Instruction manual

RS232 interface with ProPar protocol for digital multibus Mass Flow / Pressure instruments

Doc. no.: 9.17.027AI Date: 03-05-2022

ATTENTION
Please read this instruction manual carefully before installing and operating the instrument.
Not following the guidelines could result in personal injury and/or damage to the equipment.

Bronkhorst®

Disclaimer

The information in this manual has been reviewed and is believed to be wholly reliable. No responsibility, however, is assumed for inaccuracies. The material in this manual is for information purposes only.

Copyright

Symbols

Important information. Discarding this information could cause injuries to people or damage to the Instrument or installation.

Helpful information. This information will facilitate the use of this instrument.

Additional info available on the internet or from your local sales representative.

Warranty

Bronkhorst® products are warranted against defects in material and workmanship for a period of three years from the date of shipment, provided they are used in accordance with the ordering specifications and the instructions in this manual and that they are not subjected to abuse, physical damage or contamination. Products that do not operate properly during this period may be repaired or replaced at no charge. Repairs are normally warranted for one year or the balance of the original warranty, whichever is the longer.

See also paragraph 9 of the Conditions of sales:
http://www.bronkhorst.com/files/corporate_headquarters/sales_conditions/en_general_terms_of_sales.pdf

The warranty includes all initial and latent defects, random failures, and undeterminable internal causes.

It excludes failures and damage caused by the customer, such as contamination, improper electrical hook-up, physical shock etc.

Re-conditioning of products primarily returned for warranty service that is partly or wholly judged non-warranty may be charged for.

Bronkhorst High-Tech B.V. or affiliated company prepays outgoing freight charges when any party of the service is performed under warranty, unless otherwise agreed upon beforehand. However, if the product has been returned collect to our factory or service center, these costs are added to the repair invoice. Import and/or export charges, foreign shipping methods/carriers are paid for by the customer.

Table of contents

1 GENERAL PRODUCT INFORMATION

1.1 INTRODUCTION

This manual will explain how to communicate with a Bronkhorst® instrument to your PC/PLC using RS232 serial communication. You have to write software yourself using the information of this document in order to be able to operate these instruments. Bronkhorst® also offers software to easily operate digital instruments with your PC using Microsoft Windows. On the highest supported communication level, you may use FlowDDE channels for Windows application-programs with this facility. You can use the program FlowDDE for easy connection between Windows applications (e.g. Excel, Visual Basic, LabVIEW, Delphi, Borland C) and digital instruments. There are several examples available for LabVIEW, Visual Basic and Excel environments.

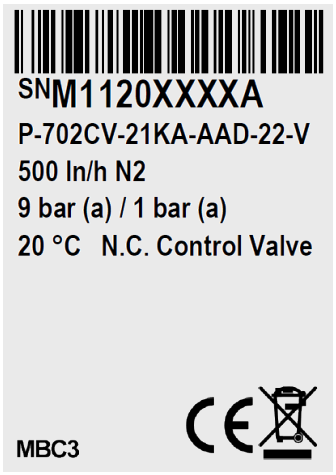
On a lower communication level, you can also use the FLOWB32.DLL for reading/changing parameter values.

To read and write parameter values from or to ProPar devices directly through the available interfaces there is a special protocol for messages between these devices. This protocol is named ProPar and has been specially developed for Bronkhorst® equipment so no third party equipment can be connected. It consists of a hierarchical setup for instruments / nodes (max. 126) containing processes (max. 127) with parameters (FBnr) (max. 32) which values can be set to certain values to enable settings/properties for the instruments.

When operating a ProPar system with a HOST computer, you need to know this message protocol if you choose to drive the interfaces directly. When you communicate directly via RS232 on a Multibus instrument or when you use a RS232 to FLOW-BUS (baud rates up to 38K4 with switch and 2 LED's) interface, no special initialisation is needed.

1.2 MULTIBUS TYPES

In 2000 Bronkhorst® developed their first digital instruments according to the "multibus" principle. The basic pc-board on the instrument contained all of the general functions needed for measurement and control, including alarm, totalizing and diagnostic functions. It had **analog** I/O-signals and also an **RS232** connection as a standard feature. In addition to this there is the possibility of integrating an interface board with **DeviceNet™**, **PROFIBUS DP**, **Modbus** , **FLOW-BUS** or **EtherCAT** protocol. The first generation (**MBC-I**) was based on a 16 bit Fujitsu controller. It was superseded in 2003 by the Multibus type 2 (**MBC-II)**. This version was also based on the 16 bit Fujitsu controller but it had several improvements to the MBC-I. One of them is the current steering of the valve. It reduced heat production and improved control characteristics. The latest version Multibus controller type 3 (**MBC3**) is introduced in 2011. It is built around a 72MHz 32 bit NXP ARM controller. It has AD and DA controllers on board which makes it possible to measure noise free and control valves without delays. The internal control loop runs 6 times faster compared to the MBC-II therefore control stability has improved significantly. It also has several improved functions like reverse voltage protection, inrush current limitation and overvoltage protection.

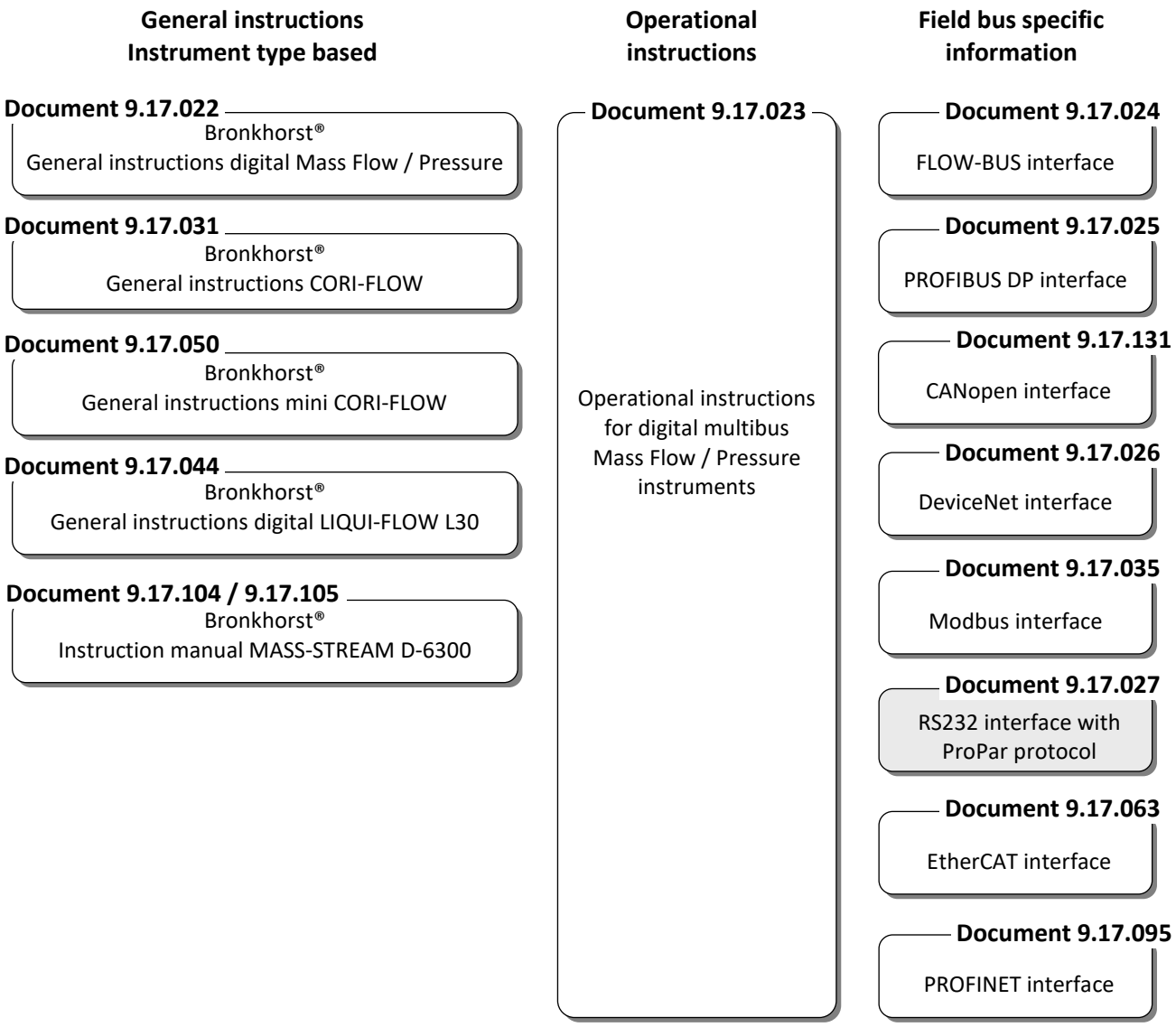


MBC3 instruments can be recognised by the "MBC3" placed on lower left side of the instrument label (see example).

1.3 REFERENCES TO OTHER APPLICABLE DOCUMENTS

Manuals and guides for digital instruments are modular. General instructions give information about the functioning and installation of instruments. Operational instructions explain the use of the digital instruments features and parameters. Field bus specific information explains the installation and use of the field bus installed on the instrument.

1.3.1 Manuals and user guides:



1.3.2 Technical Drawings:

Hook-up diagram laboratory-style MBC RS232 + analog	(document nr. 9.16.062)
Hook-up diagram industrial style MBC-II RS232 + analog	(document nr. 9.16.051)
Hook-up diagram CORI-FLOW RS232 + analog	(document nr. 9.16.044)
Hook-up diagram LIQUI-FLOW L30 digital RS232 + analog	(document nr. 9.16.073)

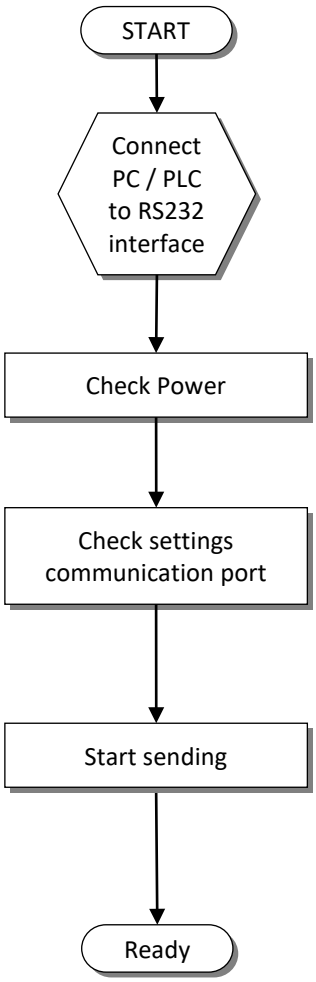
1.3.3 Software tooling:

FlowPlot
FlowView
Flowfix
FlowDDE

All these documents can be found at:
http://www.bronkhorst.com/en/downloads

1.4 SHORT FORM START-UP

All necessary settings for this device are already performed at Bronkhorst®.
To follow next steps carefully is the quickest way to get this module operational in your own system.



Make sure your PC or PLC is connected to the RS232 interface by means of the correct cable.

- Multibus instruments need a special cable with T-connector.
- RS232 to ProPar interfaces need a one-on-one 9-pole cable without crossings with male and female connector

Cable lengths for RS232 must not exceed 10 meters.

Make sure instrument or interface is powered (+15…24Vdc)

Use settings [38400,n ,8 ,1] for your COM-port:
Baud rate = 38K4 Baud, no parity, 8 data bits, 1 stop bit.

Start sending messages as described in following paragraphs.

In case of trouble programs like Hyper terminal (available in MS-Windows) or FlowDDE (from Bronkhorst®) could be very useful.

2 INTERFACES

2.1 RS232/FLOW-BUS INTERFACE

The RS232 to ProPar interface is an interface between the FLOW-BUS and the RS232 V24 serial (computer) port. It will either be supplied as a separate enclosed unit with a FLOW-BUS connector and a RS232 connector or as an integral 14TE module of your E-8000 readout and control system. The converter offers communication with a baud rate up to 38400 baud. Communication software support is available. Communication settings are: 38400, n, 8, 1.

2.1.1 D-connector for RS232

The female RS232 (x) (sub miniature 9-pin) D-connector has the following pin configuration:

Pinnumber	Description
1	not connected
2	TXD
3	RXD
4	not connected
5	0 Vd
6	DTR
7	CTS
8	RTS
9	Shield

2.2 RS232 ON MULTIBUS INSTRUMENT

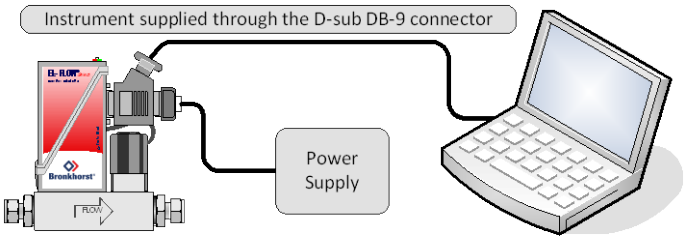
The RS232 interface on a Multibus instrument can be connected to any RS232 V24 serial (computer) port. Make sure to respect the hook-up diagram. Bronkhorst® offers special cables for communication, separating the RS232 lines from the power and analog in- and output. On the 9-pin male D-sub connector of the instrument RX and TX are available on pin 6 and pin 1.

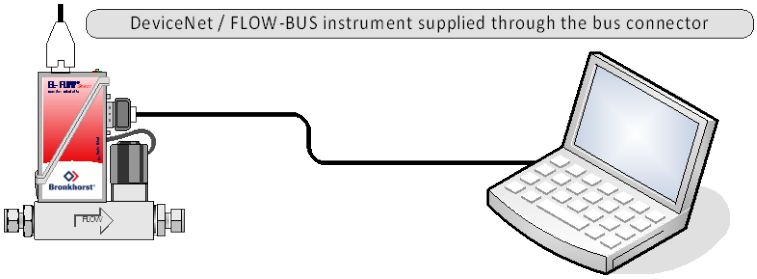
RS232 communication is possible by:

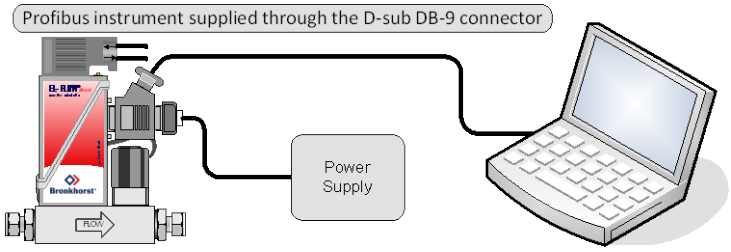
- 9-pin Sub D-connector (non IP65 applications, e.g. EL-FLOW)
- 8 DIN connector (IP65 applications, e.g. CORI-FLOW)

For the exact connections please advise your hook-up diagram.

2.2.1 Applications, e.g. EL-FLOW







By default, the interface offers communication at a baud rate of 38400 baud. On instruments that offer the possibility to change the RS232 baud rate, the baud rate may be configured differently. See the technical documentation of your instrument which baud rates are supported.

2.2.2 Facilities

No handshaking facilities are used. On the side of the PC/PLC a nul-modem connector is needed.
Communication settings are: 38400,n,8,1.
Bronkhorst® offers a special cable needed for communication.
Communication software support is available.

2.2.3 Baudrates multibus RS232:

Baud rates MBC-II	38400 Baud
Baud rates MBC3	9600 Baud 19200 Baud 38400 Baud 57600 Baud 115200 Baud 230400 Baud 460800 Baud

2.2.4 Changing the RS232 Baudrate via FLOWDDE software (MBC3 only)

Start the FLOWDDE Server software, open the communication and write and read the parameters as adviced below.
FlowDDE Server software: menu "Flow-BUS" → "test Flow-BUS and DDE"
At 'Test FLOW-BUS' select your Channel and Parameter(see below):

- Parameter 7: (initreset) → Write value 64 (actual value is 82)
- Parameter 7 (initreset)→ Read parameter and check value

activate 'Configuration Mode' - Serial communication is forced to RS232 ProPar protocol at a baudrate of 38400 baud.

- Parameter 307 (Fieldbus2 mode) → Write value 1 (actual value is 0)

configure baudrate for RS232 ProPar communicaton at the SubD9 / 8DIN /M12 connector.

- Parameter 310 (Fieldbus2 baudrate) → Write your desired baudrate value e.g. 19200

Disable 'Configuration Mode' – Your baudrate will be applied immediately
Likely you will loose communcuation between the FLOWDDE software and the instrument.

- Parameter 307 (Fieldbus2 mode) → Write value 0 (actual value is 1)

Test your new RS232 Baudrate:
Reconfigure the FLOWDDE server communication settings to match your new configured baudrate

Start the FLOWDDE Server software, open the communication and write and read the parameters as adviced below.
FlowDDE Server software: menu "Flow-BUS" → "test Flow-BUS and DDE"
At 'Test FLOW-BUS' select your Channel and Parameter(see below):

- Parameter 7: (initreset) → Write value 82 (actual value is 64)
- Parameter 7 (initreset)→ Read parameter and check value

Now the RS232 Baudrate is re-configured.

3 PROPAR PROTOCOL DESCRIPTION

3.1 GENERAL

On the highest supported communication level, you may use DDE-channels for Windows application-programs with this facility. On a lower communication level, you can use the FLOWB32.DLL, for changing parameter values.

To read and write parameter values from or to ProPar devices directly through the available interfaces there is a special protocol for messages between these devices. When operating a ProPar system with a HOST computer, you need to know this message protocol if you choose to drive the interfaces directly.

There are two different communication protocols for the PC and the RS232 HOST:

- an ASCII protocol for communication that is compatible with existing ProPar applications.
  This protocol serves only one master/slave dialog at a time.

- an enhanced binary protocol that supports concurrent sending of messages to different nodes. This protocol contains a message-sequence number and serves more than one master/slave dialogs at a time.

A ProPar device automatically recognises the protocol used by the PC and adapts its behaviour to the protocol in use. The type of protocol is determined by the first character of a message.

- The first character is ':' (0x3A) existing type of message.
- The first character is DLE (0x10) enhanced type of message.

Via the ProPar DLL (FLOWB32.DLL) the PC determines which protocol is in use.

The communication relation is always master (PC) and slave (HOST). The HOST will always respond on a request from the PC.

3.2 INITIALISATION OF LOCAL HOST INTERFACES ON MULTIBUS INSTRUMENTS

When you use a digital instrument with RS232 interface, baud rate is fixed on 38K4 baud and no special initialisation is needed. Through the serial line connected to a COM-port of your computer or to a PLC you have to communicate with the instrument using the ProPar protocol. Each instrument has its own node address (3...120). If you want to send a message to the instrument you have to know this node address. However, if you send a message to node address 128 the instrument will always respond to your message. On a point-to-point connection like RS232 it is the easiest way to make the communication work under all circumstances (it is independent of the real node address of the instrument).

3.3 PROPAR MESSAGE FORMAT

3.3.1 Basic ProPar datalink format

The basic data link message format has the following fields:

node	message destination	length	data field length	data	data	Data	etc.

In the ProPar environment the data field may contain up to 256 bytes of data. In the HOST application described here, the messages are according to PROPAR coding rules and the data field will contain a maximum of 64 bytes.

3.3.2 ASCII table

The **American Standard Code for Information Interchange** (**ASCII**) is a character-encoding scheme based on the ordering of the English alphabet. ASCII codes represent text in computers, communications equipment, and other devices that use text. Most modern character-encoding schemes are based on ASCII.

Dec	Oct	Hex	Binary	Code	Dec	Oct	Hex	Binary	Code	Dec	Oct	Hex	Binary	Code	Dec	Oct	Hex	Binary	Code
32	040	20	0100000	SP	56	070	38	0111000	8	80	120	50	1010000	P	104	150	68	1101000	h
33	041	21	0100001	!	57	071	39	0111001	9	81	121	51	1010001	Q	105	151	69	1101001	i
34	042	22	0100010	"	58	072	3A	0111010	:	82	122	52	1010010	R	106	152	6A	1101010	j
35	043	23	0100011	#	59	073	3B	0111011	;	83	123	53	1010011	S	107	153	6B	1101011	k
36	044	24	0100100	$	60	074	3C	0111100	<	84	124	54	1010100	T	108	154	6C	1101100	l
37	045	25	0100101	%	61	075	3D	0111101	=	85	125	55	1010101	U	109	155	6D	1101101	m
38	046	26	0100110	&	62	076	3E	0111110	>	86	126	56	1010110	V	110	156	6E	1101110	n
39	047	27	0100111	'	63	077	3F	0111111	?	87	127	57	1010111	W	111	157	6F	1101111	o
40	050	28	0101000	(	64	100	40	1000000	@	88	130	58	1011000	X	112	160	70	1110000	p
41	051	29	0101001	)	65	101	41	1000001	A	89	131	59	1011001	Y	113	161	71	1110001	q
42	052	2A	0101010	*	66	102	42	1000010	B	90	132	5A	1011010	Z	114	162	72	1110010	r
43	053	2B	0101011	+	67	103	43	1000011	C	91	133	5B	1011011	[	115	163	73	1110011	s
44	054	2C	0101100	,	68	104	44	1000100	D	92	134	5C	1011100	\	116	164	74	1110100	t
45	055	2D	0101101	-	69	105	45	1000101	E	93	135	5D	1011101	]	117	165	75	1110101	u
46	056	2E	0101110	.	70	106	46	1000110	F	94	136	5E	1011110	^	118	166	76	1110110	v
47	057	2F	0101111	/	71	107	47	1000111	G	95	137	5F	1011111	_	119	167	77	1110111	w
48	060	30	0110000	0	72	110	48	1001000	H	96	140	60	1100000	`	120	170	78	1111000	x
49	061	31	0110001	1	73	111	49	1001001	I	97	141	61	1100001	a	121	171	79	1111001	y
50	062	32	0110010	2	74	112	4A	1001010	J	98	142	62	1100010	b	122	172	7A	1111010	z
51	063	33	0110011	3	75	113	4B	1001011	K	99	143	63	1100011	c	123	173	7B	1111011	{
52	064	34	0110100	4	76	114	4C	1001100	L	100	144	64	1100100	d	124	174	7C	1111100	|
53	065	35	0110101	5	77	115	4D	1001101	M	101	145	65	1100101	e	125	175	7D	1111101	}
54	066	36	0110110	6	78	116	4E	1001110	N	102	146	66	1100110	f	126	176	7E	1111110	~
55	067	37	0110111	7	79	117	4F	1001111	O	103	147	67	1100111	g	127	177	7F	1111111	DEL

3.3.3 ProPar ASCII protocol

An ASCII protocol is used on the existing RS232-HOST. To be compatible with existing driver software the ASCII protocol is available.

A basic data link message is coded in ASCII as follows:

	length		**node**		**data**		
:	len1	len2	node1	node2	data1	data2	CR

Byte	**Explanation**
:	Initial character (colon)
Len1, len2	Length of message including the node address in *bytes*, so (len1, len2) is the basic message length +1.
node1, node2	node address of destination (PC to HOST) node address of source (HOST to PC)
data1, data2	message field
CR	termination character

All bytes (except the initial and termination character) are converted from 1 binary byte to 2 hexadecimal bytes in ASCII representation.

Example: binary data byte 0x2A --> hexadecimal ASCII characters 0x32, 0x41.

A special message type is used to pass error messages from the HOST to the PC. Its structure is as follows:

	0x01		error		
:	0x30	0x31	error1	error2	CR

Byte	**Explanation**
:	initial character
0x30, 0x31	length of the message (1 byte)
error	error code, two digit HEX number
CR	termination character

The error code can have the following values:

Value	**Meaning**	**ASCII error response**
1	general error	:0101
2	general error	:0102
3	propar protocol error	:0103
4	propar protocol error (or CRC error)	:0104
5	destination node address rejected	:0105
8	general error	:0108
9	response message timeout	:0109

3.3.3.1 ProPar enhanced binary protocol

The enhanced protocol is binary coded. Control sequences are used to recognise the beginning and end of a message in a byte stream. A control sequence starts with a DLE byte (0x10) and is followed by a control byte. The following control sequences are defined:

First byte	Second byte	Function
DLE (0x10)	STX (0x02)	Start of message
DLE (0x10)	ETX (0x03)	End of message
DLE (0x10)	DLE (0x10)	Data byte 0x10 extra byte must not be counted for the 'len' byte.
DLE (0x10)	any other character	Not allowed. Messages that contain such a sequence will be ignored. The receiver waits until a new DLE STX sequence.

The [DLE DLE] sequence is used to prevent possible DLE bytes in the transmitted binary data stream from being recognised as the start of a control sequence. The sender replaces any DLE bytes in the data by two DLE bytes. The data link of the receiver will convert a [DLE DLE] sequences to one DLE byte.
See chapter 'Examples data byte 0x10 - ProPar enhanced binary protocol'

If a RS232 error (receiver overrun, framing error, not allowed control sequence) occurs, the data link frame is ignored.

The enhanced binary coded messages between PC and HOST are structured as follows:

DLE	STX	seq	node	len	data	DLE	ETX

Byte	Explanation
DLE, STX	start sequence
seq	message sequence number
node	node address of destination (PC to HOST) node address of source (HOST to PC)
len	length of data field in bytes (extra 0x10 byte must not be counted)
data	message field
DLE, ETX	end sequence

The enhanced protocol allows the transmission of more than one request at a time. The sequence number makes it possible to associate the answer to the according request. The HOST has more than one message buffer where messages may be stored (typical 5). When the message buffers are full, the HOST responds with an error message.
The responses from the HOST to the PC have the same message format as the request. An error message has a special format:

DLE	STX	seq	node	0x00	error	DLE	ETX

Byte	Explanation
DLE, STX	start sequence
seq	message sequence number, as in request
node	node address of source, as in request
error	error code
DLE, ETX	end sequence

The error code can have the following values:

Error code	Meaning
0x01	general error
0x02	general error
0x03	propar protocol error
0x04	propar protocol error (or CRC error)
0x05	destination node address rejected
0x08	general error
0x09	response message timeout

3.4 PROPAR COMMUNICATION MESSAGES

Communication messages between ProPar interfaces and other devices consist of command strings with specific information. This command string is either ASCII (RS232) or BINARY. Basically the string contains several information bytes. Through RS232 these hexadecimal bytes are converted in ASCII (e.g.: byte value 0x0A is "0A" in ASCII and capital letters should be used). Messages via RS232 are preceded by the ":" character and terminated with "\r\n" (Carriage return-Line-feed).

There are several COMMANDS available in the ProPar messages. Only command RD (04) and WR (01) are required for all the standard parameter reading and writing. A RD command will be answered with a WR command, containing the value asked for or a status message, containing an error number. A WR command will be answered with a status message, containing an error number (if error number = 0, than WR command was OK).

ASCII character : has hexadecimal value: 3A
ASCII character ' \r ' has hexadecimal value: 0D
ASCII character ' \n ' has hexadecimal value: 0A

3.4.1 Communication commands

Communication commands	
Command	**Description**
00	Status message
01	Send parameter with destination address, will be answered with type 00 command
02	Send parameter with destination address, no status requested
03	Send parameter with source address, no status requested
04	Request parameter, will be answered with type 02 or 00 command
06	Stop process
07	Start process
08	Claim process
09	Unclaim process

To access a specific parameter you need to know the following points.

- Node address — each ProPar device is connected to a specific node address in the system.
- Process number — each device (node) consists of several processes.
- Parameter number (FBnr) — each process consists of several parameters.
- Parameter type — each parameter can be of a different type and value.

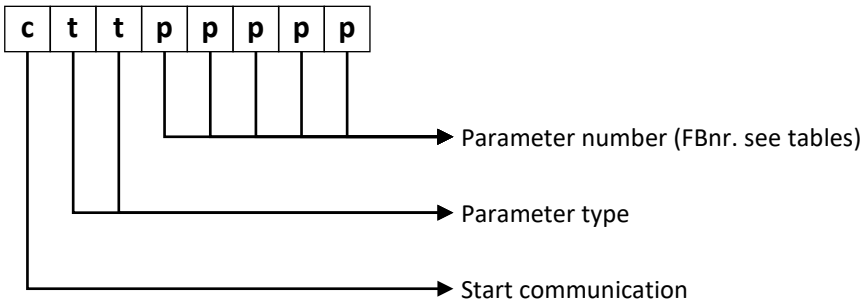
For parameters numbers and values see tables "parameter properties" and "parameter values" in this manual.

3.4.2 Parameter types

Parameter types				
Type	**Id**	**Bytes**	**Bits**	**Range**
Character	00h	1	00000000	0…255
Integer	20h	2	00100000	0…65535
Float	40h	4	01000000	+-1.18e-38...+-3.39e+38
Long	40h	4	01000000	4 bytes 0... 4294967296
String	60h	X	01100000	length needs to be specified

Number of bytes per parameter type					
Value 1	Value 2	Value 3	Value 4	Value 5	Value 6
Byte					
Integer					
Float					
Long					
String length	String				

The messages make use of bytes which are composed of data from several bytes.
An example of such a composed byte is shown below.



3.5 CHAINING

Chaining can be used to send or request more than one parameter per message. When the parameters are all members of the same process, they can be chained at parameter level. When the parameters are members of different processes, they can be chained at process level. A combination is also possible. For chaining at parameter level the first bit of the parameter number should be set if there is following another parameter at the same process. For chaining at process level the first bit of the process number should be set if there is another process following.

3.6 STATUS MESSAGE

A write to a device with the command “01” will result in a status message. Below the possible returns are listed.

Nr	Byte	Description
0	:	Start character
1	04	Fixed message length 4.
2	Node	Node address
3	00	Command status
4	Status	00 No error 01 Process claimed 02 Command error 03 Process error 04 Parameter error 05 Parameter type error 06 Parameter value error 07 Network not active 08 Time-out start character 09 Time-out serial line 0A Hardware memory error 0B Node number error 0C General communication error 0D Read only parameter. 0E Error PC-communication 0F No RS232 connection 10 PC out of memory 11 Write only parameter 12 System configuration unknown 13 No free node address 14 Wrong interface type 15 Error serial port connection 16 Error opening communication 17 Communication error 18 Error interface bus master 19 Timeout answer 1A No start character 1B Error first digit 1C Buffer overflow in host 1D Buffer overflow 1E No answer found 1F Error closing communication 20 Synchronisation error 21 Send error 22 Protocol error 23 Buffer overflow in module
5	Index or Claimed process	Index pointing to the first byte in the send message for which the above status applies. In case of the status CLAIM ERROR, this field contains the claimed process.
6	\r	Carriage Return
7	\n	Line Feed

Value from byte 5 of status message may be neglected if value of byte 4 = 0

3.7 SEND PARAMETERS

To write to a device command codes “01”and “02”must be used. “01” will give as result a status message for error checking. Command “02” will give no reply from the device. In the table below the total command line is explained.

Sending a parameter				
Nr	**Byte**	**Layout**	**Description**	
0	:		start character	
1	Length		Message length	
2	Node		Node address	
3	01 or 02		Command write, for type 01 a status message (00) will be returned	
4	Process	cppppppp	c	Process chained
			p	Process number
5	Parameter	cttppppp	c	Parameter chained
			t	Parameter type
			p	Parameter number (FBnr.)
6	Value 1		Value for all types. For ‘strings’ this field contains the string length.	
7	Value 2		Value for type ‘integer’, ‘float’ or ‘long’.	
8	Value 3		Value for type ‘float’ or ‘long’.	
9	Value 4		Value for type ‘float’ or ‘long’.	
X	Value x		More value fields follow for type ‘string’ depending on string length. If given string length is zero, the final field should also contain a zero.	
X+1	\r		Carriage Return	
X+2	\n		Line Feed	

3.8 REQUEST PARAMETER

For each requested parameter an index number can be given. The answering node will return this index number with the requested parameter. This can be used to check which parameter is returned when several parameters are requested.

Requesting a parameter				
Nr	**Byte**	**Layout**	**Description**	
0	:		start character	
1	Length		Message length	
2	Node		Node address	
3	04		Command read	
4*	Process	cppppppp	c	Process chained
			p	Process number
5*	Parameter	cttnnnnn	c	Parameter chained
			t	Parameter type
			n	Parameter index 0…31
6	Process	-ppppppp	-	Not used (use a zero)
			p	Process number
7	Parameter	-ttppppp	-	Not used (use a zero)
			t	Type parameter
			t	Parameter number (FBnr.)
8	String length		For parameter type ‘string’ this field contains the expected string length.	
9	\r		Carriage Return	
10	\n		Line Feed	

Advise:
Use the same process name for Nr 4 and 6. If no index is required, fill in the FBnr. so the return message will return with the requested parameter number.

Answer to the request				
Nr	**Byte**	**Layout**	**Description**	
0	:		start character	
1	Length		Message length	
2	Node		Node address	
3	02		Command write	
4*	Process	cppppppp	c	Process chained
			p	Process number
5*	Parameter	cttnnnnn	c	Parameter chained
			t	Parameter type
			n	Parameter index 0…31
6	Value 1		Value for all types. For ‘strings’ this field contains the string length.	
7	Value 2		Value for type ‘integer’, ‘float’ or ‘long’.	
8	Value 3		Value for type ‘float’ or ‘long’.	
9	Value 4		Value for type ‘float’ or ‘long’.	
X	Value x		More value fields follow for type ‘string’ depending on string length. If given string length is zero, the final field should also contain a zero.	
X+1	\r		Carriage Return	
X+2	\n		Line Feed	

** The requested device copies these values from the request message directly into the answer message.*

3.9 EXAMPLES – PROPAR ASCII PROTOCOL

3.9.1 Sending setpoint

Send setpoint = 50% to node 3 process 1. Setpoint values should be given in a range from 0 to 32000 so for this example 16000 should be send.

Send setpoint = 50% to node 3 process 1 parameter 1						
Nr	**Byte**	**Layout**	**Description**			**Description per block**
0	:		Start character			Start
1	06		Length 6			
2	03		Node 3			
3	01		Command write with status response			
4	01	cppppppp 00000001	c	00	Process not chained	Process 1
			p	01	Process 1	
5	21	cttppppp 00100001	c	00	Parameter not chained	Parameter number 1 Setpoint
			t	20	Parameter type ‘integer’	
			p	01	Parameter number (FBnr.) 1	
6	3E		Setpoint 16000 = 3E80h			
7	80					
8	\r		Carriage Return			End
9	\n		Line Feed			
Total String			:06030101213E80\r\n			

Answer from node 3			
Nr	**Byte**	**Description**	**Description per block**
0	:	Start character	Start
1	04	Fixed message length 4.	
2	03	Node address 03	
3	00	Command status	
4	00	Status ok.	Status answer
5	05	Status ok, value points to end of send message.	
6	\r	Carriage Return	End
7	\n	Line Feed	
Total String		:0403000005\r\n	

Position of nr 5 is as follows.

String	:	06	03	01	01	21	3E	80	\r\n
Position			1	2	3	4	5	6	

3.9.2 Request setpoint

Request setpoint from node 3 process 1, type integer.						
Nr	**Byte**	**Layout**	**Description**			**Description per block**
0	:					Start
1	06		Length 6			
2	03		Node 3			
3	04		Command read			
4	01	00000001	C	00	Process not chained (return)	Process 1
			P	01	Process 1 (return)	
5	21	00100001	C	00	Parameter not chained (return)	Parameter index 1 Process 1 Parameter number 1 Setpoint
			T	20	Parameter type ‘integer’ (return)	
			N	01	Parameter index 1 (return)	
6	01	00000001	P	01	Process 1	
7	21	00100001	T	20	Parameter type ‘integer’	
			P	01	Parameter number (FBnr.) 1 (setpoint)	
8	\r				Carriage Return	End
9	\n				Line Feed	
Total String			:06030401210121\r\n			

Answer from node 3 process 1, type integer.						
Nr	**Byte**	**Layout**	**Description**			**Description per block**
0	:					Start
1	06		Length 6			
2	03		Node 3			
3	02		Command write			
4	01	00000001	C	00	Process not chained	Process
			P	01	Process 1 (receiving process)	
5	21	00100001	C	00	Parameter not chained	Parameter index 1 Setpoint
			T	20	Parameter type ‘integer’	
			N	01	Parameter index 1	
6	3E		Value 3E80h = 16000 = 50%			
7	80					
8	\r		Carriage Return			End
9	\n		Line Feed			
Total String			:06030201213E80\r\n			

ASCII character “SPACE” (20 HEX) is returned for every empty space in the string.

3.9.3 Request measure

Request measure from node 3 process 1, type integer						
Nr	**Byte**	**Layout**	**Description**			**Description per block**
0	:					Start
1	06		Length 6			
2	03		Node 3			
3	04		Command read			
4	01	00000001	C	00	Process not chained (return)	Process 1
			P	01	Process 1 (return)	
5	21	00100001	C	00	Parameter not chained (return)	Parameter type 1 Process 1 Parameter number 0 Measure
			T	20	Parameter type ‘integer’ (return)	
			N	01	Parameter index 1 (return)	
6	01	-0000001	P	01	Process 1	
7	20	-0100000	T	20	Parameter type ‘integer’	
			P	00	Parameter number (FBnr.) 0 (measure)	
8	\r				Carriage Return	End
9	\n				Line Feed	
Total String			:06030401210120\r\n			

Answer from node 3 process 1						
Nr	**Byte**	**Layout**	**Description**			**Description per block**
0	:					Start
1	06		Length 6			
2	03		Node 3			
3	02		Command write			
4	01	00000001	C	00	Process not chained	Process 1
			P	01	Process 1 (receiving process)	
5	21	00100001	C	00	Parameter not chained	Parameter index 1 measure
			T	20	Parameter type ‘integer’	
			N	01	Parameter index 1	
6	3E		Value 3E80h = 16000 = 50%			
7	80					
8	\r		Carriage Return			End
9	\n		Line Feed			
Total String			:06030201213E80\r\n			

3.9.4 Request counter value

Request Counter value from node 3, process 104, type float						
Nr	**Byte**	**Layout**	**Description**			**Description per block**
0	:					Start
1	06		Length 6			
2	03		Node 3			
3	04		Command read			
4	68	01101000	C	00	Process not chained (return)	Process 104
			P	68	Process 104 (return)	
5	41	01000001	C	00	Parameter not chained (return)	Parameter index 1 Process 104 Parameter number 1 Counter value
			T	40	Parameter type ‘float’ (return)	
			N	01	Parameter index 1 (return)	
6	68	01101000	P	68	Process 104	
7	41	01000001	T	40	Parameter type ‘float’	
			P	01	Parameter number (FBnr.) 1 (counter value)	
8	\r				Carriage Return	End
9	\n				Line Feed	
Total String			:06030468416841\r\n			

Answer from node 3, process 104						
Nr	**Byte**	**Layout**	**Description**			**Description per block**
0	:					Start
1	08		Length 8			
2	03		Node 3			
3	02		Command write			
4	68	01101000	C	00	Process not chained	Process 104
			P	68	Process 104 (receiving process)	
5	41	01000001	C	00	Parameter not chained	Parameter index 1 Counter value
			T	40	Parameter type ‘float’	
			N	01	Parameter index 1	
6	45		Parameter value ‘float’ = 5023.96 decimal			
7	9C					
8	FF					
9	AE					
10	\r		Carriage Return			End
11	\n		Line Feed			
Total String			:0803026841459CFFAE\r\n			

3.9.5 Collection of ProPar ASCII examples

hexadecimal 7D00 = decimal 32000
hexadecimal 3E80 = decimal 16000
Float numbers are in 32-bit Single-precision floating-point format(IEEE-754), e.g. float 3F800000=dec 1
Strings contains ASCII characters e.g. hex 41 is character A, hex 4D = M, hex 6D = m , etc

It is important to know that not all parameters are available on all ProPar/Multibus devices.
For more details about parameters and their use see also document nr. 9.17.023 for description of digital instruments. If you have the program FlowDDE, you can also get an overview of which parameters are available on which devices.

NORMAL OPERATION PARAMETERS

measure: read, Process: 1, Parameter: 0, Type: integer
Request :06800401210120\r\n
Answer(example) :06800201217D00\\r\n
measure = hex 7D00 = 32000 = 100%

fmeasure: read, Process: 33, Parameter: 0, Type: float
Request :06800421402140\r\n
Answer(example :0803022140453B8000\r\n
fmeasure = 453B8000 = dec 3000

setpoint: read, Process: 1, Parameter: 1, Type: integer
Request :06800401210121\r\n
Answer(example) :06800201217D00\r\n
measure = hex 7D00 = 32000 = 100%

setpoint: write value = hex 7D00 = 32000 = 100% ,Process: 1, Parameter: 1, Type: integer
Send :06800101217D00\r\n
Answer(example) :0480000005\r\n
Status : 00 (No error)

setpoint: write value = hex 3E80 = 16000 = 50% ,Process: 1, Parameter: 1, Type: integer
Send :06800101213E80\r\n
Answer(example :0480000005\r\n
Status : 00 (No error)

setpoint: write value = 0**,** Process: 1, Parameter: 1, Type: integer
Send :06800101210000\r\n
Answer(example :0480000005\r\n
Status : 00 (No error)

fsetpoint: read, Process: 33, Parameter: 3, Type: float
Request :06800421412143\r\n
Answer(example) :0880022141453B8000\r\n
fsetpoint = 453B8000 = dec 3000

fsetpoint: write value = float 3F800000 = dec 1**,** Process: 33, Parameter: 3, Type: float
Send :08800121433F800000\r\n
Answer(example) :0480000007\r\n
Status : 00 (No error)

Control mode: read , Process: 1, Parameter: 4, Type: character
Send :06800401040104\r\n
Answer(example) :058002010401\r\n
Value = 01 (Control mode = "Analog input")

Control mode: write value = 0 , Process: 1, Parameter: 4, Type: character
Send :058001010400\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

Control mode: write value = 1 **,** Process: 1, Parameter: 4, Type: character
Send :058001010401\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

Control mode: write value = 18 = hex 12 **,** Process: 1, Parameter: 4, Type: character
Send :058001010412\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

Slave factor: read, Process: 33, Parameter: 1, Type: float
Request :06800421412141\r\n
Answer(example) :088002214142C80000\r\n
Slave factor = 453B8000 = dec 100 (values in percentage so 100%)

Slave factor: write value = 40A00000 = dec 5 , Process: 33, Parameter: 1, Type: float
Send :088001214140A00000\r\n
Answer(example) :0480000007\r\n
Status : 00 (No error)

Slave factor: write value = 42C80000= dec 100 , Process: 33, Parameter: 1, Type: float
Send :088001214142C80000\r\n
Answer(example) :0480000007\r\n
Status : 00 (No error)

Slave factor: write value = 43480000= dec 200 , Process: 33, Parameter: 1, Type: float
Request :088001214143480000\r\n
Answer(example) :0480000007\r\n
Status : 00 (No error)

Fluid number: read , Process: 1, Parameter: 16, Type: character
Request :06800401100110\r\n
Answer(example) :058002011000\r\n
Value = 00 (Fluid number = 1)

Fluid number: write value = 0 (Fluid number = 1) , Process: 1, Parameter: 16, Type: character
Send :058001011000\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

Fluid number: write value = 1 (Fluid number = 2) , Process: 1, Parameter: 16, Type: character
Send :058001011001\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

fluidname: read, Process: 1, Parameter: 17, Type: string [10]
Request :078004017101710A\r\n
Answer(example) :0F800201710A41695220202020202020\r\n
Fluidname = 41695220202020202020 = AiR

Valve output: read, Process: 114, Parameter: 1, Type: long
Request :06800472417241\r\n
Answer(example) :080302724100000000\r\n
Valve output = 00000000 = 0%

Valve output: read, Process: 114, Parameter: 1, Type: long
Request :06800472417241\r\n
Answer(example) :0803027241009DDDDD\r\n
Valve output = 009DDDDD = dec 10345949 = 61.7% (this is the typical maximum valveout)

temperature: read, Process: 33, Parameter: 7, Type: float
Request :06800421472147\r\n
Answer(example) :088002214700000000\r\n
temperature = 00000000= dec 0 (0 Degr. C, temperature measurement is not available for this type of instrument)

temperature: read, Process: 33, Parameter: 7, Type: float
Request : 06800421472147\r\n
Answer(example) :088002214741FE4FBF\r\n
temperature = 41FE4FBF = dec 31.788939 Degr. C

Actual Density: read Process: 116, Parameter: 15, Type: float
Request :068004744F744F\r\n
Answer(example) :088002744F447A0000\r\n
Actual Density = 447A0000 = dec 1000

capacity100%: read, Process: 1, Parameter: 13, Type: float
Request :068004014D014D\r\n
Answer(example) :088002014D40000000\r\n
Capacity100% = 40000000 = dec 2

capacity unit: read, Process: 1, Parameter: 31, Type: string [7]
Request :078004017F017F07\r\n
Answer(example) :0C8002017F076B672F68202020\r\n
Capacity unit = 6B672F68202020 = kg/h

ALARM / STATUS PARAMETERS

alarm info: read , Process: 1, Parameter: 20, Type: character
Request :06800401140114\r\n
Answer(example) :058002011400\r\n
Value = 00

alarm mode: read , Process: 97, Parameter: 3, Type: character
Request :06800461036103\r\n
Answer(example) :058002610301\r\n
Value = 01

alarm mode: write value = 0 , Process: 97, Parameter: 3, Type: character
Send :058001610300\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

alarm mode: write value = 1, Process: 97, Parameter: 3, Type: character
Send :058001610301\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

Alarm maximum limit: read, Process: 97, Parameter: 1, Type: integer
Request :06800461216121\r\n
Answer(example) :06800261215DC0\r\n
Alarm maximum limit = 5DC0 = dec 24000 = 75%

Alarm maximum limit: write value = 7D00 = dec 32000 , Process: 97, Parameter: 1, Type: integer
Send :06800161217D00\r\n
Answer(example) :0480000005\r\n
Status : 00 (No error)

Alarm minimum limit: read, Process: 97, Parameter: 2, Type: integer
Request :06800461226122\r\n
Answer(example) :06800261211F40\r\n
Alarm maximum limit = 1F40 = dec 8000 = 25%

Alarm minimum limit: write value = 1F40 = dec 8000 , Process: 97, Parameter: 2, Type: integer
Send :06800161211F40\r\n
Answer(example) :0480000005\r\n
Status : 00 (No error)

alarm setpoint mode: read, Process: 97, Parameter: 5, Type: character
Send :06800461056105\r\n
Answer(example) :058002610500\r\n
alarm setpoint mode = 00

alarm setpoint mode: write value = 0 **,** Process: 97, Parameter: 5, Type: character
Send :058001610500\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

alarm setpoint mode: write value = 1 **,** Process: 97, Parameter: 5, Type: character
Send :058001610501\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

Alarm new setpoint: read, Process: 97, Parameter: 6, Type: integer
Request :06800461266126\r\n
Answer(example) :068002612600000\r\n
Alarm new setpoint = 0000 = dec 0 = 0%

Alarm new setpoint: write value = 140 = dec 320 = 10%, Process: 97, Parameter: 6, Type: integer
Send :06800161260140\r\n
Answer(example) :0480000005\r\n
Status : 00 (No error)

Alarm new setpoint: write value = 0 = dec 0 = 0%, Process: 97, Parameter: 6, Type: integer
Send :06800161260000\r\n
Answer(example) :0480000005\r\n
Status : 00 (No error)

alarm delay time: read , Process: 97, Parameter: 7, Type: character
Request :06800461076107\r\n
Answer(example) :058002610703\r\n
Value = 03 (3 seconds)

alarm delay time: write value = 0 **,** Process: 97, Parameter: 7, Type: character
Send :058001610700\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

alarm delay time: write value = 3 (seconds) **,** Process: 97, Parameter: 7, Type: character
Send :058001610703\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

Reset alarm enable: read , Process: 97, Parameter: 9, Type: character
Request :06800461096109\r\n
Answer(example) :058002610900F\r\n
Value = 0F

Reset alarm enable: write value = 0F = dec 15 **,** Process: 97, Parameter: 9, Type: character
Send :058001610900F\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

COUNTER PARAMETERS

Counter value: read, Process: 104, Parameter: 1, Type: float
Request :06800468416841\r\n
Answer(example) :0880026841444A6E18\r\n
Counter value = 444A6E18 = dec 809.72021

Counter mode: read , Process: 104, Parameter: 8, Type: character
Request :06800468086808\r\n
Answer(example) :058002680801\r\n
Value = 01

Counter mode: write value = 0 **,** Process: 104, Parameter: 8, Type: character
Send :058001680800\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

Counter mode: write value = 2 **,** Process: 104, Parameter: 8, Type: character
Send :058001680802\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

Counter setpoint mode: read , Process: 104, Parameter: 5, Type: character
Request :06800468056805\r\n
Answer(example) :058002680500\r\n
Value = 00

Counter setpoint mode: write value = 1 **,** Process: 104, Parameter: 5, Type: character
Send :058001680501\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

counter new setpoint: write value = 0 , Process: 104, Parameter: 6, Type: integer
Send :06800168260000\r\n
Answer(example) :0480000005\r\n
Status : 00 (No error)

counter new setpoint: write value = hex 140 = dec 320 ,Process: 104, Parameter: 6, Type: integer
Send :06800168260140\r\n
Answer(example) :0480000005\r\n
Status : 00 (No error)

counter limit: read, Process: 104, Parameter: 3, Type: float
Request :06800468436843\r\n
Answer(example) :088002684343FA0000\r\n
fmeasure= 43FA0000 = dec 500

counter limit: write value 3F800000 = dec 1, Process: 104, Parameter: 3, Type: float
Send :08800168433F800000\r\n
Answer(example) :0480000007\r\n
Status : 00 (No error)

counter unit index: read , Process: 104, Parameter: 2, Type: character
Request :06800468026802\r\n
Answer(example) :0580026805 00\r\n
Value = 00

counter unit index: write value = 0 **,** Process: 104, Parameter: 2, Type: character
Send :058001680500\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

Counter unit: read, Process: 104, Parameter:7, Type: string [4]
Request :0780046867686704\r\n
Answer(example) :0980026867046D6C6E20\r\n
Capacity unit = 6D6C6E20 = mln

Counter unit: write value = mln**,** Process: 104, Parameter:7, Type: string [4]
Send :0980016867046D6C6E20\r\n
Answer(example) :0980026867046D6C6E20\r\n
Capacity unit = 6D6C6E20 = mln

Reset counter enable: read , Process: 104, Parameter: 9, Type: character
Request :06800468096809\r\n
Answer(example) :0580026809 07\r\n
Value = 07

Reset counter enable: write value = 0F = dec 15 **,** Process: 104, Parameter: 9, Type: character
Send :05800168090F\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

Counter controller overrun correction : read, Process: 104, Parameter: 10, Type: float
Request :068004684A684A\r\n
Answer(example :088002684A00000000\r\n
Value = 00000000 = dec 0

Counter controller overrun correction: write value = float 00000000 = dec 0**,** Process: 104, Parameter: 10, Type: float
Send :088001684A00000000\r\n
Answer(example) :0480000007\r\n
Status : 00 (No error)

Counter controller overrun correction: write = float 3F4CCCCD= dec 0.8**,** Process: 104, Parameter: 10, Type: float
Send :088001684A3F4CCCCD\r\n
Answer(example) :0480000007\r\n
Status : 00 (No error)

Counter controller gain: read, Process: 104, Parameter: 11, Type: float
Request :068004684B684B\r\n
Answer(example) :088002684B00000000\r\n
Value = 00000000 = dec 0

Counter controller gain: write value = float 00000000 = dec 0**,** Process: 104, Parameter: 11, Type: float
Send :088001684B00000000\r\n
Answer(example) :04800000 07\r\n
Status : 00 (No error)

Counter controller gain: write value = float 41200000 = dec 10**,** Process: 104, Parameter: 11, Type: float
Send :088001684B41200000\r\n
Answer(example) :04800000 07\r\n
Status : 00 (No error)

IDENTIFICATION PARAMETERS

serial number: read, Process: 113, Parameter: 3, Type: string
Request :0780047163716300\r\n
Answer(example) :1080027163004D31353231303633344100\r\n
serial number = 4D313532313036333441 = M15210634A

bhtmodel number: read, Process: 113, Parameter: 2, Type: string
Request :0703047162716200\r\n
Answer(example) :1A0302716200462D32303143562D354B302D4141442D33332D5600\r\n
bhtmodel number = 462D32303143562D354B302D4141442D33332D56 = F-201CV-5K0-AAD-33-V

firmware version: read, Process: 113, Parameter: 5, Type: string [6]
Request :0780047165716506\r\n
Answer(example) :0B800271650656382E333700\r\n
Firmware = 56382E333700 = (ASCII characters) V8.37

usertag: Process: 113, Parameter: 6, Type: string
Request :0703047166716600\r\n
Answer(example) :0D80027166005553455254414700\r\n
usertag = 55534552544147 = (ASCII characters) USERTAG

customer model: read, Process: 113, Parameter: 4, Type: string
Request :0703047164716400\r\n
Answer(example) :0E03027164005354414E4441524400\r\n
Value = 5354414E4441524400 = (ASCII characters) STANDARD

device type: read, Process: 113, Parameter: 5, Type: string [6]
request :0780047161716106\r\n
Answer(example) :0B8002716106434F52494643\r\n
bhtmodel number = 434F52494643 = (ASCII characters) CORIFC

SPECIAL PARAMETERS

reset: write value = 0, Process: 115, Parameter: 8, Type: character
Send :058001730800\r\n
Answer(example) :04800000 04\r\n
Status : 00 (No error)

reset: write value = 3, Process: 115, Parameter: 8, Type: character
Send :058001730803\r\n
Answer(example) :04800000 04\r\n
Status : 00 (No error)

initreset: read , Process: 0, Parameter: 10, Type: character
Request :068004000A000A\r\n
Answer(example) :058002000A52\r\n
Value = 52 = dec 82

initreset: write value = 40 = dec 64**,** Process: 0, Parameter: 10, Type: character
Send :058001000A40\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

initreset: write value = 52 = dec 82, Process: 0, Parameter: 10, Type: character
Send :058001000A52\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

wink: write value = 39 (= character 9), Process: 0, Parameter: 0, Type: character
Send :06800100600139\r\n
Answer(example) :0480000005\r\n
Status : 00 (No error)

iostatus: read , Process: 114, Parameter: 11, Type: character
Request :068004720B720B\r\n
Answer(example) :058002720B4F\r\n
Value = 4F = dec 79

iostatus: write value = 0F = dec 15, Process: 114, Parameter: 11, Type: character
Send :058001720B0F\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

iostatus: write value = 4F = dec 79, Process: 114, Parameter: 11, Type: character
Send :058001720B4F\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

iostatus: write value = 07 = dec 07, Process: 114, Parameter: 11, Type: character
Send :058001720B07\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

SPECIAL INSTRUMENT FEATURES - ZEROING

initreset: write value = 40 = dec 64**,** Process: 0, Parameter: 10, Type: character
Send :058001000A40\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

initreset: read , Process: 0, Parameter: 10, Type: character
Request :068004000A000A\r\n
Answer(example) :058002000A52\r\n
Value = 52 = dec 82

Control mode: read , Process: 1, Parameter: 4, Type: character
Send :06800401040104\r\n
Answer(example) :0580020104**00**\r\n
Value = 00 (Control mode = "BUS / RS232")

initreset: write value = 00 = dec 00**,** Process: 0, Parameter: 10, Type: character
Send :058001000A00\r\n
Answer(example) :0480000004\r\n
Status : 00 (No error)

Control mode: write value = 9 **,** Process: 1, Parameter: 4, Type: character
Send :050301010409\r\n
Answer(example) :0403000004\r\n
Status : 00 (No error)

calibration mode: read , Process: 115, Parameter: 1, Type: character
Send :06800473017301\r\n
Answer(example) :058002010409\r\n
Value = 09

calibration mode: write value = FF = dec 255 **,** Process: 115, Parameter: 1, Type: character
Send :0580017301FF\r\n
Answer(example) :0403000004\r\n
Status : 00 (No error)

calibration mode: write value = 0 **,** Process: 115, Parameter: 1, Type: character
Send :058001730100\r\n
Answer(example) :0403000004\r\n
Status : 00 (No error)

calibration mode: write value = 9 **,** Process: 115, Parameter: 1, Type: character
Send :058001730109\r\n
Answer(example) :0403000004\r\n
Status : 00 (No error)

3.9.6 Request chained parameters

The Interface sends a request for the following parameters to device at node 3:
Process 113: Serial number (3), USERTAG (6)
Process 1: Measure (0), Capacity (13), Capacity unit (31), Fluid name (17)

Request chained parameters to node 3						
Nr	**Byte**	**Layout**	**Description**			**Description per block**
0	:					Start
1	1A		Length 26			
2	03		Node 3			
3	04		Command read			
4	F1	11110001	C	80	Process chained (return)	Process 113 (return)
			P	71	Process 113 (return)	
5	EC	11101100	C	80	Parameter chained (return)	Parameter index 12 (return) Process 113 Parameter number 3 Serial number
			T	60	Parameter type ‘string’ (return)	
			N	0C	Parameter index 12 (return)	
6	71	01110001	P	71	Process 113	
7	63	01100011	T	60	Parameter type ‘string’	
			P	03	Parameter number (FBnr.) 3 – Serial number	
8	00	00000000		00	String length 00, length not defined	
9	6D	01101101	C	00	Parameter not chained (return)	Parameter index 13 (return) Process 113 Parameter number 6 USERTAG
			T	60	Parameter type ‘string’ (return)	
			N	0D	Parameter index 13 (return)	
10	71	01110001	P	71	Process 113	
11	66	01100110	T	60	Parameter type ‘string’	
			P	06	Parameter number (FBnr.) 6 – USERTAG	
12	00	00000000		00	String length 00, length not defined	
13	01	00000001	C	00	Parameter not chained (return)	Process 1 (return)
			P	01	Process 1 (return)	
14	AE	10101110	C	80	Parameter chained (return)	Parameter index 14 (return) Process 1 Parameter number 0 Measure
			T	20	Parameter type ‘integer’ (return)	
			N	0E	Parameter index 14 (return)	
15	01	00000001	P	00	Process 1	
16	20	00100000	T	20	Parameter type ‘integer’	
			P	00	Parameter number (FBnr.) 0 – Measure	
17	CF	11001111	C	80	Process chained (return)	Parameter index 15 (return) Process 1 Parameter number 15 Capacity
			T	40	Parameter type ‘float’ (return)	
			N	0F	Parameter index 15 (return)	
18	01	00000001	P	01	Process 1	
19	4D	01001101	T	40	Parameter type ‘float’	
			P	0D	Parameter number (FBnr.) 13 – Capacity	
20	F0	11110000	C	80	Parameter chained (return)	Parameter index 16 (return) Process 1 Parameter number 31 Capacity unit
			T	60	Parameter type ‘string’ (return)	
			N	10	Parameter index 16 (return)	
21	01	00000001	P	01	Process 1	
22	7F	01111111	T	60	Parameter type ‘string’	
			P	1F	Parameter number (FBnr.) 31 – Capacity unit	
23	07	00001110		07	String length 7	
24	71	01110001	C	00	Parameter not chained (return)	Parameter index 17 (return) Process 1
			T	60	Parameter type ‘string’ (return)	

			N	11	Parameter index 17 (return)	Parameter number 17 Fluid name
25	01	00000001	P	01	Process 1	
26	71	01110001	T	60	Parameter type 'string'	
			P	11	Parameter number (FBnr.) 17 – Fluid name	
27	0A				String length 10	
29	\n				Line Feed	
28	\r				Carriage Return	End
Total String			:1A0304F1EC7163006D71660001AE0120CF014DF0017F077101710A\r\n			

Answer by node 3						
Nr	**Byte**	**Layout**	**Description**			**Description per block**
0	:					Start
1	37		Number of bytes which do follow: 65 bytes			
2	03		Node 3			
3	02		Command write			
4	F1	11110001	C	80	Process chained	Process 113
			P	71	Process 113 (receiving process)	
5	EC	11101100	C	80	Parameter chained	Parameter index 12 Serial number
			T	60	Parameter type 'string'	
			N	0C	Parameter index 12	
6	00		Length of the answer 10 Bytes			
7-26			4D 36 32 31 32 33 34 35 41 00 Parameter value converted from hex to ASCII : M6212345A			
27	6D	01101101	C	00	Process not chained	Parameter index 13 USERTAG
			T	60	Parameter type 'string'	
			N	0D	Parameter index 13	
28	00		String length 00, length not defined			
29-36			55 53 45 52 54 41 47 00 Parameter value converted from hex to ASCII, the values do read : USERTAG			
37	01	00000001	C	00	Process not chained	Process 1
			P	01	Process 1 (receiving process)	
38	AE	10101110	C	80	Parameter chained	Parameter index 14 Measure
			T	20	Parameter type 'integer'	
			N	0E	Parameter index 14	
39	1C		Parameter value is: 1CD8 (hex) Measure Value is: 7384 (dec)			
40	D8					
41	CF	11001111	C	80	Parameter chained	Parameter index 15 Capacity
			T	40	Parameter type 'float'	
			N	0F	Parameter index 15	
42	3F		3F 80 00 00 Parameter Value in IEEE-floating point notation, 32-bit single precision Parameter value converted from float to decimal, the values reads : 1.0			
43	80					
44	00					
45	00					
46	F0	11110000	C	80	Parameter chained	Parameter index 16 Capacity unit
			T	60	Parameter type 'string	
			N	10	Parameter index 16	
47	07		Length of the answer 7 Bytes			
48-54			6D 6C 6E 2F 6D 69 6E Parameter value converted from hex to ASCII, the values do read : mln/min			

55	71	01110001	C	00	Parameter not chained	Parameter index 17 Fluid name
			T	60	Parameter type 'string'	
			N	11	Parameter index 17	
56	0A		Length of the answer 10 Bytes			
57-66			4E 32 20 20 20 20 20 20 20 20 Parameter value converted from hex to ASCII, the values do read: N2			
Total String			:370302F1EC004D3632313233343541006D0055534552544147000 1AE1CD8CF3F800000F0076D6C6E2F6D696E710A4E322020202020202020\r\n			

3.9.7 Example Request chained parameters, setpoint and measure

setpoint: Process: 1, Parameter: 1, Type: integer
measure: Process: 1, Parameter: 0, Type: integer

Chained Request to read setpoint and measure:
Request :0A800481210121012101 20\r\n
Answer(example) :0A800281213E8001213E80\r\n

setpoint = 3E80 = dec. 16000 = 50%
measure = 3E80 = dec. 16000= 50%

3.9.8 Example Request chained parameters, measure and temperature

measure : Process: 1, Parameter: 0, Type: integer
temperature : Process: 33, Parameter: 7, Type: float

Chained Request to read measure and temperature:
Request :0A8004812101202147 2147\r\n
Answer(example) :0C800281213E80214742033089\r\n

measure = 3E80 = dec. 16000 = 50%
temperature = 42033089 = dec. 32.797398 (Degr.C)

Note: Temperature parameter value is in IEEE-floating point notation, 32-bit single precision

3.9.9 Example Request chained parameters, fmeasure and temperature

fmeasure : Process: 33, Parameter: 0, Type: float
temperature : Process: 33, Parameter: 7, Type: float

Chained Request to read fmeasure and temperature:
Request :0A8004A14021402no1472147\r\n
Answer(example) :0E8002A14041000000214741F30956\r\n

fmeasure = 41000000 = dec. 8
temperature = 41F30956 = dec. 30.379559 (Degr.C)

3.10 EXAMPLES - PROPAR ENHANCED BINARY PROTOCOL

3.10.1 Sending setpoint - ProPar Enhanced binary protocol

Send setpoint = 50% to node 3 process: 1, parameter: 1, type: integer
Setpoint values should be given in a range from 0 to 32000 so for this example 16000 should be send.

Send setpoint = 50% to node 3 process 1 parameter 1						
Nr	**Byte**	**Layout**	**Description**			**Description per block**
0	10		DLE, STX			start sequence
1	02					
3	01		message sequence number			sequence number
4	03		node address of destination			node address
5	05		length of data field in bytes			length
6	01		Send parameter with destination address, will be answered with type 00 command			Communication command
7	01	cppppppp 00000001	c	00	Process not chained	Process number
			p	01	Process 1	
8	21	cttppppp 00100001	c	00	Parameter not chained	Parameter number
			t	20	Parameter type ‘integer’	
			p	01	Parameter number (FBnr.) 1	
9	3E		Setpoint 16000 = 3E80h = 50%			message field
10	80					
11	10		DLE, ETX			end sequence
12	03					
Total String			10020103050101213E801003			

Answer from node 3			
Nr	**Byte**	**Description**	**Description per block**
0	10	DLE, STX	start sequence
1	02		
2	01	message sequence number, as in request	sequence number
3	03	node address of source, as in request	node address
4	03	length of data field in bytes	length
5	00	Status message	Communication command
6	00	Status ok.	message field
7	05	Index pointing to the first byte in the send message for which the above status applies. Value from this byte may be neglected if value of byte 6 = 0	
8	10	DLE, ETX	end sequence
9	03		
Total String		10020103030000051003	

3.10.2 Request setpoint - ProPar enhanced binary protocol

Read setpoint node 3 process 1 parameter 1						
Nr	**Byte**	**Layout**	**Description**			**Description per block**
0	10		DLE, STX			start sequence
1	02					
2	01		message sequence number			sequence number
3	03		node address of destination			node address
4	05		length of data field in bytes			length
5	04		Request parameter, will be answered with type 02 or 00 command			Communication command
6	01	cppppppp 00000001	c	00	Process not chained (return)	Process number
			p	01	Process 1 (return)	
7	21	cttnnnnn 00100001	c	00	Parameter not chained (return)	Parameter number
			t	20	Parameter type ‘integer’ (return)	
			n	01	Parameter index 1 (return)	
8	01	-ppppppp 00000001	P	01	Process 1	Process number
9	21	-ttppppp 00100001	t	20	Parameter type ‘integer’	Parameter number
			p	01	Parameter number (FBnr.) 1 (setpoint)	
10	10		DLE, ETX			end sequence
11	03					
Total String			10020103050401210121 1003			

Answer from node 3						
Nr	**Byte**	**Layout**	**Description**			**Description per block**
0	10		DLE, STX			start sequence
1	02					
2	01		message sequence number, as in request			sequence number
3	03		node address of source, as in request			node address
4	05		length of data field in bytes			length
5	02		Send parameter with destination address, no status requested			Communication command
6	01	cppppppp 00000001	c	00	Process not chained	Process
			p	01	Process 1 (receiving process)	
7	21	cttnnnnn 00100001	c	00	Parameter not chained	Parameter index 1
			t	20	Parameter type ‘integer’	
			n	01	Parameter index 1	
8	7D		Value 7D00h = 32000 = 100%			Setpoint
9	00					
10	10		DLE, ETX			end sequence
11	03					
Total String			10020103050201217D001003			

3.10.3 Request measure - ProPar enhanced binary protocol

Read measure node 3 process 1, parameter 0, type integer						
Nr	**Byte**	**Layout**	**Description**			**Description per block**
0	10		DLE, STX			start sequence
1	02					
3	01		message sequence number			sequence number
4	03		node address of destination			node address
5	05		length of data field in bytes			length
6	04		Request parameter, will be answered with type 02 or 00 command			Communication command
7	01	cppppppp 00000001	c	00	Process not chained (return)	Process number
			p	01	Process 1 (return)	
8	21	cttnnnnn 00100001	c	00	Parameter not chained (return)	Parameter number
			t	20	Parameter type ‘integer’ (return)	
			n	01	Parameter index 1 (return)	
9	01	-ppppppp 00000001	p	01	Process 1	Process number
10	20	-ttppppp 00100000	t	20	Parameter type ‘integer’	Parameter number
			p	00	Parameter number (FBnr.) 0 (measure)	
11	10		DLE, ETX			end sequence
12	03					
Total String			10020103050401210120 1003			

Answer from node 3						
Nr	**Byte**	**Layout**	**Description**			**Description per block**
0	10		DLE, STX			start sequence
1	02					
2	01		message sequence number, as in request			sequence number
3	03		node address of source, as in request			node address
4	05		length of data field in bytes			length
5	02		Send parameter with destination address, no status requested			Communication command
6	01	cppppppp 00000001	c	00	Process not chained	Process
			p	01	Process 1 (receiving process)	
7	21	cttnnnnn 00100001	c	00	Parameter not chained	Parameter index 1
			t	20	Parameter type ‘integer’	
			n	01	Parameter index 1	
8	7D		Value 7D00h = 32000 = 100%			Measure
9	00					
10	10		DLE, ETX			end sequence
11	03					
Total String			10020103050201217D001003			

3.10.4 Collection of ProPar enhanced binary examples

NORMAL OPERATION PARAMETERS

measure: read, Process: 1, Parameter: 1, Type: integer
Request 10020180050401210120 1003
Answer(example) 1002018005010121 7D00 1003
measure = hex 7D00 = 32000 = 100%

fmeasure: read, Process: 33, Parameter: 0, Type: float
Request 10020180050421402140 1003
Answer(example) 10020180070221404170000001003
fmeasure = 41700000 = dec 15

setpoint: read, Process: 1, Parameter: 1, Type: integer
Request 10020180050401210121 1003
Answer(example) 1002018005020121 7D00 1003
measure = hex 7D00 = 32000 = 100%

setpoint: write value = hex 7D00 = 32000 = 100% ,Process: 1, Parameter: 1, Type: integer
Send 1002018005010121 7D00 1003
Answer(example) 100201800300 00 051003
Status : 00 (No error)

setpoint: write value = hex 3E80 = 16000 = 50% ,Process: 1, Parameter: 1, Type: integer
Send 1002018005010121 3E80 1003
Answer(example) 100201800300 00 051003
Status : 00 (No error)

setpoint: write value = 0**,** Process: 1, Parameter: 1, Type: integer
Send 1002018005010121 0000 1003
Answer(example) 100201800300 00 051003
Status : 00 (No error)

fsetpoint: read, Process: 33, Parameter: 3, Type: float
Request 10020180050421412143 1003
Answer(example) 1002018007022141 40F00000 1003
fmeasure = 40F00000 = dec 7.5

fsetpoint: write value = float 3F800000 = dec 1**,** Process: 33, Parameter: 3, Type: float
Send 1002018007012143 3F800000 1003
Answer(example) 100201800300 00 071003
Status : 00 (No error)

3.10.5 Example chained parameters - ProPar enhanced binary protocol

measure: Process: 1, Parameter: 0, Type: integer
setpoint: Process: 1, Parameter: 1, Type: integer

Individual request for Measure and Setpoint from node 3 :
Read Measure 1002 01030504 01210120 1003
Read Setpoint 1002 01030504 01210121 1003

Chained request for Measure and Setpoint:
Note; the chain bit of the process number is set to indicate another process will follow.
Request 1002 01800904 **81**210120 01210121 1003
↑ ↑
Measure Setpoint

Anwer: 10020180090281213E8001213E801003

measure = 3E80 = dec. 16000 = 50%
setpoint = 3E80 = dec. 16000 = 50%

3.10.6 Examples data byte 0x10 – ProPar enchanced binary protocol

The need for the extra DLE 0x10 byte is explained in the chapter “ProPar enhanced binary protocol”.

The extra 0x10 is a must for all data bytes and should not be taken into account for the ‘length of data field in bytes’.

Examples using values with a byte 0x10 :

Hex	Dec
1003	4099
1010	4112
10	16

Example 1	**Write Setpoint 0x1003**
Request	extra 0x10 byte ↓ 10020103 05 0101210101003 1003 ↑ ↑ length=5 nr of data bytes = 6
Answer	10020103030000051003

Example 2	**Read Setpoint 0x1003**
Request	10020103050401210121003
Answer	10020103 05 020121101003 1003 ↑ ↑ length=5 nr of data bytes = 6

Example 3	**Write Setpoint 0x1010**
Request	10020103 05 01012110101010 1003 ↑ ↑ length=5 nr of data bytes = 7
Answer	10020103030000051003

Example 4	**Read Setpoint 0x1010**
Request	10020103050401210121 1003
Answer	10020103 05 02012110101010 1003 ↑ ↑ length=5 nr of data bytes = 7

Example 5	**Read Measure from node 0x10 (node 16)**
Request	extra 0x10 byte ↓ 10020110 10 05 0401210120 1003 ↑ ↑ length=5 nr of data bytes = 5
Answer	10020110 10 05 0201217D00 1003

Example 6	**Read Measure using message sequence number 0x10 (16)**
Request	extra 0x10 byte ↓ 1002 1010 80 05 0401210120 1003 ↑ message sequence number
Answer	1002 1010 80 05 0201217D00 1003

4 DUAL INTERFACE OPERATION

When operating a controller (reading measured value and sending setpoint) for proper operation it is important that the controller gets its setpoint from the right source. Setpoints may come from different sources: analog input, field bus interface or RS232 or may be overruled by close valve or open valve (purge) commands. Therefore it is important to know what the setpoint source of the controller is. This can be set by means of parameter control mode (process 1, parameter 4).

In some cases it is possible that the setpoints may come from 2 sources at the same time. The last send setpoint will be valid and send to the controller. This is the case in control mode = 0, when setpoints may come through any field bus interface or RS232.

However, there could be situations where control over the instrument seems impossible. This is the case when the instrument comes into a safe-state e.g. when field bus communication is disturbed or disconnected. Valve will be forced to a safe state automatically: closed (NC) or fully open (NO).

In case you want to get control back via RS232 operation, you have to change the control mode. When control mode gets value 18, safe state will be overruled and sending setpoints via RS232 interface will have effect on the controller again.

See also document nr. 9.17.023 for more detailed description about digital instrument parameters and their behaviour.
http://www.bronkhorst.com/en/downloads/instruction_manuals/

5 PARAMETER INFORMATION

ProPar is used for parameter value exchange between instruments and operation devices (keyboard or PC-interface). Parameter information consists of several properties.. In the 'parameter properties' table you will find a list of parameters and their properties. In the 'parameter values' table, the values are described more detailed. This list consists mostly of parameters for mode settings.

Property description in parameter properties table:	
Item	**Description**
Parameter(DDE)	unique parameter number (also used for DDE-communication : P(x))
Name	name of the parameter, used for parameter identification
process	process where parameter is used on ProPar device used for communication directly through RS232 when filled in the table, this value has to be used (for parameters located in only 1 process) when empty in the table, process has to be determined from the ProPar system information (for parameters located in more than one process, e.g. setpoint, measure);
FBnr(parameter)	parameter number in process on ProPar device used for communication directly through RS232
VarType	variable type for information about amount of bytes • c (unsigned) char type 1 byte value 0..255 • I (unsigned) integer type 2 bytes value 0..65535 • f float type 4 bytes value +-1.18E-38..+-3.39E+38 (IEEE-floating point notation) • l (unsigned) long type 4 bytes value 0..4294967295 data types > 1 byte are MSB first.
VarLength	variable length to indicate length of string of chars used in combination with VarType c for transportation of strings through ProPar: value 0..65535 VarLength indicates the amount of bytes for a parameter type • -2 indicates that a string is zero-terminated, not defined for length • X indicates a string with a length of X bytes (characters) • 0 means no info required, i.e. zero-terminated.
Min	minimum value of parameter allowed when parameter is read/written via RS232, the value will be checked on this limit (error when out of limit)
Max	maximum value of parameter allowed when parameter is read/written via RS232, the value will be checked on this limit (error when out of limit)
Read	indication if parameter is allowed to be read via ProPar
Write	indication if parameter is allowed to be written via ProPar
Poll	indication if parameter should be polled continuously by RS232 application in order to keep (changing) parameter information up to date
Advanced	indication if parameter is for advanced users only these are mainly parameters for maintenance/service
Secured	indication if parameter is secured for use through ProPar reading this parameter is possible, but changing it needs special handling
High security	indication if parameter is highly secured (only few parameters) reading this parameter is possible, but changing it needs special handling
Description	short description about meaning of parameter or what it is used for
DDE str	parameter DDE string (max. 10 characters), DDE value when parameter is not available on instrument; also: until FlowDDE V4.58 used for parameter identification

Parameter acceptance:
Changing parameter values is possible when a parameter is not read-only and not secured. The range and type of parameters are described in the tables. When parameter values are out of range they will be either 'clipped' on the nearest value allowed or you will get an error message: 'parameter value error'.

(FlowDDE) Parameter numbers:
All parameter information is referenced to the parameter number. This is a unique number for a parameter to avoid redundancy. These numbers are needed for DDE communication only.

For communication with ProPar through other ways than DDE: directly via RS232 ASCII-strings or via C-libraries (DOS or Windows), use the parameter numbers for the ProPar devices (in column FBnr of table Parameter properties). Now you will always have to know the node-address of the instrument on the ProPar, the process number on the instrument and the parameter number on the instrument.

Process nr could be read from the table or has to be determined, when nothing is filled in. In most cases process number will be 1. Node-address should be determined also. This is the node-address of the instrument on the ProPar. Newer RS232 protocols on Multibus instruments accept node = 128. When sending messages to this node address, the message will be always accepted, unregarding the node address of the instrument on the bus.

It is important to know that not all parameters are available on all ProPar/Multibus devices.
For more details about parameters and their use see also document nr. 9.17.023 for description of digital instruments. If you have the program FlowDDE, you can also get an overview of which parameters are available on which devices.

APPENDIX 1 AND 2 WILL GIVE INFORMATION ABOUT PARAMETERS, THEIR PROPERTIES AND THEIR POSSIBLE VALUES.

6 TROUBLESHOOTING

6.1 LED INDICATIONS

LED indications can be very useful in case of problems with the instrument.
The green LED is normally used for instrument status indication, like normal operation or special function mode. For DeviceNet it is also possible to show that the instrument is in abort state and idle state. The red LED is normally used to for error/warning indication (how longer the flash, how greater the trouble).

More information can be found in the manual "917023 Operational instructions digital instruments"
This document can be found at:
http://www.bronkhorst.com/en/downloads/instruction_manuals/

6.2 TROUBLESHOOTING HINTS AND TIPS

SITUATION	Description
RS232 communication problems	Check cables. Make sure correct cables are used for specific purpose. Check address of interface (slave). Sending messages to node 128 will mostly be accepted by the interface. Try to reset the instrument and/or restart your PC/PLC. Make sure your messages are assembled according to ProPar protocol description. Make sure the parameter values you try to read/write are available and in the correct ranges (check tables). Controller doesn't respond on setpoints: • Check control mode, when 0 and other field bus gives error: safe state will be entered, resulting in safe setpoint. Can be overruled by making control mode = 18 (RS232 only operation) • Alarm or counter module in instrument forces setpoint to alarm setpoint. Reset alarm or counter and proceed. • Setpoint slope could have very high value. New setpoints will be reached when this slope time has been elapsed. Make setpoint slope smaller. • Control mode could have other value than 0 or 18. Check function when value is different. • If measure doesn't change check forward pressure and piping (evt. shut-off valves). • Make sure setpoints are within allowed range: 0...32000 (= 0...100%). • Make sure setpoints are send to proper instrument and process (mostly = 1) and parameter (FBnr for setpoint = 1), and type of data is correct (short integer = 2 bytes MSB first)
Other (ProPar) problems	Contact Bronkhorst® local sales representative or send e-mail describing your problem to: see service chapter.

7 SERVICE

For current information on Bronkhorst® and service addresses please visit our website:

http://www.bronkhorst.com

Do you have any questions about our products? Our Sales Department will gladly assist you selecting the right product for your application. Contact sales by e-mail:

sales@bronkhorst.com

For after-sales questions, our Customer Service Department is available with help and guidance. To contact CSD by e-mail:

support@bronkhorst.com

No matter the time zone, our experts within the Support Group are available to answer your request immediately or ensure appropriate further action. Our experts can be reached at:

+31 859 02 18 66

8 APPENDIX

8.1 PARAMETER VALUES TABLE

See Below

8.2 PARAMETER PROPERTIES TABLE

See Below

Parameter values table

Parameter number (DDE)	Parameter name	Filter	Value	Description
6	Arbitrage		1	temporary busmaster
6	Arbitrage		2	always busmaster
6	Arbitrage		3	automatic busmaster
6	Arbitrage		67	auto busmaster and auto bus optimalization (fast token ring)
12	Control mode		0	setpoint = BUS setpoint
12	Control mode		1	setpoint = analog input
12	Control mode		2	setpoint = master output(FLOW-BUS) * slave factor(FLOW-BUS)
12	Control mode		3	close valve
12	Control mode		4	controller idle (no reaction on changes in sensor signal)
12	Control mode		5	testmode enable (select subject with par 70)
12	Control mode		6	tuningmode enable (select subtject with par 79)
12	Control mode		7	setpoint = 100%
12	Control mode		8	purge valve (fully open)
12	Control mode		9	calibration mode enable (select subject with par 58)
12	Control mode		10	setpoint = master output(analog in) * slave factor(FLOW-BUS)
12	Control mode		11	setpoint = keyboard OR FLOW-BUS setpoint
12	Control mode		12	setpoint = 0%
12	Control mode		13	setpoint = master output(FLOW-BUS) * slave factor(analog in)
12	Control mode		14	(FPP) Range select mode
12	Control mode		15	(FPP) Manual start sensor select, automatic end sensor
12	Control mode		16	(FPP) Automatic start sensor select, manual end sensor
12	Control mode		17	(FPP) Automatic start and end sensor
12	Control mode		18	setpoint = RS232 setpoint
12	Control mode		19	RS232 broadcast mode
12	Control mode		20	valve stearing (valve = setpoint)
12	Control mode		21	analog valve stearing (valve = analog setpoint)
12	Control mode		22	valve safe state
22	Sensor type		0	pressure (controller)
22	Sensor type		1	liquid volume (controller)
22	Sensor type		2	liquid/gas mass (controller)
22	Sensor type		3	gas volume (controller)
22	Sensor type		4	other sensor type (controller)
22	Sensor type		128	pressure (sensor)
22	Sensor type		129	liquid volume (sensor)
22	Sensor type		130	liquid/gas mass (sensor)
22	Sensor type		131	gas volume (sensor)
22	Sensor type		132	other sensor type (sensor)
28	Alarm info	&H01	0	no error message in alarm error status register
28	Alarm info	&H01	1	at least 1 error message in alarm error status register
28	Alarm info	&H02	0	no warning message in alarm warning status register
28	Alarm info	&H02	1	at least 1 warning message in alarm warning status register
28	Alarm info	&H04	0	no minimum alarm message (measure>minimum limit)
28	Alarm info	&H04	1	minimum alarm message for measured signal
28	Alarm info	&H08	0	no maximum alarm message (measure<maximum limit)
28	Alarm info	&H08	1	maximum alarm message for measured signal
28	Alarm info	&H10	0	batch counter has not reached its limit
28	Alarm info	&H10	1	batch counter has reached its limit
28	Alarm info	&H20	0	response O.K. (setpoint-measure within limit)
28	Alarm info	&H20	1	response alarm message: setpoint-measure is too high
28	Alarm info	&H40	0	master output signal O.K. (or not used)
28	Alarm info	&H40	1	master output signal not received: check master instrument
28	Alarm info	&H80	0	hardware O.K.
28	Alarm info	&H80	1	hardware error message: check your hardware
44	Operation mode T/A		0	OFF
44	Operation mode T/A		1	A: MAX & RESP AUTO; T: UP TO LIMIT
44	Operation mode T/A		2	A: MIN & RESP AUTO; T: UP AND REPEAT
44	Operation mode T/A		3	A: MAX & RESP; T: DOWN FROM LIMIT
44	Operation mode T/A		4	A: MIN & RESP; T: DOWN AND REPEAT
44	Operation mode T/A		5	A: MAXIMUM ALARM; T: ALWAYS UP
44	Operation mode T/A		6	A: MINIMUM ALARM
44	Operation mode T/A		7	A: RESPONSE ALARM
53	Analog mode	&H3F	0	0...5 Vdc operation
53	Analog mode	&H3F	1	0...10 Vdc operation
53	Analog mode	&H3F	2	0...20 mA operation
53	Analog mode	&H3F	3	4...20 mA operation
53	Analog mode	&H3F	4	15...20 mA operation
53	Analog mode	&H40	0	Analog input enabled
53	Analog mode	&H40	1	Analog input disabled
53	Analog mode	&H80	0	Analog output enabled
53	Analog mode	&H80	1	Analog output disabled
58	Calibration mode		0	idle: no action
58	Calibration mode		1	adc self calibration
58	Calibration mode		2	dmfc
58	Calibration mode		3	dmfc
58	Calibration mode		4	dmfc
58	Calibration mode		5	dmfc
58	Calibration mode		6	dmfc
58	Calibration mode		7	dmfc
58	Calibration mode		8	dmfc
58	Calibration mode		9	customer zero

Parameter number (DDE)	Parameter name	Filter	Value	Description
58	Calibration mode		10	adjust Vref output by connecting it to analog in
58	Calibration mode		11	adjust analog out by connecting it to analog in
58	Calibration mode		12	adjust valveoutput by connecting it to analog in
58	Calibration mode		13	dmfc
58	Calibration mode		14	dmfc
58	Calibration mode		15	analog output = 0 %
58	Calibration mode		16	analog output = 100 %
58	Calibration mode		17	analog output = 50 %
58	Calibration mode		18	factory zero
58	Calibration mode		19	sensor differentiator (setpoint steps are needed!)
58	Calibration mode		20	automatic sensor configuration
58	Calibration mode		21	sensor temperature calibration
58	Calibration mode		22	customer zero (no control mode 9 needed)
58	Calibration mode		255	Error mode (result of previous cal mode)
60	Monitor mode		0	(filtered) setpoint
60	Monitor mode		1	controller error input signal / raw sensor signal
60	Monitor mode		2	controller output signal to valve
60	Monitor mode		3	sensor signal slow
60	Monitor mode		4	sensor signal slow filtered
60	Monitor mode		5	linearization output
60	Monitor mode		6	differentiator output
60	Monitor mode		7	differentiator output filtered
60	Monitor mode		8	normal sensor signal (Output)
60	Monitor mode		9	analog input signal
60	Monitor mode		10	power supply voltage
60	Monitor mode		11	mass flow in display unit (normally ln/min)
60	Monitor mode		12	volume flow in l/min
60	Monitor mode		13	temperature in ℃
60	Monitor mode		14	pressure absolute in mbara
60	Monitor mode		15	time in msec/frequency in Hz.
60	Monitor mode		16	calibrated volume at actual sensor in ml
60	Monitor mode		17	delta-P pressure in mbarg
60	Monitor mode		18	atmospheric (barometer) pressure in mbara
60	Monitor mode		19	mass flow in kg/min
61	Alarm register1	&H8000000000000000	0	No diagnostics available in warning register
61	Alarm register1	&H8000000000000000	1	Diagnostics available in warning register
62	Alarm register2	&H8000000000000000	0	No diagnostics available in error register
62	Alarm register2	&H8000000000000000	1	Diagnostics available in error register
67	ADC control register	&H001000	0	ADC bipolar mode
67	ADC control register	&H001000	1	ADC unipolar mode
67	ADC control register	&H1000000	0	Disable zero measure threshold
67	ADC control register	&H1000000	1	Enable zero measure threshold
67	ADC control register	&H1C0000	0	ADC gain = 1x
67	ADC control register	&H1C0000	1	ADC gain = 2x
67	ADC control register	&H1C0000	2	ADC gain = 4x
67	ADC control register	&H1C0000	3	ADC gain = 8x
67	ADC control register	&H1C0000	4	ADC gain = 16x
67	ADC control register	&H1C0000	5	ADC gain = 32x
67	ADC control register	&H1C0000	6	ADC gain = 64x
67	ADC control register	&H1C0000	7	ADC gain = 128x
69	<AlarmEnble>		0	disable
69	<AlarmEnble>		1	enable
70	Test mode		0	idle; no action
70	Test mode		1	uProcessor
70	Test mode		2	IO
70	Test mode		3	RAM
70	Test mode		4	FRAM
70	Test mode		5	ADC
70	Test mode		6	DAC
70	Test mode		7	sensor
70	Test mode		8	valve drive circuit
70	Test mode		9	Vref
70	Test mode		10	FLOW-BUS
70	Test mode		11	calibration
70	Test mode		12	keyboard
71	<ADC channel select>		1	AD channel 1
71	<ADC channel select>		2	AD channel 2
79	Tuning mode		0	idle; no action
79	Tuning mode		1	sensor
79	Tuning mode		2	valve
79	Tuning mode		3	Fuzzy controller normal operation
79	Tuning mode		4	Fuzzy controller open at zero
79	Tuning mode		5	PID controller
80	Valve default		0	normally closed
80	Valve default		1	normally opened
80	Valve default		2	normally closed inverse controlled
80	Valve default		3	normally opened inverse controlled
80	Valve default		4	remain position
86	IO status	&H01	1	read diagnostic jumper (no diagnostics, read/write)
86	IO status	&H02	1	not used
86	IO status	&H04	1	read analog jumper (use cntrlmode, read/write)
86	IO status	&H08	1	read micro switch (read/write)
86	IO status	&H10	1	diagnostic jumper set (read only)
86	IO status	&H20	1	initialization jumper set (read only)

Parameter number (DDE)	Parameter name	Filter	Value	Description
86	IO status	&H40	1	analog jumper set (read only)
86	IO status	&H80	1	micro switch pressed (read only)
106	Pressure sensor type		0	delta-P 0..5" W.C.
106	Pressure sensor type		1	delta-P 0...10" W.C.
106	Pressure sensor type		2	absolute pressure 800-1200 mbar
106	Pressure sensor type		3	absolute pressure 800-1100 mbar
106	Pressure sensor type		4	delta-P -5...0 "W.C.
106	Pressure sensor type		5	delta-P -10...0 "W.C.
106	Pressure sensor type		6	delta-P -10...+10 "W.C.
106	Pressure sensor type		7	delta-P 0...1 PSI
106	Pressure sensor type		8	delta-P -1...0 PSI
106	Pressure sensor type		10	absolute pressure 0-10 bar
114	Reset		0	no reset
114	Reset		1	reset counter value (no mode change) or common reset
114	Reset		2	reset alarm
114	Reset		3	restart batch counter
114	Reset		4	reset counter value (counter off)
114	Reset		5	Reset module (soft reset)
118	Alarm mode		0	off
118	Alarm mode		1	alarm on absolute limits
118	Alarm mode		2	alarm on limits related to setpoint (response alarm)
118	Alarm mode		3	alarm when instrument powers-up (eg. after power-down)
119	Alarm output mode		0	no relais activity at alarm
119	Alarm output mode		1	relais pulses until reset
119	Alarm output mode		2	relais activated until reset
120	Alarm setpoint mode		0	no setpoint change at alarm
120	Alarm setpoint mode		1	new/safe setpoint at alarm enabled (set at par 121)
125	Counter output mode		0	no relais activity at batch limit
125	Counter output mode		1	relais pulses after reaching batch limit until reset
125	Counter output mode		2	relais activated after reaching batch limit until reset
126	Counter setpoint mode		0	setpoint change at batch limit disabled
126	Counter setpoint mode		1	setpoint change at batch limit enabled
130	Counter mode		0	off
130	Counter mode		1	counting upwards continuously
130	Counter mode		2	counting up to limit (batchcounter)
147	Range select		0	calibration ready/stop
147	Range select		1	run calibration until stopsensor 1/select range 1
147	Range select		2	run calibration until stopsensor 2/select range 2
147	Range select		3	run calibration until stopsensor 3/select range 3
147	Range select		4	run calibration until stopsensor 4/select range 4
147	Range select		5	run calibration and select range 5
147	Range select		9	run calibration with automatic range selection
147	Range select		19	run until stopsensor 1 until 3 values between limit
147	Range select		29	run until stopsensor 2 until 3 values between limit
147	Range select		39	run until stopsensor 3 until 3 values between limit
147	Range select		49	run until stopsensor 4 until 3 values between limit
147	Range select		59	run and select range 5 until 3 values between limit
147	Range select		99	run with auto-select + 3 values between limit
156	Reset alarm enable		0	no reset possible
156	Reset alarm enable		1	reset: keyboard
156	Reset alarm enable		2	reset: external
156	Reset alarm enable		3	reset: keyboard or external
156	Reset alarm enable		4	reset: FLOW-BUS
156	Reset alarm enable		5	reset: FLOW-BUS or keyboard
156	Reset alarm enable		6	reset: FLOW-BUS or external
156	Reset alarm enable		7	reset: FLOW-BUS or keyboard or external
156	Reset alarm enable		8	reset: automatic
156	Reset alarm enable		9	reset: automatic or keyboard
156	Reset alarm enable		10	reset: automatic or external
156	Reset alarm enable		11	reset: automatic or keyboard or external
156	Reset alarm enable		12	reset: automatic or FLOW-BUS
156	Reset alarm enable		13	reset: automatic or FLOW-BUS or keyboard
156	Reset alarm enable		14	reset: automatic or FLOW-BUS or external
156	Reset alarm enable		15	reset: automatic or FLOW-BUS or keyboard or external
157	Reset counter enable		0	no reset possible
157	Reset counter enable		1	reset: keyboard
157	Reset counter enable		2	reset: external
157	Reset counter enable		3	reset: keyboard or external
157	Reset counter enable		4	reset: FLOW-BUS
157	Reset counter enable		5	reset: FLOW-BUS or keyboard
157	Reset counter enable		6	reset: FLOW-BUS or external
157	Reset counter enable		7	reset: FLOW-BUS or keyboard or external
157	Reset counter enable		8	reset: automatic
157	Reset counter enable		9	reset: automatic or keyboard
157	Reset counter enable		10	reset: automatic or external
157	Reset counter enable		11	reset: automatic or keyboard or external
157	Reset counter enable		12	reset: automatic or FLOW-BUS
157	Reset counter enable		13	reset: automatic or FLOW-BUS or keyboard
157	Reset counter enable		14	reset: automatic or FLOW-BUS or external
157	Reset counter enable		15	reset: automatic or FLOW-BUS or keyboard or external
166	Controller features	&H01	0	valve in normal position after startup
166	Controller features	&H01	1	valve in safe position after startup
166	Controller features	&H02	0	open from zero with PID output to valve
166	Controller features	&H02	1	open from zero with ramp output to valve

Parameter number (DDE)	Parameter name	Filter	Value	Description
166	Controller features	&H04	0	fixed monitor output signal
166	Controller features	&H04	1	monitor output changed at setpoint steps
166	Controller features	&H08	0	voltage drift compensation for valve output turned on
166	Controller features	&H08	1	voltage drift compensation for valve output turned off
166	Controller features	&H10	0	auto slope disabled
166	Controller features	&H10	1	auto slope enabled for pilot valves
166	Controller features	&H20	0	automatic correction for valve open turned on
166	Controller features	&H20	1	automatic correction for valve open turned off
166	Controller features	&H40	0	controller special mode (valve output steps) turned off
166	Controller features	&H40	1	controller special mode (valve output steps) turned on
166	Controller features	&H80	0	valve overshoot protection turned off
166	Controller features	&H80	1	valve overshoot protection turned on
175	Identification number		0	UFO?: Unidentified FLOW-BUS Object
175	Identification number		1	RS232/FLOW-BUS interface
175	Identification number		2	PC(ISA) interface
175	Identification number		3	ADDA4 (4 channels)
175	Identification number		4	R/C-module, 32 channels
175	Identification number		5	T/A-module
175	Identification number		6	ADDA1: 1 channel ADDA converter module
175	Identification number		7	DMFC: digital mass flow controller
175	Identification number		8	DMFM: digital mass flow meter
175	Identification number		9	DEPC: digital electronic pressure controller
175	Identification number		10	DEPM: digital electronic pressure meter
175	Identification number		11	ACT: single actuator
175	Identification number		12	DLFC: digital liquid flow controller
175	Identification number		13	DLFM: digital liquid flow meter
175	Identification number		14	DSCM-A: digital single channel module for analog instruments
175	Identification number		15	DSCM-D: digital single channel module for digital instr.
175	Identification number		16	FRM: FLOW-BUS rotor meter (calibration-instrument)
175	Identification number		17	FTM: FLOW-BUS turbine meter (calibration-instrument)
175	Identification number		18	FPP: FLOW-BUS piston prover/tube (calibration-instrument)
175	Identification number		19	F/A-module: special version of T/A-module
175	Identification number		20	DSCM-E: evaporator controller module (single channel)
175	Identification number		21	DSCM-C: digital single channel module for calibrators
175	Identification number		22	DDCM-A: digital dual channel module for analog instruments
175	Identification number		23	DMCM-D: digital multi channel module for digital instruments
175	Identification number		24	Profibus-DP/FLOW-BUS interface module
175	Identification number		25	FLOW-BUS Coriolis Meter
175	Identification number		26	FBI: FLOW-BUS Balance Interface
175	Identification number		27	CORIFC: CoriFlow Controller
175	Identification number		28	CORIFM: CoriFlow Meter
175	Identification number		29	FICC: FLOW-BUS Interface Climate Control
175	Identification number		30	IFI: Instrument FLOW-BUS Interface
175	Identification number		31	KFI: Keithley FLOW-BUS Interface
175	Identification number		32	FSI: FLOW-BUS Switch Interface
175	Identification number		33	MSCI: Multi-Sensor/Confroller Interface
175	Identification number		34	APP-D: Active Piston Prover
175	Identification number		35	LFI: Leaktester FLOW-BUS Interface
185	Device function		0	Unknown
185	Device function		1	Interface
185	Device function		2	ADDA
185	Device function		3	Operator
185	Device function		4	Supervisor (totalizer/alarm)
185	Device function		5	Controller
185	Device function		6	Meter
185	Device function		7	Special
185	Device function		8	(Protocol) converter
197	Calibrations options	&H01	0	Automatic capacity setting for optimal resolution
197	Calibrations options	&H01	1	Manual capacity setting for optimal resolution
197	Calibrations options	&H02	0	Barometer value input via parameter 107: BaroPress
197	Calibrations options	&H02	1	Barometer is master; input automatically from master
200	Interface configuration		0	Configuration A: 14 ch. Standard parms. with network scan
200	Interface configuration		1	Configuration B: 14 ch. Standard parms with fixed chan list
200	Interface configuration		2	Configuration C: 7 ch. Extended parms with fixed chan list
200	Interface configuration		3	Configuration D: 11 ch. Extended parms with network scan
208	Manufacturer status regi	&H800000	0	No diagnostics available in manufacturer status register
208	Manufacturer status regi	&H800000	1	Diagnostics available in manufacturer status register
209	Manufacturer warning re	&H800000	0	No diagnostics available in manufacturer warning register
209	Manufacturer warning re	&H800000	1	Diagnostics available in manufacturer warning register
210	Manufacturer error regis	&H800000	0	No diagnostics available in manufacturer error register
210	Manufacturer error regis	&H800000	1	Diagnostics available in manufacturer error register
212	Diagnostic mode		0	Debug mode off
212	Diagnostic mode		1	Debug mode on
213	Manufacturer status ena		0	set status bit (range 0…127)
213	Manufacturer status ena		127	set status bit (range 0…127)
213	Manufacturer status ena		254	clear all status bits
213	Manufacturer status ena		255	set all status bits
232	Valve mode		0	voltage drive mode
232	Valve mode		1	current drive mode
238	Fluidset properties	&H01	0	Fluidset is disabled
238	Fluidset properties	&H01	1	Fluidset is enabled
238	Fluidset properties	&H02	0	Fluidset is not set by Bronkhorst High-Tech
238	Fluidset properties	&H02	2	Fluidset is set by Bronkhorst High-Tech
238	Fluidset properties	&H04	0	Fluidset is not calibrated on actual gas

Parameter number (DDE)	Parameter name	Filter	Value	Description
238	Fluidset properties	&H04	4	Fluidset is calibrated on actual gas
295	Sensor bridge settings	&H01	1	Bridge on
295	Sensor bridge settings	&H02	1	3 windings C
295	Sensor bridge settings	&H04	1	3 windings D
295	Sensor bridge settings	&H100	1	Automatic sensor configuration on
295	Sensor bridge settings	&H200	1	Sensor protection enabled
301	Valve safe state		0	0 mA
301	Valve safe state		1	max mA
301	Valve safe state		2	Close
301	Valve safe state		3	Open
301	Valve safe state		4	Idle
301	Valve safe state		5	Value (for DeviceNet only)
305	Bus1 selection		0	FLOW-BUS
305	Bus1 selection		1	Modbus
305	Bus1 selection		2	ProPar
305	Bus1 selection		10	DeviceNet
305	Bus1 selection		13	Profibus-DP
305	Bus1 selection		255	NoBus
306	Bus1 medium		0	RS232
306	Bus1 medium		1	RS485
307	Bus2 mode		0	Normal
307	Bus2 mode		1	Config mode
308	Bus2 selection		0	FLOW-BUS
308	Bus2 selection		1	Modbus
308	Bus2 selection		2	ProPar
311	Bus2 medium		0	RS232
311	Bus2 medium		1	RS485
314	PIO channel selection		0	Analog input
314	PIO channel selection		1	Analog output
314	PIO channel selection		2	General purpose in-/output
319	PIO configuration selecti		0	Voltage output (0..10 V)
319	PIO configuration selecti		1	Current output (0..20 mA)
319	PIO configuration selecti		2	Digital output
319	PIO configuration selecti		3	Frequency output
319	PIO configuration selecti		4	Duty cycle output (20 kHz)
319	PIO configuration selecti		5	Digital pulse output
319	PIO configuration selecti		6	Voltage input (0..10 V)
319	PIO configuration selecti		7	Current input (0..20 mA)
319	PIO configuration selecti		8	Digital input
319	PIO configuration selecti		255	Disabled
329	Setpoint monitor mode		0	Setpoint
329	Setpoint monitor mode		1	Filtered setpoint
329	Setpoint monitor mode		2	Setpoint after linear slope
335	Bus1 parity		0	None
335	Bus1 parity		1	Odd
335	Bus1 parity		2	Even
336	Bus2 parity		0	None
336	Bus2 parity		1	Odd
336	Bus2 parity		2	Even

Parameter properties table

Parameter number (DDE)	Parameter name	Group 0	Group 1	Group 2	Process number	FB nr (par)	Var Type	Var Length	Min value	Max value	Read	Write	Poll	Secured	Highly Secured	Default Value	DDE str	Description
1	Identification string	13			0	0	c	-2			Yes	Yes	No	No	No	7SN999999	identstrng	identnr.+softwareversion[+serialnr.]
2	Primary node address	1			0	1	c		0	128	Yes	Yes	No	Yes	Yes	0	pna	primary node address: network parameter FLOW-BUS
3	Secondary node address	1			0	2	c		0	128	Yes	Yes	No	Yes	Yes	0	sna	secundary node address: network parameter FLOW-BUS
4	Next node address	1			0	3	c		0	128	Yes	Yes	No	No	No	1	nna	next node address: network parameter FLOW-BUS
5	Last node address	1			0	4	c		0	128	Yes	Yes	No	No	No	32	lna	last node address: network parameter FLOW-BUS
6	Arbitrage	1			0	5	c		0	255	Yes	Yes	No	Yes	Yes	67	arbitrage	FLOW-BUS arbitrage setting and/or automatic optimization
7	Initreset	12			0	10	c		0	255	Yes	Yes	No	No	No		initreset	init and reset security key commands for network/parameter settings
8	Measure	2				0	i		-23593	41942	Yes	Yes	Yes	No	No	0	measure	measured value (100% = 32000)
9	Setpoint	2	18			1	i		0	32767	Yes	Yes	Yes	No	No	0	setpoint	setpoint: wanted value (100% = 32000)
10	Setpoint slope	18				2	i		0	30000	Yes	Yes	No	No	No	0	setpslope	setpoint ramp signal 0..100 % in up to slope x 0.1 sec.
11	Analog input	2	18			3	i		-23593	41942	Yes	No	Yes	No	No	0	analoginp	analog input signal, normally used for ext. setp. (100% = 32000)
12	Control mode	18				4	c		0	255	Yes	Yes	No	No	No	0	cntrlmode	control mode selection for instrument or module
13	Polynomial constant A	3				5	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0.0	polycnst A	polynomial constant A (offset)
14	Polynomial constant B	3				6	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	1.0	polycnst B	polynomial constant B (span]
15	Polynomial constant C	3				7	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0.0	polycnst C	polynomial constant C
16	Polynomial constant D	3				8	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0.0	polycnst D	polynomial constant D
17	Polynomial constant E	3				9	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0.0	polycnst E	polynomial constant E (offset) for setpoint or power value
18	Polynomial constant F	3				10	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	1.0	polycnst F	polynomial constant F (span) for setpoint or power value
19	Polynomial constant G	3				11	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0.0	polycnst G	polynomial constant G for setpoint or power value
20	Polynomial constant H	3				12	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0.0	polycnst H	polynomial constant H for setpoint or power value
21	Capacity	3	19			13	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	1.0	capacity	readout value at 100% in capacity (readout) unit
22	Sensor type	3				14	c		0	255	Yes	Yes	No	Yes	No	3	sensortype	sensor type information for actual reading and sensor/controller indication
23	Capacity unit index	3	19			15	c		0	255	Yes	Yes	No	Yes	No	0	capunit	pointer to capacity (readout) unit (classic unit table]
24	Fluid number	3				16	c		0	8	Yes	Yes	No	No	No	0	fluidnr	fluid number: pointer to fluidset (for e.g. polynome, name and cap.)
25	Fluid name	3				17	c	10			Yes	Yes	No	Yes	No	AIR	fluidname	name of fluid
26	Claim node	12				18	c		0	128	Yes	Yes	No	Yes	No	0	claimnode	node address of module with operation rights
27	Modify	12				19	c		0	255	Yes	Yes	No	No	No	0	modify	contains number(s) of changed par (0xXX par nr, 0xFF more than one par changed)
28	Alarm info	4				20	c		0	255	Yes	No	Yes	No	No	0	alarminfo	status information of several alarms/errors in the instrument
29	Channel amount	17			0	12	c		1	120	Yes	Yes	No	No	No	32	chanamount	amount of channels which can be operated
30	First channel	17			0	13	c		1	120	Yes	Yes	No	No	No	1	firstchan	first channel that can be operated
31	Last channel	17			0	14	c		1	120	Yes	Yes	No	No	No	32	lastchan	last channel that can be operated
32	<hostcontrl>	5			9	1	c		0	1	Yes	Yes	No	No	No	0	hostcontrl	operation by HOST computer enable flag
33	Alarm message unit type	5			10	0	c	16			Yes	No	Yes	No	No		alrmmsgTA	alarm message string with unit type information
34	Alarm message number	5			10	1	c	16			Yes	No	Yes	No	No		alrmmsgnr	alarm message string with unit number information
35	Relay status	5			10	2	c	8			Yes	No	No	No	No		relstatus	status of relays/potential free contacts ('0' = not activated, '1' = activated)
36	Actual counter value	5				0	f		0	3.40282E+38	Yes	No	Yes	No	No	0	actualval	actual value of counter
37	Signal input selection	5				1	c	8			Yes	Yes	No	No	No		signinpsel	signal input selection (' '=no value,'+'=pos value,'-'=neg value input)
38	Reset input selection	5				2	c	8			Yes	Yes	No	No	No		resinpsel	external reset input enable/disable ('E'=enable,' '=disable)
39	<limit>	5				3	f		0	3.40282E+38	Yes	Yes	No	No	No		limit	limit/batch for counter in sensor standard units
40	Delay time	5				4	c	8	0	99235959	Yes	Yes	No	No	No	00000000	delaytime	delay time string in days,hours,minutes,seconds
41	Duration time	5				5	c	8	0	99235959	Yes	Yes	No	No	No	00000000	duratntime	duration time string in days,hours,minutes,seconds
42	Valve output setting	5				6	c	8			Yes	Yes	No	No	No	00000000	vlvoutset	valve output setting ('0'=do nothing, '1'=close valve)
43	Relay output setting	5				7	c	8			Yes	Yes	No	No	No	LLLLLLLL	reloutset	relay output setting ('L'=low, 'H'=high, 'P'=pulse (1 sec.))
44	Operation mode T/A	5				8	c		0	9	Yes	Yes	No	No	No	0	opermodeTA	operation mode of T/A module
45	Readout unit	5				9	c	7			Yes	No	No	No	No	ln/min	readunit	readout unit string
46	Readout factor	5				10	f		1E-10	10000000000	Yes	No	No	No	No	1	readfact	readout factor matching readout unit string
47	Reset unit	5				12	c		0	1	No	Yes	No	No	No		resetunit	reset unit command (1=reset T/A unit)
48	Valve differentiator down	6				9	f		0	3.40282E+38	Yes	Yes	No	Yes	No	0.1	TdValveDn	valve output differentiation time constant downwards
49	Valve differentiator up	6				10	f		0	3.40282E+38	Yes	Yes	No	Yes	No	0.1	TdValveUp	valve output differentiation time constant upwards
50	Sensor differentiator down	6				11	f		0	3.40282E+38	Yes	Yes	No	Yes	No	5.8	TdSensorDn	sensor signal differentiation time constant downwards
51	Sensor differentiator up	6				12	f		0	3.40282E+38	Yes	Yes	No	Yes	No	5.6	TdSensorUp	sensor signal differentiation time constant upwards
52	Cycle time	6			114	12	c		0	255	Yes	Yes	No	Yes	No	6	CycleTime	cycle time * 10 msec. main loop signal processing
53	Analog mode	10			115	3	c		0	255	Yes	Yes	No	Yes	No	0	AnalogMode	analog mode selection for analog operation
54	Reference voltage	10			116	6	i		0	65535	Yes	Yes	No	Yes	No		VrefOutput	reference voltage output signal for analog operation
55	Valve output	8			114	1	l		0	16777215	Yes	Yes	Yes	No	No	0	ValveOut	valve output signal (24-bit number in range 0..14.3Vdc/0..23.3Vdc)
56	Dynamic display factor	6			117	1	f		0	1	Yes	Yes	No	Yes	No	0,001	DynDispFct	dynamic display factor for display filter (0=max, 1=min goes with par 57)
57	Static display factor	6			117	2	f		0	1	Yes	Yes	No	Yes	No	0,000001	StaDispFct	static display factor for display filter (0=max, 1=min goes with par 56)
58	Calibration mode	7			115	1	c		0	255	Yes	Yes	No	Yes	No	0	CalMode	calibration mode selection (not active until cntrlmode has been set to value 9)
59	Valve offset	8			116	7	i		-32767	65535	Yes	Yes	No	Yes	No	61000	ValveOffst	valve offset: amount of DAC steps within 1 potmeter step
60	Monitor mode	2			115	2	c		0	255	Yes	Yes	No	Yes	No	7	Monitor	monitor: output signal (measure) selection for bus and analog output
61	Alarm register1	4			114	2	c	8			Yes	Yes	No	Yes	No		AlarmReg1	alarm register containing warning flags
62	Alarm register2	4			114	3	c	8			Yes	Yes	No	Yes	No		AlarmReg2	alarm register containing critical error flags
63	<CalRegZS1>	9			116	1	l		0	16777215	Yes	Yes	No	Yes	No	210A7D	CalRegZS1	calibration register zero scale input 1 ADC
64	<CalRegFS1>	9			116	2	l		0	16777215	Yes	Yes	No	Yes	No	52A513	CalRegFS1	calibration register full scale input 1 ADC
65	<CalRegZS2>	9			116	3	l		0	16777215	Yes	Yes	No	Yes	No	210A7D	CalRegZS2	calibration register zero scale input 2 ADC

Parameter number (DDE)	Parameter name	Group 0	Group 1	Group 2	Process number	FB nr (par)	Var Type	Var Length	Min value	Max value	Read	Write	Poll	Secured	Highly Secured	Default Value	DDE str	Description
66	<CalRegFS2>	9			116	4	l		0	16777215	Yes	Yes	No	Yes	No	52A513	CalRegFS2	calibration register full scale input 2 ADC
67	ADC control register	9			114	4	l		0	4294967295	Yes	Yes	No	Yes	No	18904E	ADCcntrReg	ADC control register
68	Bridge potmeter	9			116	5	c		0	255	Yes	Yes	No	Yes	No	0	BridgePotm	sensor bridge zero potmeter setting
69	<AlarmEnble>	4			115	4	c		0	255	Yes	Yes	No	Yes	No	1	AlarmEnble	broadcast alarm messsage enable flag
70	Test mode	4			115	5	c		0	255	Yes	Yes	No	Yes	No	0	TestMode	test mode selection (not active until cntrlmode has been set to value 5)
71	<ADC channel select>	9			115	6	c		1	32	Yes	Yes	No	Yes	No	1	ChanSelect	channel selection ADC
72	Normal step controller response	8			114	5	c		0	255	Yes	Yes	No	Yes	No	0	ContrResp	controller response for normal steps (128=normal, <128=slower, >128=faster)
73	Setpoint exponential smoothing filter	6			117	3	f		0	1	Yes	Yes	No	Yes	No	1.0	ErrInFilCo	analog input filter constant (0=max, 1=min)
74	Sensor exponential smoothing filter	6			117	4	f		0	1	Yes	Yes	No	Yes	No	1.0	ExpSmooCon	sensor input filter constant (0=max, 1=min)
75	Analog output zero scale	10				21	i		-32767	65535	Yes	Yes	No	Yes	No	32767	AnOutCorZS	analog output correction factor zero scale (meas outp DSCM-A 0=0 other 32767=0)
76	Analog output full scale	10				22	i		0	65535	Yes	Yes	No	Yes	No	2000	AnOutCorFS	analog output correction factor full scale (meas outp 2000 = 1 * multiplication)
77	Analog input zero scale	10				23	i		-32767	65535	Yes	Yes	No	Yes	No	32767	AnInpCorZS	analog input correction factor zero scale (ext setp DSCM-A 0=0 other 32767=0)
78	Analog input full scale	10				24	i		0	65535	Yes	Yes	No	Yes	No	2000	AnInpCorFS	analog input correction factor full scale (ext setp 2000 = 1 * multiplication)
79	Tuning mode	7			115	7	c		0	255	Yes	Yes	No	Yes	No	0	TuningMode	(auto)tuning mode selection (not active until cntrlmode has been set to value 6)
80	Valve default	8			114	6	c		0	255	Yes	Yes	No	Yes	No	0	DefVlvType	valve type (needed for controlling behaviour)
81	Global modify	12			0	19	c		0	255	Yes	Yes	No	No	No	0	GlobModify	contains number(s) of changed processes for indirect polling (0xXX / 0xFF)
82	Valve span correction factor	8			114	7	f		0	1	Yes	Yes	No	Yes	No	0.1	SpanCorr	correction factor valve curve ratio high/low area
83	Valve curve correction	8			114	8	c	-2			Yes	Yes	No	Yes	No	20,80	VlvCrvStps	Valve curve correction for controller (max. factor*0.1, flow where factor = 1)
84	<MemShipNor>	8			114	9	c	-2			Yes	Yes	No	Yes	No	10,5000,10,5000	MemShipNor	array with memberships for normal Fuzzy controller
85	<MemShipOpn>	8			114	10	c	-2			Yes	Yes	No	Yes	No	,,10000,3750,2000(	MemShipOpn	array with memberships for 0-open Fuzzy controller
86	IO status	12	20		114	11	c		0	255	Yes	Yes	No	Yes	No	4	IOStatus	IO status byte for jumper settings and LED signal modes
87	<FuzzStNeNo>	8			114	13	c	-2			Yes	Yes	No	Yes	No	-30000,-500,-50	FuzzStNeNo	array with neg nor output steps for Fuzzy contr.
88	<FuzzStPoNo>	8			114	14	c	-2			Yes	Yes	No	Yes	No	50,500,25000	FuzzStPoNo	array with pos nor output steps for Fuzzy contr.
89	<FuzzStOpen>	8			114	15	c	-2			Yes	Yes	No	Yes	No	90,180,12000	FuzzStOpen	array with open at 0 output steps for Fuzzy contr.
90	Device type	13			113	1	c	6			Yes	No	No	No	No	DMFC	DeviceType	(FLOW-BUS) device type information string
91	BHTModel number	13			113	2	c	-2			Yes	Yes	No	Yes	No	F201C-FA	ModelNum	model number information string
92	Serial number	13			113	3	c	-2			Yes	Yes	No	Yes	Yes	SN999999A	SerialNum	serial number information string (to be changed by Bronkhorst only)
93	Customer model	11			113	4	c	-2			Yes	Yes	No	Yes	No	STANDARD	MfrConfig	manufacturing configuration information string
94	BHT1	14			118	1	c	-2			Yes	Yes	No	Yes	Yes	01,01,95	BHT1	special BHT parameter (to be changed by Bronkhorst only)
95	BHT2	14			118	2	i		0	65535	Yes	No	No	No	No	0	BHT2	special BHT parameter
96	BHT3	14			118	3	l		-3000000000	3000000000	Yes	No	No	No	No	0	BHT3	special BHT parameter
97	BHT4	14			118	4	i		0	65535	Yes	No	No	No	No	0	BHT4	special BHT parameter
98	BHT5	14			118	5	c		0	255	Yes	No	No	No	No	0	BHT5	special BHT parameter
99	BHT6	14			118	6	c		0	255	Yes	Yes	No	No	No	0	BHT6	special BHT parameter
100	BHT7	14			118	7	c		0	255	Yes	No	No	No	No	0	BHT7	special BHT parameter
101	BHT8	14			118	8	c		0	255	Yes	No	No	No	No	0	BHT8	special BHT parameter
102	BHT9	14			118	9	l		-3000000000	3000000000	Yes	No	No	No	No	0	BHT9	special BHT parameter
103	BHT10	14			118	10	c		0	1	No	Yes	No	Yes	Yes	0	BHT10	special BHT parameter (to be changed by Bronkhorst only)
104	Broadcast repeating time	8			114	16	c		0	255	Yes	Yes	No	Yes	No	128	PulseHight	broadcast repeating time (x5 ms) (old:height of open at 0 pulse train for valve)
105	Firmware version	13	20		113	5	c	6			Yes	No	No	No	No	VX.XX	Version	revision number of firmware
106	Pressure sensor type	20			115	9	c		0	255	Yes	Yes	No	Yes	No	0	PressSensr	type of pressure sensor
107	Barometer pressure	20			116	8	f		0	1200	Yes	Yes	No	No	No	1013.25	BaroPress	mbar atmospheric (central) barometer pressure
108	Sensor input zero scale	10				25	i		-32767	65535	Yes	Yes	No	Yes	No	32767	AnIn1CorZS	analog sensor signal input corr. factor zero scale (DSCM-A 0=0 other 32767=0)
109	Sensor input full scale	10				26	i		0	65535	Yes	Yes	No	Yes	No	2000	AnIn1CorFS	analog sensor signal input correction factor full scale (2000=1*multiplication)
110	Reference voltage input zero scale	10				27	i		-32767	65535	Yes	Yes	No	Yes	No	32767	AnIn2CorZS	analog Vref input correction factor zero scale (DSCM-A 0=0 other 32767=0)
111	Reference voltage input full scale	10				28	i		0	65535	Yes	Yes	No	Yes	No	2000	AnIn2CorFS	analog Vref input correction factor full scale (2000=1*multiplication)
112	Analog setpoint zero scale	10				29	i		-32767	65535	Yes	Yes	No	Yes	No	32767	AnOu1CorZS	analog setpoint output correction factor zero scale (DSCM-A 0=0 other 32767=0)
113	Analog setpoint full scale	10				30	i		0	65535	Yes	Yes	No	Yes	No	2000	AnOu1CorFS	analog setpoint output correction factor full scale (2000=1*multiplication)
114	Reset	12			115	8	c		0	255	No	Yes	No	No	No	0	Reset	reset facilities (program/alarm/batchcounter)
115	User tag	11			113	6	c	-2			Yes	Yes	No	No	No	USERTAG	UserTag	user definable alias string
116	Alarm limit maximum	15			97	1	i		0	41600	Yes	Yes	No	No	No	0	AlrmMaxLim	maximum limit for sensor signal to trigger alarm situation
117	Alarm limit minimum	15			97	2	i		0	41600	Yes	Yes	No	No	No	0	AlrmMinLim	minimum limit for sensor signal to trigger alarm situation
118	Alarm mode	15			97	3	c		0	255	Yes	Yes	No	No	No	0	AlrmMode	alarm mode
119	Alarm output mode	15			97	4	c		0	255	Yes	Yes	No	No	No	0	AlrmOutMod	alarm relais activity mode during alarm situation
120	Alarm setpoint mode	15			97	5	c		0	1	Yes	Yes	No	No	No	0	AlrmStpMod	setpoint change enable during alarm situation
121	Alarm new setpoint	15			97	6	i		0	32767	Yes	Yes	No	No	No	0	AlrmNwSetp	new/safe setpoint during alarm situation (until reset)
122	Counter value	16			104	1	f		0	3.40282E+38	Yes	Yes	Yes	No	No	0	CntrValue	actual counter value
123	Counter unit index	16			104	2	c		0	31	Yes	Yes	No	No	No	0	CntrUnit	pointer to counter unit (classic counter unit table)
124	Counter limit	16			104	3	f		0	3.40282E+38	Yes	Yes	No	No	No	0	CntrLimit	counter limit/batch
125	Counter output mode	16			104	4	c		0	255	Yes	Yes	No	No	No	0	CntrOutMod	counter relais activity mode when limit/batch has been reached
126	Counter setpoint mode	16			104	5	c		0	1	Yes	Yes	No	No	No	0	CntrStpMod	setpoint change enable during counter limit/batch situation (until reset)
127	Counter new setpoint	16			104	6	i		0	32767	Yes	Yes	No	No	No	0	CntrNwSetp	new/safe setpoint at counter limit/batch situation (until reset) (normally = 0%)
128	Counter unit	16			104	7	c	4			Yes	Yes	No	No	No	In	CntrUntstr	counter readout unit (informative only for older devices)
129	Capacity unit	3	19			31	c	7			Yes	Yes	No	Yes	No	l/min	capunitstr	capacity readout unit (informative only for older devices)
130	Counter mode	16			104	8	c		0	255	Yes	Yes	No	No	No	0	CntrMode	counter mode
131	Minimum hardware revision	13			113	7	c	1			Yes	No	No	No	No	VX.XX	HwRev	minimum required hardware revision level for firmware version
132	<RCreadfact>	17				1	f		1E-10	10000000000	Yes	No	No	No	No	1.0	RCreadfact	readout factor for direct reading (changes with readunit: local on module, R.O.)
133	<channumber>	17				2	c		1	120	Yes	Yes	No	No	No	1	channumber	channel number for operation
134	<masterchan>	17				3	c		0	120	Yes	Yes	No	No	No	0	masterchan	master channel for master-slave operation
135	<RCslavefct>	17				4	i		0	32000	Yes	Yes	No	No	No	32000	RCslavefct	RC slave factor

Parameter number (DDE)	Parameter name	Group 0	Group 1	Group 2	Process number	FB nr (par)	Var Type	Var Length	Min value	Max value	Read	Write	Poll	Secured	Highly Secured	Default Value	DDE str	Description
136	<inputnode>	17				5	c		0	128	Yes	Yes	No	Yes	No	3	inputnode	physical node address for channel number
137	<inputproc>	17				6	c		0	128	Yes	Yes	No	Yes	No	1	inputproc	physical process for channel number
138	<RCreadunit>	17				7	c	7			Yes	No	No	No	No	ln/min	RCreadunit	readout unit for direct reading (local variable on module: read only)
139	Slave factor	18			33	1	f		0	500	Yes	Yes	No	No	No	100.0	SlaveFact%	slave factor for master slave control (setp = master output * slave factor)
140	Reference voltage input	18			33	2	i		0	65535	Yes	No	Yes	No	No	0	VrefInput	reference voltage input for setpoint signal
141	Stable situation controller response	8			114	17	c		0	255	Yes	Yes	No	Yes	No	0	RespStable	controller response when controller is stable: |measure-setpoint| < 2%
142	Temperature	19	13		33	7	f		-250	500	Yes	Yes	Yes	No	No	20	temperatur	absolute temperature in degrees Celsius
143	Pressure	19	13		33	8	f		-3.40282E+38	3.40282E+38	Yes	Yes	Yes	No	No	1013.25	pressure	absolute pressure in mbar
144	Time	19			33	9	f		0	3.40282E+38	Yes	No	Yes	No	No	0	time	time in milliseconds
145	Calibrated volume	19			33	10	f		0	3.40282E+38	Yes	Yes	No	Yes	No	50	calvolume	calibrated volume in litres
146	Sensor number	19				16	c		0	4	Yes	Yes	Yes	No	No	0	sensornr	pointer to sensor number in calibration tube (FPP)
147	Range select	20	19		115	10	c		0	99	Yes	Yes	Yes	No	No	0	rangeselct	Piston Prover operation mode (write) and status information (read back)
148	Time out	20				2	i		0	30000	Yes	Yes	No	Yes	No	0	TimeOut	maximum admitted duration time for specific procedure (in 100 ms)
149	Frequency	21			33	9	f		0	100000	Yes	No	Yes	No	No	0	frequency	frequency in Hz
150	Impulses/m3	20			33	10	f		0	3.40282E+38	Yes	Yes	No	Yes	No	42773.4	imp/m3	For FRM and FTM imp/m3 and for FCM imp/kg
151	Normal volume flow	19			33	5	f		0	3.40282E+38	Yes	No	No	No	No	0	RefVolFlow	volume flow referenced to normal conditions i.e. 0 ℃, 1013.25 hPa(a) in ln/min
152	Volume flow	19			33	6	f		-3.40282E+38	3.40282E+38	Yes	No	No	No	No	0	volumeflow	volume flow at actual conditions in l/min
153	Delta-p	19			33	11	f		-100000	100000	Yes	Yes	Yes	No	No	0	delta-p	relative pressure between atmosphere and sensor position
154	<scalefact>	21			33	13	i		1	10000	Yes	No	No	No	No	1	scalefact	scaling factor (multiplication) for readout on display (for optimal resolution)
155	Sensor name	19				17	c	10			Yes	Yes	No	Yes	No	SENSOR0	sensorname	label with information about stop sensor
156	Reset alarm enable	15			97	9	c		0	255	Yes	Yes	No	No	No	15	RstAlarmEn	enable reset of alarm by: keyboard, external signal, FLOW-BUS, automatic
157	Reset counter enable	16			104	9	c		0	255	Yes	Yes	No	No	No	7	RstCountEn	enable reset of counter by: keyboard, external signal, FLOW-BUS, automatic
158	Master node	18			33	14	c		1	128	Yes	Yes	No	No	No	3	MasterNode	node number of master instrument output signal for a slave
159	Master process	18			33	15	c		1	128	Yes	Yes	No	No	No	1	MasterProc	process number of master instrument output signal for a slave
160	Remote instrument node	18			33	16	c		1	128	Yes	Yes	No	Yes	No	3	InstrNode	node number of instrument to be operated by another module (keyboard/display)
161	Remote instrument process	18			33	17	c		1	128	Yes	Yes	No	Yes	No	1	InstrProc	process number of instrument to be operated by another module (keyboard/display)
162	Minimum custom range	3			33	18	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0.0	RangeMin	Mnimum value at 0% for special user readout unit
163	Maximum custom range	3			33	20	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	100.0	RangeMax	Maximum value at 100% for special user readout unit
164	Relay/TTL output	12			115	11	c		0	255	Yes	Yes	No	Yes	No	0	Relay/TTL	Relay/TTL output setting (disabled when used by alarm or counter)
165	Open from zero controller response	8			114	18	c		0	255	Yes	Yes	No	Yes	No	0	RespOpen0%	Controller response when valve opens from zero
166	Controller features	8			114	20	c		0	255	Yes	Yes	No	Yes	No	1	ContrType	Controller settings for special purpose
167	PID-Kp	8	3		114	21	f		0	3.40282E+38	Yes	Yes	No	Yes	No	10	PIDKp	PID factor Kp
168	PID-Ti	8			114	22	f		0	3.40282E+38	Yes	Yes	No	Yes	No	0.05	PIDTi	PID factor Ti
169	PID-Td	8			114	23	f		0	3.40282E+38	Yes	Yes	No	Yes	No	0	PIDTd	PID factor Td
170	Density	3			33	21	f		0	3.40282E+38	Yes	Yes	No	Yes	No	1.293	Density	Density of selected fluid in kg/m3
171	Calibration certificate	13	3		113	8	c	-2			Yes	Yes	No	Yes	Yes		CalCertNr	Number of calibration certificate (last basic calibration)
172	Calibration date	13	3		113	9	c	8			Yes	Yes	No	Yes	Yes	19991231	CalDate	Date of last (basic) calibration
173	Service number	13			113	10	c	15			Yes	Yes	No	Yes	Yes	00000000	ServiceNr	Service number for repair/rebuilding/recalibration
174	Service date	13			113	11	c	8			Yes	Yes	No	Yes	Yes	19991231	ServDate	Date of last service action
175	Identification number	13			113	12	c		0	255	Yes	Yes	No	Yes	Yes	7	IdentNr	Identification number (type) of instrument/device
176	BHT11	14			118	11	c		0	255	No	Yes	No	Yes	Yes	0	BHT11	special BHT parameter (to be changed by Bronkhorst only)
177	Power mode	12			115	12	c		0	50	Yes	Yes	No	Yes	No	0	PowerMode	power supply indication in Vdc
178	Pressure inlet	13	3		113	13	f		-100000	100000	Yes	Yes	No	Yes	No	3	Pupstream	pressure inlet (upstream) of fluid in bara (for first fluidnr only)
179	Pressure outlet	13	3		113	14	f		-100000	100000	Yes	Yes	No	Yes	No	1	Pdownstrm	pressure outlet (downstream) of fluid in bara (for first fluidnr only)
180	Orifice	13			113	15	f		0	1000	Yes	Yes	No	Yes	No	1	Orifice	orifice diameter in mm
181	Fluid temperature	13	3		113	16	f		-273.15	3.40282E+38	Yes	Yes	No	Yes	No	20	FluidTemp	temperature of fluid through instrument (for first fluidnr only)
182	Alarm delay	15			97	7	c		0	255	Yes	Yes	No	No	No	3	AlrmDelay	time alarm and reset action will be delayed when alarm limit has been exceeded
183	Capacity 0%	3			33	22	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0	capacity0%	capacity of instrument at zero 0% in sensor base units (mostly equal to zero)
184	Number of channels	12			0	18	c		0	120	Yes	No	No	No	No	1	NumOfChan	number of instrument channels available for this device
185	Device function	12			0	20	c		0	255	Yes	No	No	No	No	5	DeviceFunc	function of device
186	Scan channel	4			123	1	c		1	255	Yes	Yes	No	No	No	1	ScanChan	Channel number to scan with real time information (to be set once)
187	Scan parameter	4			123	3	c		0	255	Yes	Yes	No	No	No	8	ScanPar	Parameter number to scan with real time information (to be set once)
188	Scan time	4			123	4	i		0	65535	Yes	Yes	No	No	No	50	ScanTime	Scan interval time in msec between two samples (to be set once)
189	Scan data	22	4		123	10	c	-2			Yes	No	No	No	No		ScanData	Scanned data with time label (can be readout event by event)
190	Valve open	8			114	24	f		0	24	Yes	Yes	No	Yes	No	0.04	ValveOpen	First-step offset current/voltage for valve when opening from 0%
191	Number of runs	20			115	13	c		0	255	Yes	Yes	No	No	No	1	NrOfRuns	Amount of runs of a piston prover (0 = stability check)
192	Minimum process time	20			115	14	c		0	255	Yes	Yes	No	No	No	10	MinProTime	Minimum process time of a piston prover in 0.1 seconds
193	Leak rate	20			116	9	f		0	1	Yes	Yes	No	Yes	No	0.0001	LeakRate	Leak rate piston prover
194	Mode info request	12			115	15	c	4			Yes	Yes	No	Yes	No		ModeInfReq	Sets instr. in info mode for 1 read-cycle to check available parameter options
195	Mode info option list	12			115	16	c	255			Yes	No	No	No	No		ModeInfOpt	Gives info about possible values of a mode in an array as result of ModeInfo req
196	Mode info option description	12			115	17	c	255			Yes	No	No	No	No		ModeInfDes	Gives description about one of the mode options
197	Calibrations options	20			115	18	c		0	255	Yes	Yes	No	Yes	No	0	CalType	Enables/disables options for calibration device (8 bits for 8 options)
198	Mass flow	20			33	4	f		-3.40282E+38	3.40282E+38	Yes	No	Yes	No	No	0	MassFlow	Real mass flow in kg/min
199	Bus address	23			125	10	c		0	255	Yes	Yes	No	Yes	No	2	BusAddress	Fieldbus address (top interface)
200	Interface configuration	23			125	3	c		0	3	Yes	Yes	No	No	Yes	1	InterfConf	Configuration setting for interface to other bus-systems
201	Baudrate	23			125	9	l		0	10000000000	Yes	Yes	No	Yes	No	12000000	Baudrate	Fieldbus baudrate (top interface)
202	Bus diagnostic string	23			125	20	c	-2			Yes	No	No	No	No		BusDiagnos	Fieldbus diagnose string (top interface)
203	Number of vanes	20			115	22	c		0	255	Yes	Yes	No	Yes	No	10	NrOfVanes	Number of vanes for use in a rotor meter
204	Fieldbus	23			125	21	c	-2			Yes	No	No	No	No	FLOW-BUS	Fieldbus	Fieldbus name (top interface)
205	fMeasure	2			33	0	f		-3.40282E+38	3.40282E+38	Yes	No	Yes	No	No	0	fMeasure	measured value for direct reading (in capunits, max.= capacity)

Parameter number (DDE)	Parameter name	Group 0	Group 1	Group 2	Process number	FB nr (par)	Var Type	Var Length	Min value	Max value	Read	Write	Poll	Secured	Highly Secured	Default Value	DDE str	Description
206	fSetpoint	2	18		33	3	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	No	No	0	fSetpoint	setpoint: wanted value for direct reading (in capunits, max.= capacity)
207	Mass	20			33	23	f		-3.40282E+38	3.40282E+38	Yes	No	Yes	No	No	0	Mass	Mass in g
208	Manufacturer status register	4			119	1	c	8			Yes	No	No	No	No		Mstatus	Manufacturer Status register (64 diagnostic bits)
209	Manufacturer warning register	4			119	2	c	8			Yes	No	No	No	No		Mwarning	Manufacturer Warning register (64 diagnostic bits)
210	Manufacturer error register	4			119	3	c	8			Yes	No	No	No	No		Merror	Manufacturer Error register (64 diagnostic bits)
211	Diagnostic history string	4			119	4	c	-2			Yes	Yes	No	No	No		DiagHist	Diagnostic history string (contains history of diag codes)
212	Diagnostic mode	4			119	5	c		0	255	Yes	Yes	No	Yes	No	0	DiagMode	Diagnostic mode (0 = diagnostics off, 1 = diagnostics on)
213	Manufacturer status enable	4			119	6	c		0	255	No	Yes	No	No	No	0	MStatEnabl	Manufacturer Status enable (0-127 or 254 = disable all, 255 = enable all)
214	Analog output zero adjust	10			116	21	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0	AnOutZA	Analog measure output, zero adjust
215	Analog output span adjust	10			116	22	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	1	AnOutSA	Analog measure output, span adjust
216	Analog input zero adjust	10			116	23	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0	AnInZA	Analog setpoint input, zero adjust
217	Analog input span adjust	10			116	24	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	1	AnInSA	Analog setpoint input, span adjust
218	Sensor input zero adjust	10			116	25	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0	SensInZA	Sensor input, zero adjust
219	Sensor input span adjust	10			116	26	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	1	SensInSA	Sensor input, span adjust
220	Temperature input zero adjust	10			116	27	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0	TempInZA	Sensor temperature input, zero adjust
221	Temperature input span adjust	10			116	28	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	1	TempInSA	Sensor temperature input, span adjust
222	Adaptive smoothing factor	6			117	5	f		0	1	Yes	Yes	No	Yes	No	1.0	ExpSmooAd	Sensor input filter adapt setting
223	Slope setpoint step	18			33	24	i		0	32000	Yes	Yes	No	Yes	No	32000	SlopeSetp	Slope setpoint step. Setpoint step for the given slopetime
224	Filter length	6			117	6	i		0	255	Yes	Yes	No	Yes	No	1	FilterLen	Number of samples for average filter
225	Absolute accuracy	2			33	25	f		-3.40282E+38	3.40282E+38	Yes	No	No	No	No	0	fAccuracy	Actual accuracy in current unit
226	Lookup table index	3			33	26	c		0	255	Yes	Yes	No	No	No	0	LookI	Lookup table for linearisation index (x and y direction)
227	Lookup table X	3			33	27	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0	LookX	Lookup table for linearisation x
228	Lookup table Y	3			33	28	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0	LookY	Lookup table for linearisation y
229	Lookup table temperature index	3			33	29	c		0	255	Yes	Yes	No	No	No	0	LookTempI	Lookup table for linearisation at certain temperature index (z direction)
230	Lookup table temperature	3			33	30	f		-273.15	3.40282E+38	Yes	Yes	No	Yes	No	0	LookTemp	Lookup table for linearisation at certain temperature (z)
231	Valve maximum	8			114	25	f		0	24	Yes	Yes	No	Yes	No	0.2	ValveMax	Maximum current/voltage for valve
232	Valve mode	8			114	26	c		0	255	Yes	Yes	No	Yes	No	1	ValveMode	Valve output mode selection
233	Valve open correction	8			114	27	f		0	1.5	Yes	Yes	No	Yes	No	0.96	VlvOpenCor	Valve open current/voltage correction (example: 0.96, Open = ValveOpen * 0.96)
234	Valve zero hold	8			114	28	f		0	1	Yes	Yes	No	Yes	No	0	VlvZeroHld	Valve hold current/voltage at %0 setp (example: 0.8, Hold = ValveOpen * 0.8)
235	Valve slope	8			114	29	f		0	50	Yes	Yes	No	Yes	No	0.009	ValveSlope	Valve slope time (seconds)
236	IFI data	23			0	21	c	-2			Yes	Yes	No	No	No	0	IFIData	IFI data dump protocol communication string
237	Range used	20	19		115	20	c	-2	0	99	Yes	No	Yes	No	No	0	RangeUsed	Piston Prover information about used sensors
238	Fluidset properties	3			33	31	c		0	255	Yes	Yes	No	Yes	Yes	0	FldSetProp	Fluidset properties
239	Lookup table unit type index	3			33	12	c		0	255	Yes	Yes	No	Yes	Yes	0	LUnitType	Lookup table unit type
240	Lookup table unit type	3			33	13	c	20			Yes	Yes	No	Yes	Yes		LUnTypName	Lookup table unit type name
241	Lookup table unit index	3			33	16	c		0	255	Yes	Yes	No	Yes	Yes	0	LUnit	Lookup table unit (unit LUTy)
242	Lookup table unit	3			33	17	c	7			Yes	Yes	No	Yes	Yes	kg/s	LUnitName	Lookup table unit name
243	Capacity unit type index	3				29	c		0	255	Yes	Yes	No	Yes	No	0	CUnitType	Capacity (readout) unit type
244	Capacity unit type	3				30	c	20			Yes	Yes	No	Yes	No		CUnTypName	Capacity (readout) unit type name
245	Capacity unit type temperature	3			33	10	f		-273.15	3.40282E+38	Yes	Yes	No	Yes	No	0	CUnTypTemp	Capacity (readout) unit type temperature (℃)
246	Capacity unit pressure	3			33	11	f		0	3.40282E+38	Yes	Yes	No	Yes	No	1	CUnTypPres	Capacity (readout) unit type pressure (bar (a))
247	Capacity minimum	3				27	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0	CapMin	Minimum capacity in output capacity units
248	Capacity maximum	3				28	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	1	CapMax	Maximum capacity in output capacity units
249	Formula type	3			113	17	i		0	65535	Yes	Yes	No	Yes	No	0	FormulaTyp	Formula type needed for conversion
250	Heat capacity	3			113	18	f		0	3.40282E+38	Yes	Yes	No	Yes	No	0	HeatCap	Heat capacity (Cp) (sensor conditions)
251	Thermal conductivity	3			113	20	f		0	3.40282E+38	Yes	Yes	No	Yes	No	0	ThermCond	Thermal conductivity (sensor conditions)
252	Viscosity	3			113	21	f		0	3.40282E+38	Yes	Yes	No	Yes	No	0	Viscosity	Dynamic viscosity (fluid conditions)
253	Standard flow	3			113	22	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	1	NormMasFlw	Standard mass flow in ln/min (20℃, 1.01325 bar (a)) air or g/h H2O equivalent
254	Controller speed	8	3		114	30	f		0	3.40282E+38	Yes	Yes	No	Yes	No	1	Kspeed	Controller speed factor (gain)
255	Sensor code	13			113	23	i		0	65535	Yes	Yes	No	Yes	Yes	0	SensorCode	Sensor code
256	Sensor configuration code	13			113	24	c		0	255	Yes	Yes	No	Yes	Yes	0	SensorRevC	Sensor configuration code
257	Restriction code	13			113	25	i		0	65535	Yes	Yes	No	Yes	Yes	0	RestrCode	Restriction code
258	Restriction configurator code	13			113	26	c		0	255	Yes	Yes	No	Yes	Yes	0	RestrRevC	Restriction configuration code
259	Restriction NxP	13			113	27	l		0	2147483648	Yes	Yes	No	Yes	Yes	0	RestrNxP	Restriction NxP (proportional to air equivalent capacity of LFE)
260	Seals information	13			113	28	c	16	0	255	Yes	Yes	No	Yes	Yes	V,V	Seals	Seals information (1st byte = other, 2nd = plunger seal)
261	Valve code	13			113	29	i		0	65535	Yes	Yes	No	Yes	Yes	0	ValveCode	Valve code
262	Valve configuration code	13			113	30	c		0	255	Yes	Yes	No	Yes	Yes	0	ValveRevC	Valve configuration code
263	Instrument properties	13			113	31	l		0	2147483648	Yes	Yes	No	Yes	Yes	0	InstrProp	Instrument properties
264	Lookup table frequency index	3			116	10	c		0	1	Yes	Yes	No	Yes	No	0	LookFreqI	Lookup table for frequency index
265	Lookup table frequency frequency	3			116	11	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0	LFFreq	Lookup table for frequency frequency
266	Lookup table frequency temperature	3			116	12	f		-273.15	3.40282E+38	Yes	Yes	No	Yes	No	0	LFTemp	Lookup table for frequency temperature
267	Lookup table frequency density	3			116	13	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0	LFDensity	Lookup table for frequency density
268	Lookup table frequency span adjust	3			116	14	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0	LFSpanAdj	Lookup table for frequency span adjust
269	Capacity unit index (ext)	3	19		65	15	c		0	255	Yes	Yes	No	Yes	No	0	CUnit	Capacity (readout) unit index (extended unit table)
270	Density actual	3			116	15	f		-3.40282E+38	3.40282E+38	Yes	No	Yes	No	No	0	DensityAct	Actual density, measured by instrument
271	Measured restriction	13			116	18	f		0	3.40282E+38	Yes	Yes	No	Yes	No	0	RestrMeas	Measured restriction
272	Temperature potmeter	3			116	8	c		0	255	Yes	Yes	No	Yes	No	0	TempPotm	Potmeter for sensor temperature compensation
273	Temperature potmeter gain	3			116	9	c		0	255	Yes	Yes	No	Yes	No	0	TempGain	Gain for sensor temperature compensation
274	Counter controller overrun correction	16			104	10	f		0	3.40282E+38	Yes	Yes	No	No	No	1	CntrCConv	Counter controller overrun correction
275	Counter controller gain	16			104	11	f		0	3.40282E+38	Yes	Yes	No	No	No	1	CntrCGain	Counter controller gain

Parameter number (DDE)	Parameter name	Group 0	Group 1	Group 2	Process number	FB nr (par)	Var Type	Var Length	Min value	Max value	Read	Write	Poll	Secured	Highly Secured	Default Value	DDE str	Description
276	Sub fluid number	3			65	1	c		0	255	Yes	Yes	No	No	No	0	FluidSub	Sub fluid number
277	Temperature compensation factor	9			116	17	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0	FreqTCor	Temperature compensation factor
278	DSP register address	13			116	29	l		0	4294967295	Yes	Yes	No	No	No	0	DSPRegI	DSP register address
279	DSP register long	13			116	30	l		-4294967295	4294967295	Yes	Yes	No	Yes	No	0	DSPRegLng	DSP register long
280	DSP register floating point	13			116	30	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0	DSPRegFlt	DSP register floating point
281	DSP register integer	13			116	31	i		-65535	65535	Yes	Yes	No	Yes	No	0	DSPRegInt	DSP register integer
282	Standard deviation	19			121	0	f		-3.40282E+38	3.40282E+38	Yes	No	Yes	No	No	0	StdDev	Standard deviation
283	Measurement status	19			121	1	i		0	65535	Yes	No	No	No	No	0	MeasStatus	Measurement status
284	Measurement stop criteria	19			121	2	i		0	65535	Yes	No	No	No	No	0	MStopCrit	Measurement stop criteria
285	Measurement time out	20			121	3	i		0	65535	Yes	Yes	No	No	No	0	MTimeOut	Measurement time out
286	Maximum number of runs	20			121	4	i		0	65535	Yes	Yes	No	No	No	0	MMaxNrRuns	Measurement maximum number of runs
287	Minimum standard deviation	20			121	5	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	No	No	0	MMinStdDev	Measurement minimum standard deviation
288	IO switch status	10			114	31	l		0	4294967295	Yes	Yes	No	No	No	0	IOSwitchSt	IO status for switches
295	Sensor bridge settings	9			65	21	i		0	65535	Yes	Yes	No	Yes	Yes	513	SensBridge	Sensor bridge settings
296	Sensor bridge current	9			65	22	f		0	3.40282E+38	Yes	Yes	No	Yes	Yes	0	SensCurren	Sensor bridge current (0..0.02A)
297	Sensor resistance	9			65	23	f		0	3.40282E+38	Yes	No	No	No	No	0	SensResist	Sensor resistance (Ohm)
298	Sensor bridge voltage	9			65	24	f		0	3.40282E+38	Yes	No	No	No	No	0	SensVolt	Sensor bridge voltage (0..11.5V)
299	Sensor group name	9			65	25	c	-2			Yes	No	No	No	No		SensName	Sensor name (based on sensor detection / sensor resistance)
300	Sensor calibration temperature	9			116	20	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0	SensCalTmp	Sensor temperature at calibration
301	Valve safe state	8			115	31	c		0	255	Yes	Yes	No	No	No	0	ValveSafe	Valve safe state
302	Counter unit type index	16			104	12	c		0	255	Yes	Yes	No	No	No	0	CnUnitType	Counter unit type
303	Counter unit type	16			104	13	c	20			Yes	Yes	No	No	No		CnUnTypNam	Counter unit type name
304	Counter unit index (ext)	16			104	14	c		0	255	Yes	Yes	No	No	No	0	CnUnit	Counter unit index (extended counter unit table)
305	Bus1 selection	23			125	8	c		0	255	Yes	Yes	No	Yes	No	255	Bus1Select	Fieldbus select
306	Bus1 medium	23			125	11	c		0	255	Yes	Yes	No	Yes	No	1	Bus1Medium	Fieldbus medium id
307	Bus2 mode	23			124	7	c		0	255	Yes	Yes	No	Yes	No	0	Bus2Mode	Fieldbus2 mode
308	Bus2 selection	23			124	8	c		0	255	Yes	Yes	No	Yes	No	2	Bus2Select	Fieldbus2 select
309	Bus2 address	23			124	10	c		0	255	Yes	Yes	No	Yes	No	2	Bus2Addr	Fieldbus2 address (side interface)
310	Bus2 baudrate	23			124	9	l		0	10000000000	Yes	Yes	No	Yes	No	12000000	Bus2Baud	Fieldbus2 baudrate (side interface)
311	Bus2 medium	23			124	11	c		0	255	Yes	Yes	No	Yes	No	0	Bus2Medium	Fieldbus2 medium id
312	Bus2 diagnostics	23			124	20	c	-2			Yes	No	No	No	No		Bus2Diagn	Fieldbus2 diagnose string (side interface)
313	Bus2 name	23			124	21	c	-2			Yes	No	No	No	No	Propar	Bus2Name	Fieldbus2 name (side interface)
314	PIO channel selection	10			120	0	c		0	255	Yes	Yes	No	No	No	0	PIOChSel	PIO channel selection
315	PIO parameter	10			120	2	i		0	65535	Yes	Yes	No	No	No	0	PIOParm	PIO parameter connected to i/o channel (=process number*256+parameter number)
316	PIO input/output filter	10			120	6	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	No	No	0	PIOFilter	PIO input/output filter constant (0..1, 0=max, 1=min)
317	PIO parameter capacity 0%	10			120	7	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	No	No	0	PIOPrmCap0	PIO parameter capacity, 0% value
318	PIO parameter capacity 100%	10			120	3	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	No	No	0	PIOPrmCap	PIO parameter capacity, 100% value
319	PIO configuration selection	10			120	1	c		0	255	Yes	Yes	No	No	No	0	PIOCfgSel	PIO i/o channel configuration selection
320	PIO analog zero adjust	10			120	4	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0	PIOAnZA	PIO analog input/output, zero adjust value
321	PIO analog span adjust	10			120	5	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0	PIOAnSA	PIO analog input/output, span adjust factor
322	PIO hardware capacity max	10			120	8	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	No	No	0	PIOHwCapMx	PIO max hardware capacity (max capacity value in i/o units, V, A, Hz, etc.)
323	PIO capacity set selection	10			120	9	c		0	255	Yes	Yes	No	No	No	0	PIOCapSel	PIO capacity set selection
324	PIO hardware capacity 0%	10			120	10	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	No	No	0	PIOHwCap0	PIO hardware capacity, 0% value (capacity value in i/o units, V, A, Hz, etc.)
325	PIO hardware capacity 100%	10			120	11	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	No	No	0	PIOHwCap	PIO hardware capacity, 100% value (capacity value in i/o units, V, A, Hz, etc.)
326	Hardware platform id	12			0	6	i		0	65535	Yes	No	No	No	No	0	HardwId	Hardware platform identification number
327	Hardware platform sub id	12			0	7	c		0	255	Yes	No	No	No	No	0	HardwSubId	Hardware platform sub identification number
328	Temporary baudrate	23			124	31	l		0	10000000000	Yes	Yes	No	No	No	0	TempBaud	Temporary volatile Fieldbus2 baudrate (side interface)
329	Setpoint monitor mode	2			115	23	c		0	255	Yes	Yes	No	Yes	No	0	SetMonitor	Monitor mode for setpoint
330	BHT12	14			118	12	l		0	10000000000	Yes	No	No	No	No	0	BHT12	Special BHT parameter
331	Nominal sensor voltage	9			65	26	f		0	3.40282E+38	Yes	Yes	No	Yes	No	0	SenNomVolt	Nominal sensor voltage (used for sensor temperature voltage compensation)
332	Sensor voltage compensation factor	9			116	16	f		-3.40282E+38	3.40282E+38	Yes	Yes	No	Yes	No	0	VoltTCor	Sensor voltage compensation factor (used for sensor temperature voltage comp.)
333	PCB serial number	4			119	31	c	-2			Yes	Yes	No	Yes	Yes		PCBSerial	PCB serial number (unique number or string written by PCB manufacturer)
334	Minimum measure time	20			115	24	c		0	255	Yes	Yes	No	No	No	0	MinMTime	Minimum measure time between sensors of a piston prover in 0.1 seconds
335	Bus1 parity	23			125	12	c		0	255	Yes	Yes	No	Yes	No	0	Bus1Parity	Fieldbus parity
336	Bus2 parity	23			124	12	c		0	255	Yes	Yes	No	Yes	No	0	Bus2Parity	Fieldbus2 parity (side interface)
337	Firmware id	12			0	8	c		0	255	Yes	No	No	No	No	0	FirmwId	Firmware identification number