred from the unit to the host.
40 values are always transferred at the same time. After polling, these first 40 values are transferred.
Once the host has confirmed receipt with <ACK> , the next 40 values are transferred. This continues until all values have been transferred. When all values have been transferred, the unit answers with <EOT> following the <ACK>. Full 40 value containing blocks are always transferred. The last block of 40 is padded with dummies. There may be up to 4000 values. Thus there may be up to 100 blocks. **The host must previously have retrieved the number of values using the RANZ? command.**

Host sends: <Address>sr<STX>RYTR?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,....P40<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <STX>P41,P42,P43,....P80<ETX>[<BCC>]
Host sends: <ACK>
.
.
.
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1-P4000	Y mean values of the reference curve	Hex values without 0 termination

3.3.10.3.12 RACC calculate envelope

RACC!

After the reference curve has been completely received from the host, the RACC! command can be used to start calculation of the envelope.
Completely mean values the following commands must previously have been implemented:
(RXWE! or KXWE!, RYMI! or KYMI!, RYMA! or KYMA!, RYTR! or KYTR!, RANZ!)

Host sends:	<Address>sr<STX>RACC!<ETX>[<BCC>]
DIGIFORCE answers:	<ACK>
Host sends:	<EOT>

There are no parameters or question form.

3.3.10.4 Selecting the display from the setup menu

3.3.10.4.1 AEHK display selection in the envelopes setup menu

Input of a new value

With the AEHK! command, the display selection when setting up the envelope, can be set.
(Current curve / reference curve / trend curve)
If trend is not activated, then display of the trend curve is not possible.
If envelope is not active, then display of reference and trend curves is not possible.

Host sends: <Address>sr<STX>AEHK! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	What is displayed in the envelopes setup menu	MESS -> display of the current measurement curve REF -> display of the reference curve TREND -> display of the trend curve (with reference curve)

Readout of present value

With the AEHK? command, the display selection when setting up the envelope can be queried.
(Current curve / reference curve / trend curve)

Host sends: <Address>sr<STX>AEHK?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	What is displayed in the envelopes setup menu	MESS -> display of the current measurement curve REF -> display of the reference curve TREND -> display of the trend curve (with reference curve)

3.3.10.4.2 AEHF switch on/off window display in the envelopes setup menu

Input of a new value

With the AEHF! command, window display in the envelopes setup menu can be switched on or off.
If trend curve is switched on, then window display cannot be switched on.

Host sends: <Address>sr<STX>AEHF! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Window display in the envelopes setup menu	0 -> windows are not displayed 1 -> windows are displayed

Readout of present value

With the AEHF? command, it can be queried whether window display in the envelopes setup menu is switched on or off.

Host sends: <Address>sr<STX>AEHF?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Window display in the envelopes setup menu	0 -> windows are not displayed 1 -> windows are displayed

3.3.10.4.3 AEHK display selection in the window setup menu

Input of a new value

With the AEFK! command, the display selection when setting up the window can be set.
If trend is not activated, then display of the trend limits is not possible.
If envelope is not active, then display of reference curve and trend limits is not possible.
(Current curve / reference curve / trend limits)

Host sends: <Address>sr<STX>AEFK! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	What is displayed in the window setup menu	MESS -> display of the current measurement curve REF -> display of the reference curve TREND -> display of the trend limits (with window)

Readout of present value

With the AEFK? command, the display selection when setting up the window can be queried.
(Current curve / reference curve / trend curve)

Host sends: <Address>sr<STX>AEFK?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	What is displayed in the window setup menu	MESS -> display of the current measurement curve REF -> display of the reference curve TREND -> display of the trend limits (with window)

3.3.10.4.4 AEHF switch on/off envelopes display in the window setup menu

Input of a new value

With the AEFH! command, envelopes display in the window setup menu can be switched on or off.
(If envelope is not activated, then envelope display cannot be switched on.

Host sends: <Address>sr<STX>AEFH! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Envelope display in window setup menu	0 -> envelope is not displayed 1 -> envelope is displayed

Readout of present value

With the AEFH? command, it can be queried whether envelope display in the window setup menu is switched on or off.

Host sends: <Address>sr<STX>AEFH?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Envelope display in window setup menu	0 -> envelope is not displayed 1 -> envelope is displayed

3.3.11 Remaining blade thickness (RBD) / Plate pack thickness (BPD)

Commands on this optional function are on request.

Pleae call: burster präzisionsmesstechnik gmbh & co kg
Gernsbach / Germany
++49 – 7224 - 6450

3.4 Commands optimized in respect of transfer time

3.4.1 Transfer of the measurement curve

3.4.1.1 MRED Command

Input of a new value

With the MRED! command, the reducing factor can be set with which the curve is transferred using the KURX and KURY commands. The values 1 to 20 can be set. 1 means no reduction. Example: a reducing factor of 4 means that when the KURX and KURY commands are applied, only every 4th value is transferred. The first and last values of the measurement curve are always transferred independent of the set reducing factor.

Host sends: <Address>sr<STX>MRED! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Reducing factor	Integer value between 1 and 20

Readout of present value

With the MRED? command, the current reducing value can be queried.
Description - see over for the ! form of the command.

Host sends: <Address>sr<STX>MRED?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Reducing factor	Integer value between 1 and 20

3.4.1.2 KURX optimized transfer of the X values of the measurement curve

With the **KURX**? the X values of the measurement curve can be read in a time-optimized manner.
The measurements are transferred in hexadecimal format without 0-termination (to save time).

The first value is transferred as an absolute value, all other values are transferred as a difference from the preceding value. If more that two sequential difference values are the same, then these are no transferred individually,
but rather in the following form: M<factor>*<difference value>
M is the marker that indicates that this is not a single difference value.
It is followed by a factor that indicates the number of sequential difference values that are the same.
The * character separates factor and difference value.
It is followed by the difference value.

The parameter separator between the individual values is the comma.
20 values are always transferred, then a LF is transferred.
After host acknowledgement with ACK, the next 20 value pairs are transferred and so on.
Querying continues until there are no more available measurements.
If the last block has less than 20 difference values, then transfer is terminated early.
If there are no more available measurements, then an EOT is sent in response to the ACK acknowledgement.

The host must calculate the absolute values from the difference values (with the exception of the 1st value).
Absolute value[n] = absolute value[n-1] + difference value[n]

The transferred difference values and the absolute values calculated from them are integer values (16 bit). These values are used with the zero point and gradient values, which are retrieved using the KRVA command, to calculate the floating-point values in their respective units.

Floating-point value = (absolute value - zero point) * gradient

Host sends: <Address>sr<STX>KURX? (P1)<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>abs_1, dif_2, dif_3, dif_4, dif_5, dif_6, dif_7, , dif_8,...., dif_{20} LF<ETX>[<BCC>]
Host sends: <ACK>

DIGIFORCE answers: <STX> dif_{21},..... , dif_{40} LF<ETX>[<BCC>]
Host sends: <ACK>

.....

DIGIFORCE answers: <EOT>

The command KURX!, can be used to delete any data still in the output buffer if the KURX? command has been interrupted.

Parameter P1 is optional: no parameter means the curve will not be reduced prior to transfer.
Parameter = 0 means the curve will not be reduced prior to transfer.
Parameter = 1 means the curve is reduced prior to transfer corresponding to the reducing factor (1 to 20) which can be set using the MRED command.
Parameter = 2 means the curve is transferred with minus optimization
Parameter = 3 means the curve is reduced and transferred with the minus optimization

Example: Reducing factor = 4 means only every 4th value is transferred, although the first and last values of the measurement curve are always transferred.

3.4.1.3 KURY optimized transfer of the Y values of the measurement curve

With the **KURY?** command the Y values of the measurement curve can be read in a time-optimized manner.
The measurements are transferred in hexadecimal format without 0-termination (to save time).

The first value is transferred as an absolute value, all other values are transferred as a difference from the preceding value. If more that two sequential difference values are the same, then these are no transferred individually,
but rather in the following form: M<factor>*<difference value>
M is the marker that indicates that this is not a single difference value.
It is followed by a factor that indicates the number of sequential difference values that are the same.
The * character separates factor and difference value.
It is followed by the difference value.

The parameter separator between the individual values is the comma.
20 values are always transferred, then a LF is transferred.
After host acknowledgement with ACK, the next 20 value pairs are transferred and so on.
Querying continues until there are no more available measurements.
If the last block has less than 20 difference values, then transfer is terminated early.
If there are no more available measurements, then an EOT is sent in response to the ACK acknowledgement.

The host must calculate the absolute values from the difference values (with the exception of the 1st value).
Absolute value[n] = absolute value[n-1] + difference value[n]

The transferred difference values and the absolute values calculated from them are integer values (16 bit). These values are used with the zero point and gradient values, which are retrieved using the KRVA command, to calculate the floating-point values in their respective units.

Floating-point value = (absolute value - zero point) * gradient

Host sends: <Address>sr<STX>KURY? (P1)<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>abs_1, dif_2, dif_3, dif_4, dif_5, dif_6, dif_7, , dif_8,...., dif_{20} LF<ETX>[<BCC>]
Host sends: <ACK>

DIGIFORCE answers: <STX> dif_{21},..... , dif_{40} LF<ETX>[<BCC>]
Host sends: <ACK>

.....

DIGIFORCE answers: <EOT>

The command KURY!, can be used to delete any data still in the output buffer if the KURY? command has been interrupted.

Parameter P1 is optional: no parameter means the curve will not be reduced prior to transfer.
Parameter = 0 means the curve will not be reduced prior to transfer.
Parameter = 1 means the curve is reduced prior to transfer corresponding to the reducing factor (1 to 20) which can be set using the MRED command.
Parameter = 2 means the curve is transferred with minus optimization
Parameter = 3 means the curve is reduced and transferred with the minus optimization

Example: Reducing factor = 4 means only every 4th value is transferred, although the first and last values of the measurement curve are always transferred.

3.4.2 Envelope

3.4.2.1 KXWE optimized transfer of the X values of the reference curve

KXWE! Transfer values from host to unit in an optimized manner

With this command, the X values (***normalized integer values***: ***only corrected by the zero point, without taking into account the gradient***) of the reference curve can be transferred from the host to the unit in a time optimized manner.
The values are transferred in hexadecimal format without 0-termination (to save time).

The first value is transferred as an absolute value, all other values are transferred as a difference from the preceding value.

The difference values are transferred in blocks of up to 20 values (parameters 2 to 21). Parameter 1 is the block number. With a maximum of 4000 values there is thus a maximum of 200 blocks. The block number therefore gives the number of blocks, which each contain 20 values. Full 20 value containing blocks must always be transferred. However, the unit only considers values up until the number of values has been reached. In the last block, any of the 20 values, which remain unfilled can be padded with dummies. **The number of values must therefore be transferred prior to this command using the command RANZ!.**

Host sends: <Address>sr<STX>KXWE! P1,P2,....,P21<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Block number	Integer value: 1 to 200
P2-P21	X difference values of the reference curve	Hex values without 0 termination

KXWE? Transfer values from unit to host in an optimized manner

With this command, the X values (***normalized values***: ***only corrected by the zero point, without taking into account the gradient***) of the reference curve can be transferred from the unit to the host in a time-optimized manner.

The first value is transferred as an absolute value, all other values are transferred as a difference from the preceding value. If more that two sequential difference values are the same, then these are no transferred individually,
but rather in the following form: M<factor>*<difference value>
M is the marker that indicates that this is not a single difference value.
It is followed by a factor that indicates the number of sequential difference values that are the same.
The * character separates factor and difference value.
It is followed by the difference value.

The parameter separator between the individual values is the comma.
20 values are always transferred, then a LF is transferred.
After host acknowledgement with ACK, the next 20 value pairs are transferred and so on.
Querying continues until there are no more available measurements.
If the last block has less than 20 difference values, then transfer is terminated early.
If there are no more available values, then an EOT is sent in response to the ACK acknowledgement.
There may be up to 4000 values. Thus there may be up to 200 blocks. The host must previously have retrieved the number of values using the RANZ? command.

The host must calculate the absolute values from the difference values (with the exception of the 1st value).
Absolute value[n] = absolute value[n-1] + difference value[n]

The transferred difference values and the absolute values calculated from them are normalized integer values (16 bit). These values are used with the gradient values, which are retrieved using the KRVA command, to calculate the floating-point values in their respective units.

Floating-point value = Normalized absolute value * gradient

Host sends: <Address>sr<STX>KXWE?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,....P2+0<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <STX>P21,P22,P23,....P40<ETX>[<BCC>]
Host sends: <ACK>
.
.
.
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1-P4000	Normalized X difference values of the reference curve	Hex values without 0 termination

3.4.2.2 KYWE optimized transfer of the Y mean values of the reference curve (tracked values)

If trend is activated, this relates to the tracked values.

KYWE! Transfer values from host to unit in an optimized manner

With this command, the Y mean values (***normalized integer values: values corrected by the zero point, without taking into account the gradient***) of the reference curve can be transferred from the host to the unit.
The values are transferred in hexadecimal format without 0-termination (to save time).

The first value is transferred as an absolute value, all other values are transferred as a difference from the preceding value.

The difference values are transferred in blocks of up to 20 values (parameters 2 to 21). Parameter 1 is the block number. With a maximum of 4000 values there is thus a maximum of 200 blocks. The block number therefore gives the number of blocks, which each contain 20 values. Full 20 value containing blocks must always be transferred. However, the unit only considers values up until the number of values has been reached. In the last block, any of the 20 values, which remain unfilled can be padded with dummies. **The number of values must therefore be transferred prior to this command using the command RANZ!.**

Host sends: <Address>sr<STX>KYWE! P1,P2,....,P21<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Block number	Integer value: 1 to 200
P2-P21	Y difference values of the reference curve	Hex values without 0 termination

KYWE? Transfer values from unit to host in an optimized manner

With this command, the Y mean values (***normalized integer values: values corrected by the zero point, without taking into account the gradient***) of the reference curve can be transferred from the unit to the host.

The first value is transferred as an absolute value, all other values are transferred as a difference from the preceding value. If more that two sequential difference values are the same, then these are no transferred individually,
but rather in the following form: M<factor>*<difference value>
M is the marker that indicates that this is not a single difference value.
It is followed by a factor that indicates the number of sequential difference values that are the same.
The * character separates factor and difference value.
It is followed by the difference value.

The parameter separator between the individual values is the comma.
20 values are always transferred, then a LF is transferred.
After host acknowledgement with ACK, the next 20 value pairs are transferred and so on.
Querying continues until there are no more available measurements.
If the last block has less than 20 difference values, then transfer is terminated early.
If there are no more available values, then an EOT is sent in response to the ACK acknowledgement.
There may be up to 4000 values. Thus there may be up to 200 blocks. The host must previously have retrieved the number of values using the RANZ? command.

The host must calculate the absolute values from the difference values (with the exception of the 1st value).
Absolute value[n] = absolute value[n-1] + difference value[n]

The transferred difference values and the absolute values calculated from them are normalized integer values (16 bit). These values are used with the gradient values, which are retrieved using the KRVA command, to calculate the floating-point values in their respective units.

Floating-point value = Normalized absolute value * gradient

Host sends: <Address>sr<STX>KYWE?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,....P20<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <STX>P21,P22,P23,....P40<ETX>[<BCC>]
Host sends: <ACK>
.
.
.
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1-P4000	Y difference values of the reference curve	Hex values without 0 termination

3.4.2.3 KYSU optimized transfer of the Y cumulative values of the reference curve

KYSU! Transfer values from host to unit in an optimized manner

With this command, the Y cumulative values (***normalized value: values corrected by the zero point, without taking into account the gradient***) of the reference curve can be transferred from the host to the unit.
The values are transferred in hexadecimal format without 0-termination (to save time).

The first value is transferred as an absolute value, all other values are transferred as a difference from the preceding value.

The difference values are transferred in blocks of up to 20 values (parameters 2 to 21). Parameter 1 is the block number. With a maximum of 4000 values there is thus a maximum of 200 blocks. The block number therefore gives the number of blocks, which each contain 20 values. Full 20 value containing blocks must always be transferred. However, the unit only considers values up until the number of values has been reached. In the last block, any of the 20 values, which remain unfilled can be padded with dummies. **The number of values must therefore be transferred prior to this command using the command RANZ!.**

Host sends: <Address>sr<STX>KYSU! P1,P2,....,P21<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Block number	Integer value: 1 to 200
P2-P21	Difference Y cumulative values of the reference curve	Hex values without 0 termination

KYSU? Transfer values from unit to host in an optimized manner

With this command, the Y cumulative values (***normalized value: values corrected by the zero point, without taking into account the gradient***) of the reference curve can be transferred from the unit to the host.

The first value is transferred as an absolute value, all other values are transferred as a difference from the preceding value. If more that two sequential difference values are the same, then these are no transferred individually,
but rather in the following form: M<factor>*<difference value>
M is the marker that indicates that this is not a single difference value.
It is followed by a factor that indicates the number of sequential difference values that are the same.
The * character separates factor and difference value.
It is followed by the difference value.

The parameter separator between the individual values is the comma.
10 values are always transferred, then a LF is transferred.
After host acknowledgement with ACK, the next 10 value pairs are transferred and so on.
Querying continues until there are no more available measurements.
If the last block has less than 10 difference values, then transfer is terminated early.
If there are no more available values, then an EOT is sent in response to the ACK acknowledgement.
There may be up to 4000 values. Thus there may be up to 400 blocks. The host must previously have retrieved the number of values using the RANZ? command.

The host must calculate the absolute values from the difference values (with the exception of the 1st value).
Absolute value[n] = absolute value[n-1] + difference value[n]

The transferred difference values and the absolute values calculated from them are normalized integer values (16 bit). These values are used with the gradient values, which are retrieved using the KRVA command, to calculate the floating-point values in their respective units.

Floating-point value = Normalized absolute value * gradient

Host sends: <Address>sr<STX>KYSU?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,....P10<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <STX>P11,P12,P13,....P20<ETX>[<BCC>]
Host sends: <ACK>
.
.
.
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1-P4000	Difference Y cumulative values of the reference curve	Hex values without 0 termination

3.4.2.4 KYTR optimized transfer of the Y mean value values of the reference curve (non-tracked values)

If trend is activated, this relates to the non-tracked values.

KYTR! Transfer values from host to unit in an optimized manner

With this command, the Y mean values (***normalized integer values: values corrected by the zero point, without taking into account the gradient***) of the reference curve can be transferred from the host to the unit.
The values are transferred in hexadecimal format without 0-termination (to save time).

The first value is transferred as an absolute value, all other values are transferred as a difference from the preceding value.

The difference values are transferred in blocks of up to 20 values (parameters 2 to 21). Parameter 1 is the block number. With a maximum of 4000 values there is thus a maximum of 200 blocks. The block number therefore gives the number of blocks, which each contain 20 values. Full 20 value containing blocks must always be transferred. However, the unit only considers values up until the number of values has been reached. In the last block, any of the 20 values, which remain unfilled can be padded with dummies. **The number of values must therefore be transferred prior to this command using the command RANZ!.**

Host sends: <Address>sr<STX>KYTR! P1,P2,....,P21<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Block number	Integer value: 1 to 200
P2-P21	Y difference values of the reference curve	Hex values without 0 termination

KYTR? Transfer values from unit to host in an optimized manner

With this command, the Y mean values (***normalized integer values: values corrected by the zero point, without taking into account the gradient***) of the reference curve can be transferred from the unit to the host.

The first value is transferred as an absolute value, all other values are transferred as a difference from the preceding value. If more that two sequential difference values are the same, then these are no transferred individually,
but rather in the following form: M<factor>*<difference value>
M is the marker that indicates that this is not a single difference value.
It is followed by a factor that indicates the number of sequential difference values that are the same.
The * character separates factor and difference value.
It is followed by the difference value.

The parameter separator between the individual values is the comma.
20 values are always transferred, then a LF is transferred.
After host acknowledgement with ACK, the next 20 value pairs are transferred and so on.
Querying continues until there are no more available measurements.
If the last block has less than 20 difference values, then transfer is terminated early.
If there are no more available values, then an EOT is sent in response to the ACK acknowledgement.
There may be up to 4000 values. Thus there may be up to 200 blocks. The host must previously have retrieved the number of values using the RANZ? command.

The host must calculate the absolute values from the difference values (with the exception of the 1st value).
Absolute value[n] = absolute value[n-1] + difference value[n]

The transferred difference values and the absolute values calculated from them are normalized integer values (16 bit). These values are used with the gradient values, which are retrieved using the KRVA command, to calculate the floating-point values in their respective units.

Floating-point value = Normalized absolute value * gradient

Host sends: <Address>sr<STX>KYTR?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,....P20<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <STX>P21,P22,P23,....P40<ETX>[<BCC>]
Host sends: <ACK>
.
.
.
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1-P4000	Y difference values of the reference curve	Hex values without 0 termination

3.4.2.5 KYMI optimized transfer of the delta Y minimum values of the reference curve

KYMI! Transfer values from host to unit in an optimized manner

With this command, the delta Y minimum values (***normalized integer values: values corrected by the zero point, without taking into account the gradient***) of the reference curve can be transferred from the host to the unit.
The values are transferred in hexadecimal format without 0-termination (to save time).

The first value is transferred as an absolute value, all other values are transferred as a difference from the preceding value.

The difference values are transferred in blocks of up to 20 values (parameters 2 to 21). Parameter 1 is the block number. With a maximum of 4000 values there is thus a maximum of 200 blocks. The block number therefore gives the number of blocks, which each contain 20 values. Full 20 value containing blocks must always be transferred. However, the unit only considers values up until the number of values has been reached. In the last block, any of the 20 values, which remain unfilled can be padded with dummies. **The number of values must therefore be transferred prior to this command using the command RANZ!.**

Host sends: <Address>sr<STX>KYMI! P1,P2,....,P21<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Block number	Integer value: 1 to 200
P2-P21	Difference values of the delta Y minimum values of the reference curve	Hex values without 0 termination

KYMI? Transfer values from unit to host in an optimized manner

With this command, the minimum Y values (***normalized integer values: values corrected by the zero point, without taking into account the gradient***) of the reference curve can be transferred from the unit to the host.

The first value is transferred as an absolute value, all other values are transferred as a difference from the preceding value. If more that two sequential difference values are the same, then these are no transferred individually,
but rather in the following form: M<factor>*<difference value>
M is the marker that indicates that this is not a single difference value.
It is followed by a factor that indicates the number of sequential difference values that are the same.
The * character separates factor and difference value.
It is followed by the difference value.

The parameter separator between the individual values is the comma.
20 values are always transferred, then a LF is transferred.
After host acknowledgement with ACK, the next 20 value pairs are transferred and so on.
Querying continues until there are no more available measurements.
If the last block has less than 20 difference values, then transfer is terminated early.
If there are no more available values, then an EOT is sent in response to the ACK acknowledgement.
There may be up to 4000 values. Thus there may be up to 200 blocks. The host must previously have retrieved the number of values using the RANZ? command.

The host must calculate the absolute values from the difference values (with the exception of the 1st value).
Absolute value[n] = absolute value[n-1] + difference value[n]

The transferred difference values and the absolute values calculated from them are normalized integer values (16 bit). These values are used with the gradient values, which are retrieved using the KRVA command, to calculate the floating-point values in their respective units.

Floating-point value = Normalized absolute value * gradient

Host sends: <Address>sr<STX>KYMI?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,....P20<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <STX>P21,P22,P23,....P40<ETX>[<BCC>]
Host sends: <ACK>
.
.
.
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1-P4000	Difference values of the delta Y minimum values of the reference curve	Hex values without 0 termination

3.4.2.6 KYMA optimized transfer of the delta Y maximum values of the reference curve

KYMA! Transfer values from host to unit in an optimized manner

With this command, the delta Y maximum values (***normalized integer values: values corrected by the zero point, without taking into account the gradient***) of the reference curve can be transferred from the host to the unit.
The values are transferred in hexadecimal format without 0-termination (to save time).

The first value is transferred as an absolute value, all other values are transferred as a difference from the preceding value.

The difference values are transferred in blocks of up to 20 values (parameters 2 to 21). Parameter 1 is the block number. With a maximum of 4000 values there is thus a maximum of 200 blocks. The block number therefore gives the number of blocks, which each contain 20 values. Full 20 value containing blocks must always be transferred. However, the unit only considers values up until the number of values has been reached. In the last block, any of the 20 values, which remain unfilled can be padded with dummies. **The number of values must therefore be transferred prior to this command using the command RANZ!.**

Host sends: <Address>sr<STX>KYMA! P1,P2,....,P21<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Block number	Integer value: 1 to 200
P2-P21	Difference values of the delta Y maximum values of the reference curve	Hex values without 0 termination

KYMA? Transfer values from unit to host in an optimized manner

With this command, the maximum Y values (***normalized integer values: values corrected by the zero point, without taking into account the gradient***) of the reference curve can be transferred from the unit to the host.

The first value is transferred as an absolute value, all other values are transferred as a difference from the preceding value. If more that two sequential difference values are the same, then these are no transferred individually,
but rather in the following form: M<factor>*<difference value>
M is the marker that indicates that this is not a single difference value.
It is followed by a factor that indicates the number of sequential difference values that are the same.
The * character separates factor and difference value.
It is followed by the difference value.

The parameter separator between the individual values is the comma.
20 values are always transferred, then a LF is transferred.
After host acknowledgement with ACK, the next 20 value pairs are transferred and so on.
Querying continues until there are no more available measurements.
If the last block has less than 20 difference values, then transfer is terminated early.
If there are no more available values, then an EOT is sent in response to the ACK acknowledgement.
There may be up to 4000 values. Thus there may be up to 200 blocks. The host must previously have retrieved the number of values using the RANZ? command.

The host must calculate the absolute values from the difference values (with the exception of the 1st value).
Absolute value[n] = absolute value[n-1] + difference value[n]

The transferred difference values and the absolute values calculated from them are normalized integer values (16 bit). These values are used with the gradient values, which are retrieved using the KRVA command, to calculate the floating-point values in their respective units.

Floating-point value = Normalized absolute value * gradient

Host sends: <Address>sr<STX>KYMA?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,....P20<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <STX>P21,P22,P23,....P40<ETX>[<BCC>]
Host sends: <ACK>
.
.
.
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1-P4000	Y difference values of the delta Y maximum values of the reference curve	Hex values without 0 termination

3.5 9310-V2xxx specific commands (Black box module)

3.5.1 TRAX Transmitter supply X channel

Input of a new value

With the command TRAX! The transmitter supply of chaanel X can be switched.

Host sends: <Address>sr<STX>TRAX! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Transmitter supply for channel X	0 -> OFF 1 -> ON

Readout of present value

With the command TRAX? the present status of the transmitter supply can be read out.

Host sends: <Address>sr<STX>TRAX?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Transmitter supply for channel X	0 -> Supply is switched OFF 1 -> Supply is switched ON

3.5.2 TRAY Transmitter supply Y channel

Input of a new value

With the command TRAY! The transmitter supply of channel Y can be switched.

Host sends: <Address>sr<STX>TRAY! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Transmitter supply for channel Y	0 -> OFF 1 -> ON

Readout of present value

With the command TRAY? the present status of the transmitter supply can be read out.

Host sends: <Address>sr<STX>TRAY?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Transmitter supply for channel Y	0 -> Supply is switched OFF 1 -> Supply is switched ON

3.6 Error status

3.6.1 FSTA Error status

The command FSTA? can read out the fault status on the serial interface. The reason for the last transferred NAK might be indicated here. The fault status is bit coded, i.e. multiple bits can be set simultaneously.

Host sends: <Address>sr<STX>FSTA?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Value	Meaning
P1	0x0001	PREFIX address error
	0x0002	Enquiry received in Slave Mode
	0x0004	Block check error
	0x0008	Commando error
	0x0010	Parameter Fehler
	0x0020	Timeout Receive Timer
	0x0040	Timeout Response Timer
	0x0080	Unvalid ! or ? sign
	0x0100	Unvalid configuration
	0x0200	Scale error
	0x0400	No valid measurement can be found
	0x0800	A/D-changer overdriven
	0x1000	EEPROM reading error during loading of basic calibration
	0x2000	Overdriven by scaling
	0x4000	The transfer of a measurement curve was cancelled by a start of a new measurement
	0x8000	Unvalid envelope borders

3.6.2 ERRO? Read out of device error status

Readout of present value

With the command ERRO? The DIGIFORCE error status can be read out.

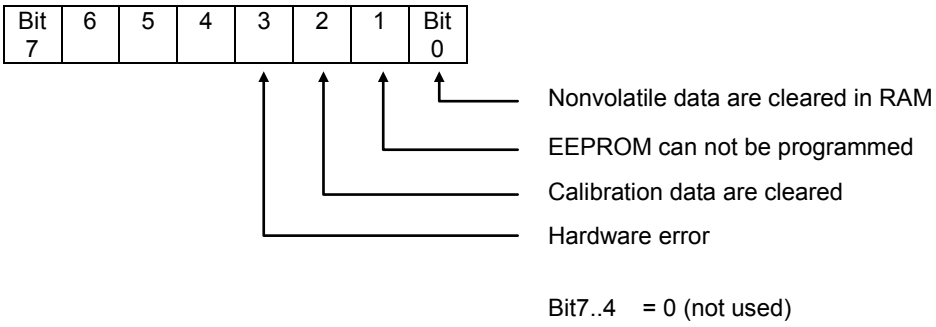
Host sends: <Address>sr<STX>ERRO?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	**meaning**	**value**
P1	DIGIFORCE error code	Error code (bit coded)

Error code



Bit7..4 = 0 (not used)

3.7 Testmode

3.7.1 TEST Testmode On/Off

Input of a new value

With the command TEST! the test mode can be switched on or off. In normal operation the test mode have to be switched off. To use the following test commands (BERX, DMSY, PIEY,SPEI, POTB, SPSO, ALED) you have to switch the test mode on. While switching off the test mode the original device settings are available again.

Host sends: <Address>sr<STX>TEST! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Testmode	1 → set testmode on 0 → set testmode off

Readout of present value

With the command TEST? The current status of testmode can be read out.

Host sends: <Address>sr<STX>TEST?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Testmode	1 → Testmode on 0 → Testmode off

3.7.2 BERX Setting of the X-channel range

This command is only vaild if the Test mode is active

Input of a new value

With the commmand BERX! the X-channel range can be set for test reason.

Host sends: <Address>sr<STX>BERX! P1 <ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	X-channel range	0 -> 5 V 1 -> 10 V

Readout of present value

With the commmand BERX! the current X-channel range can be read out.

Host sends: <Address>sr<STX> BERX?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1 <ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	X-channel range	0 -> 5 V 1 -> 10 V

3.7.3 DMSY Setting of Y-channel range (DMS type)

This command is only vaild if the Test mode is active

Input of a new value

With the command DMSY! the Y-channel range can be set for test reason.

Host sends: <Address>sr<STX> DMSY! P1 <ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Y-channel range (DMS type)	0 -> 2,5 mV range 1 -> 5 mV 2 -> 10 mV 3 -> 25 mV 4 -> 50 mV 5 -> 100 mV 6 -> 5 V

Readout of present value

With the commmand DMSY? the current Y-channel range can be read out.

Host sends: <Address>sr<STX> DMSY?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1 <ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Y-channel range (DMS type)	0 -> 2,5 mV range 1 -> 5 mV 2 -> 10 mV 3 -> 25 mV 4 -> 50 mV 5 -> 100 mV 6 -> 5 V

3.7.4 PIEY Setting of Y-channel range (Piezo type)

This command is only vaild if the Test mode is active

Input of a new value

With the command PIEY! the Y-channel range of the Piezo amplifier can be set for test reason.

Host sends: <Address>sr<STX> PIEY! P1 <ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Y-channel range (Piezo type)	0 -> 1 nC range 1 -> 2 nC 2 -> 5 nC 3 -> 10 nC 4 -> 20 nC 5 -> 50 nC 6 -> 100 nC 7 -> 200 nC 8 -> 400 nC

Readout of present value

With the command PIEY? the current Y-channel range of the Piezo amplifier can be read out.

Host sends: <Address>sr<STX> PIEY?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1 <ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Y-channel range (Piezo type)	0 -> 1 nC range 1 -> 2 nC 2 -> 5 nC 3 -> 10 nC 4 -> 20 nC 5 -> 50 nC 6 -> 100 nC 7 -> 200 nC 8 -> 400 nC

3.7.5 SPEI Setting of the Y-channel supply (DMS type)

This command is only vaild if the Test mode is active

Input of a new value

With the command SPEI! the Y-channel supply can be set for test reason.

Host sends: <Address>sr<STX> SPEI! P1 <ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Y-channel supply (DMS type)	0 -> 2,5 V 1 -> 5 V

Readout of present value

With the command SPEI! the current Y-channel supply can be read out.

Host sends: <Address>sr<STX> SPEI?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1 <ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Y-channel supply (DMS type)	0 -> 2,5 V 1 -> 5 V

3.7.6 POTB Adjustment of the potential connection for strain gauge supply

This command is only vaild if the Test mode is active

Input of a new value

The potential connection of the strain gauge supply is set using the POTB! command.

Host sends: <Address>sr<STX> POTB! P1 <ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Potential connection of strain gauge supply	0 -> internal 1 -> external source

Readout of present value

The potential connection of the DMS supply is retrieved using the POTB? command.

Host sends: <Address>sr<STX> POTB?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1 <ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Potential connection of strain gauge supply	0 -> internal 1 -> external source

3.7.7 UINT? Internal voltage values

Readout of present value

With the command UINT? device internal voltage signals can be read out.

Host sends: <Address>sr<STX> UINT?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,P4 <ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Supply X-channel	Float (value in V) (approx. 5 V)
P2	Supply Y-channel (not for Piezo interface)	Float (value in V) (approx. 2.5 V or 5 V)
P3	Knotenpunkt	Float (approx. 1,7 V)
P4	Kurzschluss (Nullpunkt)	Float (approx. 0 V)

3.7.8 ROHW? A/D converter and voltage values

Readout of present value

With the command ROHW? A/D-converter and voltage values can be read out.

Host sends: <Address>sr<STX> ROHW?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1,P2,P3,P4,P5,P6 <ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	A/D converter X-channel	Integer
P2	Voltage value X-channel	Float
P3	Unit of X-channel voltage value	V
P4	A/D converter Y-channel	Integer
P5	Voltage value X-channel (DMS type) Charge of Y-channel (Piezo type)	Float
P6	Unit of Y-Kanal (voltage or charge)	mV, V or nC

3.7.9 Test of the PLC inputs and outputs

3.7.9.1 SPSI **Read the PLC inputs**

With the command SPSI? the nine PLC inputs can be read out.

Host sends: <Address>sr<STX>SPSI?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1<ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Bitcoded PLC inputs	Short value in hex 0 to 1F

Bit number	Signal name of the PLC input
0	I-AUTO
1	I-PROG2
2	I-PROG1
3	I-PROG0
4	I-STROBE
5	I-SENSOR-TEST
6	I-RESET-STAT
7	I-TARA-Y
8	I-START

3.7.9.2 SPSO **Set the PLC outputs**

This command is only vaild if the Test mode is active
With the command SPSO! ? the twelve PLC outputs can be read out.

Host sends: <Address>sr<STX>SPSO! P1<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Bitcoded PLC outputs	Short value in hex 0 to FFF

Bit number	Signal name of the PLC input
0	O-MESS-AKTIV
1	O-PROG2
2	O-PROG1
3	O-PROG0
4	O-STROBE
5	O-IO-S-TEST
6	O-S2
7	O-S1
8	O-NIO-ONL
9	O-NIO
10	O-IO
11	O-READY

3.7.10 ALED! Test of the warning LED

This command is only vaild if the Test mode is active

Input of a new value

With the command ALED! the warning LED can be tested.

Host sends: <Address>sr<STX> ALED! P1 <ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Switch warning LED on / off	0 -> warning LED on 1 -> warning LED off

There is no xxx? Form of this command

3.7.11 RESE! Test of the RESET button

Readout of present value

With the command RESE? The status of the reset button can be read out.

Host sends: <Address>sr<STX> RESE?<ETX>[<BCC>]
DIGIFORCE answers: <ACK>
Host sends: <EOT>

Host sends: <Address>po<ENQ>
DIGIFORCE answers: <STX>P1 <ETX>[<BCC>]
Host sends: <ACK>
DIGIFORCE answers: <EOT>

Meaning of parameter Pn

Parameter	Meaning	Value
P1	Status of the reset button	0 -> RESET button is pressed 1 -> RESET button is not pressed