

◆ ◆ ◆ ◆ ◆
◆ ◆ ◆ ◆ ◆

Benutzerhandbuch

◆ ◆ ◆ ◆ ◆
◆ ◆ ◆ ◆ ◆

RS232-Schnittstelle mit ProPar Protokoll für digitale Multibus- Massendurchfluss- und Druckmesser/-regler

Dok. Nr.: 9.19.027W Datum: 11-12-2018

◆ ◆ ◆ ◆ ◆
◆ ◆ ◆ ◆ ◆
◆ ◆ ◆ ◆ ◆
◆ ◆ ◆ ◆ ◆

ACHTUNG
Es wird empfohlen, das vorliegende Benutzerhandbuch vor dem Einbau
und vor der Inbetriebnahme des Produktes sorgfältig zu lesen.
Die Nichtbeachtung der Anleitung kann Personenschäden
und/oder Beschädigungen der Anlage zur Folge haben.

◆ ◆ ◆ ◆ ◆
◆ ◆ ◆ ◆ ◆

Haftungsausschluss

Auch wenn die Informationen in diesem Handbuch geprüft wurden und als vollkommen zuverlässig erachtet werden, übernehmen wir keine Verantwortung für Ungenauigkeiten. Die Angaben in diesem Handbuch dienen lediglich der Information und können ohne vorherige Ankündigung geändert werden.

Urheberrecht

Alle Rechte vorbehalten. Dieses Dokument ist urheberrechtlich geschützt. Technische und darstellerische Änderungen sowie Änderungen durch Druckfehler vorbehalten. Die in diesem Dokument enthaltenen Informationen können jederzeit ohne Vorankündigung geändert werden. Bronkhorst High-Tech BV behält sich das Recht auf Produktänderungen und -verbesserungen vor, ohne sich verpflichtet zu fühlen nähere Angaben an Personen oder Organisationen zu machen. Die Gerätespezifikationen und der Verpackungsinhalt kann von den Ausführungen in diesen Dokument abweichen.

Symbole



Wichtige Informationen. Die Nichtbeachtung dieser Informationen könnte Verletzungen von Personen oder Schäden am Instrument oder an der Installation zur Folge haben.



Hilfreiche Informationen. Diese Informationen erleichtern die Verwendung des Instruments.



Zusätzliche Informationen erhalten Sie im Internet oder von unserem lokalen Vertriebspartner.

Gewährleistung

Für Produkte von Bronkhorst® gilt eine Gewährleistung für Material- und Verarbeitungsfehler für einen Zeitraum von 3 Jahren ab dem Versanddatum, vorausgesetzt, dass das Produkt entsprechend den Bestellspezifikationen verwendet und weder unsachgemäßem Gebrauch noch Schäden durch mechanische Einwirkungen ausgesetzt wird. Produkte, die nicht einwandfrei funktionieren, können während der Gewährleistungsfrist kostenlos repariert oder ausgetauscht werden. Für Reparaturen gilt in der Regel eine Gewährleistungsfrist von einem Jahr, es sei denn, die restliche Gewährleistungsfrist ist länger.



*Siehe auch Artikel 9 der Allgemeinen Verkaufs- und Lieferbedingungen:
http://www.bronkhorst.com/files/corporate_headquarters/sales_conditions/d_allgemeine_lieferbedingungen.pdf*

Die Gewährleistung gilt für alle offenen und verdeckten Mängel, Zufallsfehler und nicht bestimmbare Ursachen.

Ausgeschlossen sind Störungen und Schäden, die vom Kunden verursacht wurden, wie z.B. Kontaminationen, fehlerhafter elektrischer Anschluss, mechanische Einwirkungen usw.

Für die Wiederherstellung von Produkten, die zur Reparatur eingesandt wurden, bei denen ein Gewährleistungsanspruch nicht oder nur teilweise besteht, werden die Kosten entsprechend in Rechnung gestellt.

Bronkhorst High-Tech B.V. oder ein mit ihr verbundenes Unternehmen trägt die Versandkosten für ausgehende Sendungen von Geräten und Teilen, die im Rahmen unserer Gewährleistung verschickt werden, sofern im Voraus nichts anderes vereinbart wurde. Erfolgt die Anlieferung in unserem Werk oder bei unserer Servicestelle unfrei, werden die Versandkosten den Reparaturkosten hinzugeschlagen. Import- und/oder Exportabgaben sowie Kosten ausländischer Versandarten/Speditionen trägt der Kunde.

INHALTSVERZEICHNIS

1	ALLGEMEINE PRODUKTINFORMATIONEN.....	5
1.1	EINFÜHRUNG	5
1.2	MULTIBUSTYPEN.....	5
1.3	VERWEISE AUF ANDERE ANWENDBARE DOKUMENTE	6
1.3.1	Handbücher und Benutzeranleitungen	6
1.3.2	Technische Zeichnungen	6
1.3.3	Softwaretools.....	6
1.4	START IN KURZFORM	7
2	SCHNITTSTELLEN	8
2.1	RS232/FLOW-BUS-SCHNITTSTELLE	8
2.1.1	D-Steckverbinder für RS232	8
2.2	RS232 AM MULTIBUS-INSTRUMENT	9
2.2.1	Anwendungen, z.B. EL-FLOW	9
2.2.2	Einrichtungen.....	10
2.2.3	Baudraten Multibus RS232	10
3	BESCHREIBUNG DES PROPAR-PROTOKOLLS.....	11
3.1	ALLGEMEINES.....	11
3.2	INITIALISIERUNG LOKALER HOST-SCHNITTSTELLEN AN MULTIBUS-INSTRUMENTEN.....	11
3.3	PROPAR NACHRICHTENFORMAT	12
3.3.1	Basic ProPar Datalink-Format.....	12
3.3.2	ASCII-Tabelle	12
3.3.3	ProPar-ASCII-Protokoll	13
3.4	PROPAR KOMMUNIKATIONSNAHRICHTEN	15
3.4.1	Kommunikationsbefehle	15
3.4.2	Parametertypen	16
3.5	VERKETTUNG	16
3.6	STATUSMELDUNG.....	17
3.7	SENDEN VON PARAMETERN	18
3.8	ABFRAGEN VON PARAMETERN	18
3.9	BEISPIELE – PROPAR ASCII PROTOKOLL.....	20
3.9.1	Senden des Sollwerts.....	20
3.9.2	Abfragen des Sollwerts	21
3.9.3	Abfragen des Messwerts.....	22
3.9.4	Abfragen des Zählerwerts.....	23
3.9.5	Auflistung von ProPar ASCII Beispielen	24
3.9.6	Abfragen verketteter Parameter.....	33
3.9.7	Beispiel abfragen verketteter Parameter, setpoint und measure	35
3.9.8	Beispiel abfragen verketteter Parameter, measure und temperature.....	35
3.9.9	Beispiel abfragen verketteter Parameter, fmeasure und temperature	35
3.10	BEISPIELE - PROPAR VERBESSERTES BINÄRPROTOKOLL	36
3.10.1	Übertragung Sollwert - ProPar verbessertes Binärprotokoll.....	36
3.10.2	Abfrage Sollwert - ProPar verbessertes Binärprotokoll.....	37
3.10.3	Abfrage Messwert - ProPar verbessertes Binärprotokoll.....	38
3.10.4	Auflistung von ProPar verbesserten Binärbeispielen	39
3.10.5	Beispiele verketteter Parameter - ProPar verbessertes Binärprotokoll.....	40
4	BETRIEB MIT ZWEI SCHNITTSTELLEN	41
5	PARAMETERINFORMATIONEN	42
6	FEHLERSUCHE.....	44
6.1	LED-ANZEIGEN.....	44
6.2	TIPPS UND HINWEISE ZUR FEHLERSUCHE.....	44
7	SERVICE	45
8	ANHANG	46

8.1 TABELLE DER PARAMETERWERTE46
8.2 TABELLE DER PARAMETEREIGENSCHAFTEN.....46

1 ALLGEMEINE PRODUKTINFORMATIONEN

1.1 EINFÜHRUNG

In diesem Handbuch wird erläutert, wie die Kommunikation zwischen Instrumenten von Bronkhorst® und einem PC/einer SPS mit Hilfe der seriellen RS232-Verbindung abläuft. Um mit diesen Instrumenten arbeiten zu können, müssen Sie ihre Software anhand der Informationen in diesem Dokument selbst schreiben. Bronkhorst bietet auch Software für einen einfachen Betrieb digitaler Instrumente mit ihrem PC an, wenn Sie mit Microsoft Windows arbeiten. Auf dem höchsten unterstützten Kommunikationsniveau können damit FlowDDE-Kanäle für Windows-Anwendungen verwendet werden. Das Programm FlowDDE kann für die bequeme Verbindung zwischen Windows-Anwendungen (z.B. Excel, Visual Basic, LabVIEW, Delphi, Borland C) und digitalen Instrumenten genutzt werden. Für LabVIEW-, Visual Basic- und Excel-Umgebungen gibt es verschiedene Beispiele.



Auf niedrigerem Kommunikationsniveau können Sie auch FLOWB32.DLL zum Anzeigen/Ändern von Parameterwerten einsetzen.

Zum direkten Empfangen und Senden von Parameterwerten von oder an ProPar-Geräte über die verfügbaren Schnittstellen gibt es ein spezielles Protokoll für den Datenaustausch zwischen diesen Geräten. Dieses Protokoll heißt und wurde speziell für Ausrüstung von Bronkhorst® entwickelt, sodass Ausrüstung von Dritten nicht angeschlossen werden kann. Es besteht aus einer hierarchischen Anordnung der Instrumente/Knoten (Nodes) (max. 126) und enthält Prozesse (max. 127) mit Parametern (FBNr) (max. 32), die auf bestimmte Werte eingestellt werden können, um die Einstellungen/Eigenschaften der Instrumente festzulegen.

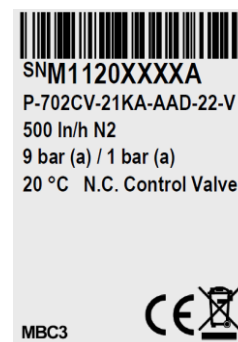
Beim Betrieb eines ProPar-Systems mit einem HOST-Computer muss dieses Datenaustauschprotokoll bekannt sein, wenn die Schnittstellen direkt angesprochen werden sollen.

Bei direkter Kommunikation über RS232 mit einem Multibus-Instrument oder bei Verwendung eines RS232 auf FLOW-BUS-Schnittstellentyps (Baudraten bis 38400 mit Schalter und 2 LEDs) ist eine spezielle Initialisierung nicht erforderlich.

1.2 MULTIBUSTYPEN

Im Jahr 2000 entwickelte Bronkhorst seine ersten digitalen Instrumente nach dem „Multibus“-Prinzip. Die Grundplatine der Instrumente enthielt alle allgemeinen Funktionen, die zum Messen und Regeln des Masseflusses notwendig waren, darunter Alarm-, Summier- und Diagnosefunktionen. **Analoge** E/A-Signale sowie eine **RS232**-Schnittstelle waren hierbei Standard. Ergänzend dazu können Zusatzschnittstellen mit **DeviceNet™**, **PROFIBUS DP**, **Modbus**, **FLOW-BUS** oder **EtherCAT**-Protokolle integriert werden. Die erste Generation (**MBC-I**) basierte auf einem 16-Bit-Controller von Fujitsu. Sie wurde 2003 durch den Multibus Typ 2 (**MBC-II**) abgelöst. Auch diese Version basierte auf dem 16-Bit-Controller von Fujitsu, zeichnete sich jedoch durch einige Verbesserungen gegenüber dem MBC-I aus, darunter die Stromsteuerung des Ventils. Dadurch wurden die Wärmezeugung reduziert und die Regeleigenschaften verbessert. Die neueste Version des Multibus-Controllers Typ 3 (**MBC3**) wird 2011 eingeführt. Sie baut auf einem 72 MHz 32 Bit NXP ARM Controller auf und verfügt über AD- und DA-On-Board-Controller, wodurch eine störfreie Messung und Regelung des Ventils ohne Verzögerungen ermöglicht wird. Der interne Regelkreis ist 6 Mal schneller verglichen mit dem MBC-II, weshalb sich die Regelstabilität deutlich verbessert hat. Außerdem wurden Funktionen wie der Verpolungsschutz, die Einschaltstrombegrenzung und der Überspannungsschutz verbessert.

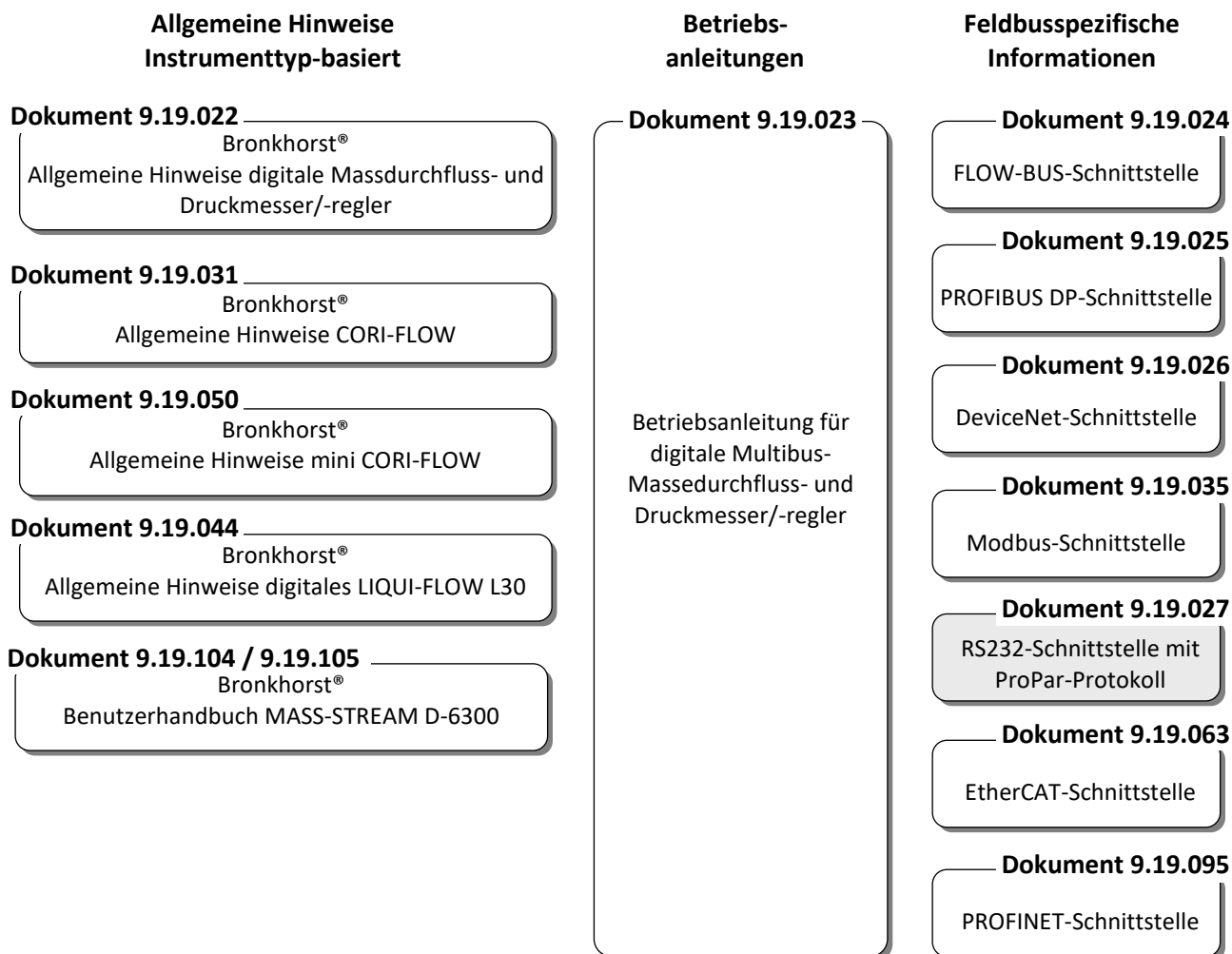
MBC3-Instrumente sind an dem links unten auf dem Typenschild platzierten „MBC3“ zu erkennen (siehe Beispiel).



1.3 VERWEISE AUF ANDERE ANWENDBARE DOKUMENTE

Die Handbücher und Anleitungen für digitale Instrumente sind modular aufgebaut. Allgemeine Hinweise enthalten Informationen über die Funktionsweise und Installation der Instrumente. Betriebsanleitungen erläutern die Nutzung der Merkmale und Parameter der digitalen Instrumente. Feldbuspezifische Informationen dienen zur Erklärung der Installation und Verwendung des im Instrument installierten Feldbusses.

1.3.1 Handbücher und Benutzeranleitungen



1.3.2 Technische Zeichnungen

Anschlussplan laboratory-style FLOW-BUS	(Dokument Nr. 9.18.063)
Anschlussplan IN-FLOW FLOW-BUS	(Dokument Nr. 9.18.052)
Anschlussplan CORI-FLOW FLOW-BUS	(Dokument Nr. 9.18.048)
Anschlussplan LIQUI-FLOW L30 digital FLOW-BUS	(Dokument Nr. 9.18.074)

1.3.3 Softwaretools

FlowPlot
FlowView
Flowfix
FlowDDE

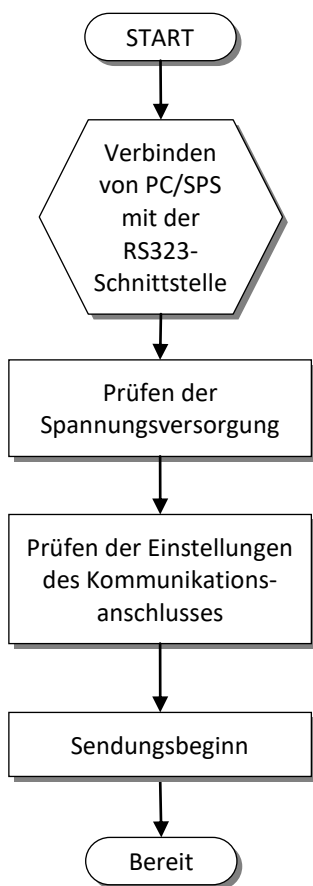


Alle diese Dokumente finden Sie unter:
<http://www.bronkhorst.com/en/downloads>

1.4 START IN KURZFORM

Alle notwendigen Einstellungen für dieses Gerät wurden bereits von Bronkhorst vorgenommen.

Der schnellste Weg, dieses Modul in ihrem eigenen System betriebsfähig zu machen, ist die sorgfältige Ausführung der folgenden Schritte.



Stellen Sie sicher, dass ihr PC oder ihre SPS über das richtige Kabel mit der RS232-Schnittstelle verbunden ist.

- Multibus-Instrumente erfordern ein Spezialkabel mit T-Stecker
- RS232 auf ProPar-Schnittstellen benötigen ein durchgehendes 9-poliges Kabel ohne Abzweigungen mit männlichem und weiblichem Steckverbinder

Die Kabellänge für RS232 darf 10 Meter nicht überschreiten.

Vergewissern Sie sich, dass Instrument oder Schnittstelle mit Spannung versorgt werden (+15...24 VDC).

Verwenden Sie die Einstellung [38400,n,8,1] für ihren COM-Port:
Baudrate = 38400 Baud, keine Parität, 8 Datenbits, 1 Stoppbit.

Beginnen Sie mit dem Senden von Daten wie in den folgenden Abschnitten beschrieben.



Bei Schwierigkeiten könnten Programme wie Hyperterminal (verfügbar in MS-Windows) oder FlowDDE (von Bronkhorst®) sehr hilfreich sein.

2 SCHNITTSTELLEN

2.1 RS232/FLOW-BUS-SCHNITTSTELLE

Die RS232 auf ProPar-Schnittstelle ist eine Schnittstelle zwischen dem FLOW-BUS und dem seriellen V24- (Computer-)Anschluss RS232. Sie wird entweder in einem eigenen Gehäuse mit einem FLOW-BUS-Steckverbinder und einem RS232-Steckverbinder oder als integrierter 14TE-Einschub ihres Anzeige- und Regelsystems E-8000 geliefert. Der Konverter sorgt für eine Kommunikation mit einer Baudrate bis zu 38400 Baud. Unterstützung durch Kommunikationssoftware ist möglich. Die Kommunikationseinstellungen sind: 38400,n,8,1.



2.1.1 D-Steckverbinder für RS232

Der weibliche RS232-D-Steckverbinder (Subminiatur, 9-polig) hat folgende Pinbelegung:

Pinnummer	Beschreibung
1	nicht belegt
2	TXD
3	RXD
4	nicht belegt
5	0 Vdc
6	DTR
7	CTS
8	RTS
9	Schirm

2.2 RS232 AM MULTIBUS-INSTRUMENT

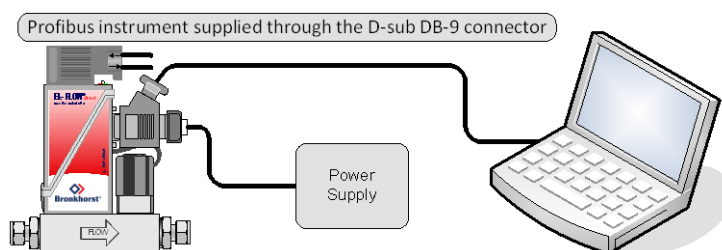
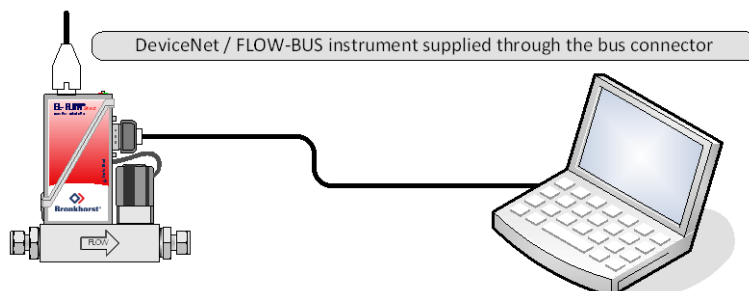
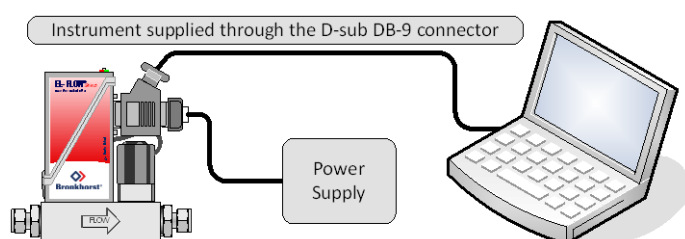
Die RS232-Schnittstelle an einem Multibus-Instrument kann mit jedem seriellen V24-(Computer-)Anschluss RS232 verbunden werden. Die Berücksichtigung des richtigen Anschlussplanes muss sichergestellt sein. Bronkhorst® bietet für die Datenübertragung spezielle Kabel an, die die RS232-Verbindung von der Anschlussversorgung und dem analogen Ein- und Ausgangssignal trennen. An dem 9-poligen männlichen D-Sub-Steckverbinder des Instruments liegen RX und TX an Pin 6 und Pin 1.

RS232-Kommunikation ist möglich durch:

- 9-poliger D-Sub-Stecker (keine IP65-Ausführung, z.B. EL-FLOW)
- 8 DIN Stecker (IP65-Ausführung, z.B. CORI-FLOW)

Die richtigen Anschlüsse sind den Anschlussplänen zu entnehmen.

2.2.1 Anwendungen, z.B. EL-FLOW



Als Voreinstellung arbeitet die Datenübertragung der Schnittstelle bei einer Baud-Rate von 38400 baud. Bei Geräten, die die Änderung der RS232 Baud-Rate gestatten, kann diese Einstellung auch anders konfiguriert werden.

2.2.2 Einrichtungen

Handshaking wird nicht angewendet. PC/SPS-seitig ist ein Null-Modem-Anschluss erforderlich.

Die Kommunikationseinstellungen sind: 38400,n,8,1.

Bronkhorst® bietet ein Spezialkabel an, das für die Kommunikation benötigt wird.

Unterstützung durch Kommunikationssoftware ist möglich.

2.2.3 Baudraten Multibus RS232

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

3 BESCHREIBUNG DES PROPAR-PROTOKOLLS

3.1 ALLGEMEINES

Auf dem höchsten unterstützten Kommunikationsniveau können mit diesem Protokoll DDE-Kanäle für Windows-Anwendungen verwendet werden. Auf niedrigerem Kommunikationsniveau kann FLOWB32.DLL zum Ändern von Parameterwerten eingesetzt werden.

Zum direkten Empfangen und Senden von Parameterwerten von oder an ProPar-Geräte über die verfügbaren Schnittstellen gibt es ein spezielles Protokoll für den Datenaustausch zwischen diesen Geräten. Beim Betrieb eines ProPar-Systems mit einem HOST-Computer muss dieses Datenaustauschprotokoll bekannt sein, wenn die Schnittstellen direkt angesprochen werden sollen.

Es gibt zwei unterschiedliche Kommunikationsprotokolle für den PC und den RS232-HOST:

- ein ASCII-Protokoll für Kommunikation, das mit bestehenden ProPar-Anwendungen kompatibel ist. Dieses Protokoll bedient nur einen Master/Slave-Dialog auf einmal.
- ein erweitertes Binärprotokoll, das den gleichzeitigen Datentransfer an verschiedene Nodes unterstützt. Dieses Protokoll enthält eine Datenfolgenummer und bedient mehrere Master/Slave-Dialoge gleichzeitig.

Ein ProPar-Gerät erkennt automatisch, welches Protokoll der PC verwendet, und passt sein Verhalten entsprechend an. Die Art des Protokolls erkennt man an dem ersten Zeichen einer Nachricht.

- Das erste Zeichen ist „:“ (0x3A) nur diese Nachricht.
- Das erste Zeichen ist DLE (0x10) erweiterte Nachricht.

Über ProPar DLL (FLOWB32.DLL) erkennt der PC das verwendete Protokoll.

Bei der Kommunikation ist immer der PC der Master und der HOST der Slave. Der HOST reagiert immer auf Anfragen des PCs.

3.2 INITIALISIERUNG LOKALER HOST-SCHNITTSTELLEN AN MULTIBUS-INSTRUMENTEN

Bei Verwendung eines digitalen Instruments mit RS232-Schnittstelle ist die Baudrate auf 38400 Baud fest eingestellt und eine spezielle Initialisierung ist nicht notwendig. Über die serielle Verbindung zu einem COM-Port ihres PCs oder einer SPS müssen Sie mit dem Instrument unter Verwendung des ProPar-Protokolls kommunizieren. Jedes Instrument hat seine eigene Node-Adresse (3...120). Diese Adresse muss bekannt sein, wenn man eine Nachricht an das Instrument senden will. Wenn man allerdings eine Nachricht an Adresse 128 schickt, wird das Instrument immer darauf reagieren. Bei einer Punkt-zu-Punkt-Verbindung wie RS232 ist dies der einfachste Weg, zu einer Kommunikation zu kommen, die unter allen Umständen funktioniert (unabhängig von der tatsächlichen Node-Adresse des Instruments).

3.3 PROPAR NACHRICHTENFORMAT

3.3.1 Basic ProPar Datalink-Format

Das Nachrichtenformat „basic data link“ hat folgende Felder:

Node	Nachrichtempfänger	Länge	Datenfeldlänge	Daten	Daten	Daten	usw.
------	--------------------	-------	----------------	-------	-------	-------	------

In der ProPar-Umgebung kann das Datenfeld bis zu 256 Byte Daten enthalten. In der hier beschriebenen HOST-Anwendung entsprechen die Datensätze den PROPAR-Kodierungsregeln und das Datenfeld enthält maximal 64 Byte.

3.3.2 ASCII-Tabelle

Der **American Standard Code for Information Interchange (ASCII)** ist ein Zeichenkodierungssystem, das auf der Anordnung des englischen Alphabets basiert. ASCII-Codes stellen Text in Computern, Kommunikationsausrüstung und anderen Geräten dar, die mit Text arbeiten. Die meisten modernen Zeichenkodierungssysteme basieren auf ASCII.

Dez.	Okt.	Hex.	Binär	Code	Dez.	Okt.	Hex.	Binär	Code	Dez.	Okt.	Hex.	Binär	Code	Dez.	Okt.	Hex.	Binär	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-Protokoll

Der vorhandene RS232-HOST verwendet ein ASCII-Protokoll, das zur Verfügung steht, um die Kompatibilität mit der vorhandenen Treibersoftware sicherzustellen.

Eine „basic data link“-Nachricht ist in ASCII wie folgt kodiert:

	Länge		Node		Daten		
:	len1	len2	node1	node2	data1	data2	CR

Byte	Erläuterung
:	Startzeichen (Semikolon)
Len1, len2	Länge der Nachricht einschließlich der Node-Adresse in Bytes, d.h. (len1, len2) ist die Basisnachrichtenlänge +1.
node1, node2	Node-Adresse des Empfängers (PC an HOST) Node-Adresse des Senders (HOST an PC)
data1, data2	Nachrichtenfeld
CR	Endzeichen

Alle Bytes (mit Ausnahme der Start- und Endzeichen) werden in der ASCII-Darstellung von 1 binären Byte in 2 hexadezimale Bytes konvertiert.

Beispiel: Binäres Datenbyte 0x2A --> hexadezimale ASCII-Zeichen 0x32, 0x41.

Zur Übertragung von Fehlermeldungen vom HOST zum PC wird ein spezieller Meldungstyp mit folgender Struktur verwendet:

	0x01		Fehler		
:	0x30	0x31	error1	error2	CR

Byte	Erläuterung
:	Startzeichen
0x30, 0x31	Länge der Meldung (1 Byte)
error	Fehlercode, zweistellige HEX-Zahl
CR	Endzeichen

Der Fehlercode kann folgende Werte haben:

Wert	Bedeutung
3	propar Protokollfehler
4	propar Protokollfehler (oder CRC-Fehler)
5	Zielknotenadresse abgelehnt
9	Antwortnachricht Timeout
1,2,8	allgemeiner Fehler

3.3.3.1 ProPar erweitertes Binärprotokoll

Das erweiterte Protokoll ist binär kodiert. Zur Erkennung des Anfangs und Endes einer Nachricht innerhalb eines Bytestroms werden Prüfsequenzen genutzt. Eine Prüfsequenz beginnt mit einem DLE-Byte (0x10), gefolgt von einem Kontrollbyte. Die Prüfsequenzen sind wie folgt definiert:

Erstes Byte	Zweites Byte	Funktion
DLE (0x10)	STX (0x02)	Start der Nachricht
DLE (0x10)	ETX (0x03)	Ende der Nachricht
DLE (0x10)	DLE (0x10)	Datenbyte 0x10
DLE (0x10)	jedes andere Zeichen	Nicht zulässig. Nachrichten, die eine solche Sequenz enthalten, werden ignoriert. Der Empfänger wartet auf eine neue DLE STX-Sequenz.

Die Sequenz [DLE DLE] wird benutzt um zu verhindern, dass möglicherweise in dem übertragenen binären Datenstrom auftauchende DLE-Bytes als Beginn einer Prüfsequenz interpretiert werden. Der Sender ersetzt jedes DLE-Byte in den Daten durch 2 DLE-Bytes. Der Datenanschluss des Empfängers wandelt eine Sequenz [DLE DLE] in ein DLE-Byte um.



Wenn ein RS232-Fehler auftritt (Empfängerüberlauf, Framing-Fehler, unzulässige Prüfsequenz), wird der Datenverbindungsframe ignoriert.

Nachrichten zwischen PC und HOST mit der erweiterten binären Kodierung haben folgende Struktur:

Byte	Erläuterung
DLE, STX	Startsequenz
seq	Nachrichtensequenznummer
node	Node-Adresse des Empfängers (PC an HOST) Node-Adresse des Senders (HOST an PC)
len	Länge des Datenfeldes in Bytes
data	Nachrichtenfeld
DLE, ETX	Endsequenz

Das erweiterte Protokoll gestattet die gleichzeitige Übertragung von mehr als einer Anfrage. Die Sequenznummerierung ermöglicht die Zuordnung der Antwort zu der betreffenden Anfrage. Der HOST hat mehr als einen Puffer zur Zwischenspeicherung von (normalerweise 5) Nachrichten. Wenn die Nachrichtenpuffer voll sind, antwortet der HOST mit einer Fehlermeldung.

Die Antworten vom HOST an den PC haben das gleiche Nachrichtenformat wie die Anfragen. Eine Fehlermeldung hat ein spezielles Format:

Byte	Erläuterung
DLE, STX	Startsequenz
seq	Nachrichtensequenznummer, wie in der Anfrage
node	Node-Adresse des Senders, wie in der Anfrage
error	Fehlercode
DLE, ETX	Endsequenz

Der Fehlercode kann folgende Werte haben:

Wert	Bedeutung
3	propar Protokollfehler
4	propar Protokollfehler (oder CRC-Fehler)
5	Zielknotenadresse abgelehnt
9	Antwortnachricht Timeout
1,2,8	allgemeiner Fehler

3.4 PROPAR KOMMUNIKATIONSNACHRICHTEN

Kommunikationsnachrichten zwischen ProPar-Schnittstellen und anderen Geräten bestehen aus Befehlsstrings mit spezifischen Informationen. Dieser Befehlsstring ist entweder in ASCII (RS232) oder BINÄR. Grundsätzlich enthält der String verschiedene Informationsbytes. Durch RS232 werden diese hexadezimalen Bytes in ASCII umgewandelt (z.B. entspricht der Bytewert 0x0A dem Wert „0A“ in ASCII und es sollten Großbuchstaben verwendet werden). Nachrichten über RS232 geht das Zeichen „:“ voran, und beendet werden sie mit „\r\n“ (Wagenrücklauf-Zeilenvorschub).

In den ProPar -Nachrichten sind verschiedene BEFEHLE verfügbar. Zum Lesen und Schreiben aller Standardparameter sind nur die Befehle RD (04) und WR (01) erforderlich. Ein RD-Befehl wird durch einen WR-Befehl, der den erfragten Wert enthält, oder eine Statusmeldung, die eine Fehlernummer enthält, beantwortet. Ein WR-Befehl wird durch eine Statusmeldung beantwortet, die eine Fehlernummer enthält (bei Fehlernummer = 0 war der WR-Befehl OK).



Das ASCII-Zeichen „:“ hat den hexadezimalen Wert 3A.
 Das ASCII-Zeichen „\r“ hat den hexadezimalen Wert 0D.
 Das ASCII-Zeichen „\n“ hat den hexadezimalen Wert 0A.

3.4.1 Kommunikationsbefehle

Kommunikationsbefehle	
Befehl	Beschreibung
00	Statusmeldung
01	Parameter mit Empfängeradresse senden, wird mit Befehl des Typs 00 beantwortet
02	Parameter mit Empfängeradresse senden, kein Status abgefragt
03	Parameter mit Senderadresse senden, kein Status abgefragt
04	Parameter abfragen, wird mit Befehl des Typs 02 oder 00 beantwortet
06	Prozess stoppen
07	Prozess starten
08	Prozess anfordern
09	Prozess nicht mehr anfordern

Für den Zugriff auf einen spezifischen Parameter müssen folgende Punkte bekannt sein:

- Node-Adresse jedes ProPar -Gerät ist mit einer spezifischen Node-Adresse im System verbunden.
- Prozessnummer jedes Gerät (Node) besteht aus mehreren Prozessen.
- Parameternummer (FBnr) jeder Prozess besteht aus mehreren Parametern.
- Parametertyp jeder Parameter kann einen unterschiedlichen Typ und Wert haben.



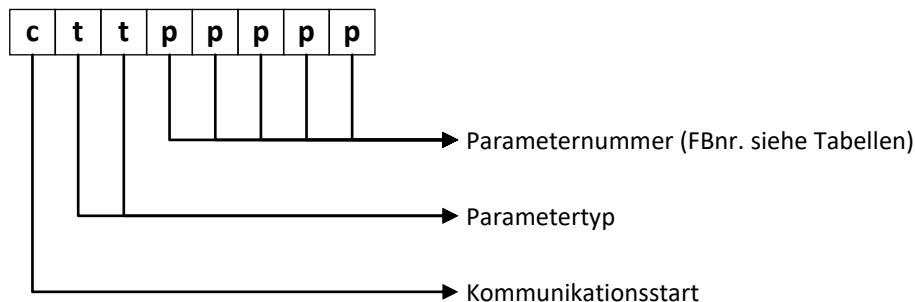
Die Parameternummern und -werte finden Sie in den Tabellen der „Parametereigenschaften“ und „Parameterwerte“ in diesem Handbuch.

3.4.2 Parametertypen

Parametertypen				
Typ	Id	Bytes	Bits	Bereich
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 Byte 0... 4294967296
String	60h	X	01100000	Längen müssen angegeben werden

Byteanzahl je Parametertyp					
Wert 1	Wert 2	Wert 3	Wert 4	Wert 5	Wert 6
Byte					
Integer					
Float					
Long					
String length	String				

Die Nachrichten nutzen Bytes, die aus Daten von mehreren Bytes zusammengesetzt sind. Nachstehend finden Sie ein Beispiel für ein solches zusammengesetztes Byte.



3.5 VERKETTUNG

Verkettung kann zum Senden oder Abfragen von mehr als einem Parameter pro Nachricht verwendet werden. Wenn alle Parameter zum gleichen Prozess gehören, können sie auf Parameterebene verkettet werden. Gehören die Parameter zu verschiedenen Prozessen, können sie auf Prozessebene verkettet werden. Auch eine Kombination ist möglich. Zur Verkettung auf Parameterebene wird das erste Bit der Parameternummer eingegeben, wenn ein anderer Parameter folgt, der zum gleichen Prozess gehört. Zur Verkettung auf Prozessebene wird das erste Bit der Prozessnummer angegeben, wenn sich ein anderer Prozess anschließt.

3.6 STATUSMELDUNG

Wird ein Gerät mit dem Befehl „01“ angeschrieben, führt dies zu einer Statusmeldung. Nachstehend sind die möglichen Erwidernungen aufgelistet.

Nr.	Byte	Beschreibung
0	:	Startzeichen
1	04	feste Nachrichtenlänge 4.
2	Node	Node-Adresse
3	00	Befehl Status
4	Status	00 kein Fehler 01 Prozess angefordert 02 Befehlsfehler 03 Prozessfehler 04 Parameterfehler 05 Parametertypfehler 06 Parameterwertfehler 07 Netzwerk nicht aktiv 08 Time-out Startzeichen 09 Time-out serielle Leitung 0A Hardwarespeicherfehler 0B Nodenummernfehler 0C allgemeiner Kommunikationsfehler 0D Parameter ohne Schreibzugriff 0E Fehler PC-Kommunikation 0F keine RS232-Verbindung 10 PC-Speicher voll 11 Parameter ohne Lesezugriff 12 Systemkonfiguration unbekannt 13 keine freie Node-Adresse 14 falscher Schnittstellentyp 15 Fehler Verbindung serieller Anschluss 16 Fehler Kommunikationsbeginn 17 Kommunikationsfehler 18 Fehler Schnittstelle Busmaster 19 Time-out Antwort 1A kein Startzeichen 1B Fehler erste Stelle 1C Pufferüberlauf im Host 1D Pufferüberlauf 1E keine Antwort gefunden 1F Fehler Kommunikationsende 20 Synchronisationsfehler 21 Sendefehler 22 Protokollfehler 23 Pufferüberlauf im Modul
5	Index oder angeforderter Prozess	Index zeigt auf das erste Byte in der Sendenachricht, für das der obige Status gilt. Im Falle des Status ANFORDERUNGSFEHLER enthält dieses Feld den angeforderten Prozess.
6	\r	Wagenrücklauf
7	\n	Zeilenvorschub



Der Wert ab Byte 5 der Statusmeldung kann vernachlässigt werden, wenn der Wert von Byte 4 = 0 ist.

3.7 SENDEN VON PARAMETERN

Zum Anschreiben eines Geräts müssen die Befehlscodes „01“ und „02“ verwendet werden. Der Befehl „01“ führt zu einer Statusmeldung zur Fehlerprüfung. Auf Befehl „02“ wird keine Antwort vom Gerät ausgegeben. In der nachstehenden Tabelle wird die gesamte Befehlszeile erläutert.

Senden eines Parameters			
Nr.	Byte	Layout	Beschreibung
0	:		Startzeichen
1	Länge		Nachrichtenlänge
2	Node		Node-Adresse
3	01 oder 02		Befehl Schreiben, für Typ 01 wird eine Statusmeldung (00) zurückgesendet
4	Prozess	c p p p p p p p	c Prozess verkettet
			p Prozessnummer
5	Parameter	c t t p p p p p	c Parameter verkettet
			t Parametertyp
			p Parameternummer (FBnr.)
6	Wert 1		Wert für alle Typen. Bei „string“ enthält dieses Feld die Stringlänge.
7	Wert 2		Wert für Typ „integer“, „float“ oder „long“.
8	Wert 3		Wert für Typ „float“ oder „long“.
9	Wert 4		Wert für Typ „float“ oder „long“.
X	Wert x		Weitere Wertfelder folgen für Typ „string“ abhängig von der Stringlänge. Wenn die gegebene Stringlänge null ist, sollte das letzte Feld auch eine Null enthalten.
X+1	\r		Wagenrücklauf
X+2	\n		Zeilenvorschub

3.8 ABFRAGEN VON PARAMETERN

Für jeden abgefragten Parameter kann eine Indexnummer vergeben werden. Der antwortende Node schickt diese Indexnummer zusammen mit dem abgefragten Parameter zurück. Dies gibt einem die Möglichkeit zu prüfen, welcher Parameter zurückgeschickt wird, wenn mehrere Parameter abgefragt werden.

Abfragen eines Parameters			
Nr.	Byte	Layout	Beschreibung
0	:		Startzeichen
1	Länge		Nachrichtenlänge
2	Node		Node-Adresse
3	04		Befehl Lesen
4*	Prozess	c p p p p p p p	c Prozess verkettet
			p Prozessnummer
5*	Parameter	c t t n n n n n	c Parameter verkettet
			t Parametertyp
			n Parameterindex 0...31
6	Prozess	- p p p p p p p p	- nicht verwendet (Null benutzen)
			p Prozessnummer
7	Parameter	- t t p p p p p p	- nicht verwendet (Null benutzen)
			t Parametertyp
			t Parameternummer (FBnr.)
8	Stringlänge		Bei Parametertyp „string“ enthält dieses Feld die erwartete Stringlänge.
9	\r		Wagenrücklauf
10	\n		Zeilenvorschub



Empfehlung:

Verwenden Sie den gleichen Prozessnamen für Nr. 4 und 6. Falls kein Index erforderlich ist, geben Sie die FBnr. ein, so dass die Rücknachricht mit der abgefragten Parameternummer zurückgeschickt wird.

Antwort auf die Abfrage			
Nr.	Byte	Layout	Beschreibung
0	:		Startzeichen
1	Länge		Nachrichtenlänge
2	Node		Node-Adresse
3	02		Befehl Schreiben
4*	Prozess	c p p p p p p p	c Prozess verkettet
			p Prozessnummer
5*	Parameter	c t t n n n n n	c Parameter verkettet
			t Parametertyp
			n Parameterindex 0...31
6	Wert 1		Wert für alle Typen. Bei „string“ enthält dieses Feld die Stringlänge.
7	Wert 2		Wert für Typ „integer“, „float“ oder „long“.
8	Wert 3		Wert für Typ „float“ oder „long“.
9	Wert 4		Wert für Typ „float“ oder „long“.
X	Wert x		Weitere Wertfelder folgen für Typ „string“ abhängig von der Stringlänge. Wenn die gegebene Stringlänge null ist, sollte das letzte Feld auch eine Null enthalten.
X+1	\r		Wagenrücklauf
X+2	\n		Zeilenvorschub



* Das abgefragte Gerät kopiert diese Werte von der Abfragenachricht direkt in die Antwortnachricht.

3.9 BEISPIELE – PROPAR ASCII PROTOKOLL

3.9.1 Senden des Sollwerts

Senden des Sollwerts = 50 % an Node 3 Prozess 1. Sollwerte werden in einem Bereich von 0 bis 32000 gesendet, d.h. für dieses Beispiel muss 16000 gesendet werden.

Senden des Sollwerts = 50 % an Node 3 Prozess 1 Parameter 1						
Nr.	Byte	Layout	Beschreibung			Beschreibung pro Block
0	:		Startzeichen			Start
1	06		Länge 6			
2	03		Node 3			
3	01		Befehl Schreiben mit Statusrückmeldung			
4	01	c p p p p p p p 00000001	c 00	Prozess nicht verkettet		Prozess 1
			p 01	Prozess 1		
5	21	c t t p p p p p 00100001	c 00	Parameter nicht verkettet		Parameternummer 1 Sollwert
			t 20	Parametertyp „integer“		
			p 01	Parameternummer (FBnr.) 1		
6	3E		Sollwert 16000 = 3E80h			Ende
7	80					
8	\r		Wagenrücklauf			Ende
9	\n		Zeilenvorschub			
Gesamtstring		:06030101213E80\r\n				

Antwort von Node 3						
Nr.	Byte	Beschreibung			Beschreibung pro Block	
0	:	Startzeichen			Start	
1	04	feste Nachrichtenlänge 4				
2	03	Node-Adresse 03				
3	00	Befehl Status				
4	00	Status in Ordnung			Statusantwort	
5	05	Status in Ordnung, Wert zeigt auf Ende der Sendenachricht				
6	\r	Wagenrücklauf			Ende	
7	\n	Zeilenvorschub				
Gesamtstring		:0403000005\r\n				

Die Position von Nr. 5 sieht wie folgt aus.

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

3.9.2 Abfragen des Sollwerts

Abfragen des Sollwerts von Node 3 Prozess 1, Typ „integer“						
Nr.	Byte	Layout	Beschreibung			Beschreibung pro Block
0	:					Start
1	06		Länge 6			
2	03		Node 3			
3	04		Befehl Lesen			Prozess 1
4	01	00000001	C	00	Prozess nicht verkettet (Rücksendung)	
			P	01	Prozess 1 (Rücksendung)	
5	21	00100001	C	00	Parameter nicht verkettet (Rücksendung)	Parameterindex 1 Prozess 1 Parameternummer 1 Sollwert
			T	20	Parametertyp „integer“ (Rücksendung)	
			N	01	Parameterindex 1 (Rücksendung)	
6	01	00000001	P	01	Prozess 1	
7	21	00100001	T	20	Parametertyp „integer“	
			P	01	Parameternummer (FBnr.) 1 (Sollwert)	
8	\r		Wagenrücklauf			Ende
9	\n		Zeilenvorschub			
Gesamtstring			:06030401210121\r\n			

Antwort von Node 3 Prozess 1, Typ „integer“						
Nr.	Byte	Layout	Beschreibung			Beschreibung pro Block
0	:					Start
1	06		Länge 6			
2	03		Node 3			
3	02		Befehl Schreiben			Prozess
4	01	00000001	C	00	Prozess nicht verkettet	
			P	01	Prozess 1 (Empfangsprozess)	
5	21	00100001	C	00	Parameter nicht verkettet	Parameterindex 1 Sollwert
			T	20	Parametertyp „integer“	
			N	01	Parameterindex 1	
6	3E		Wert 3E80h = 16000 = 50 %			
7	80					
8	\r		Wagenrücklauf			
9	\n		Zeilenvorschub			
Gesamtstring			:06030201213E80\r\n			



Das ASCII-Zeichen „LEERZEICHEN“ (20 HEX) wird für jedes Leerzeichen im String ausgegeben.

3.9.3 Abfragen des Messwerts

Abfragen des Messwerts von Node 3 Prozess 1, Typ „integer“					
Nr.	Byte	Layout	Beschreibung		Beschreibung pro Block
0	:				Start
1	06		Länge 6		
2	03		Node 3		
3	04		Befehl Lesen		Prozess 1
4	01	00000001	C 00	Prozess nicht verkettet (Rücksendung)	
			P 01	Prozess 1 (Rücksendung)	Parametertyp 1 Prozess 1 Parameterindex 0 Measure
5	21	00100001	C 00	Parameter nicht verkettet (Rücksendung)	
			T 20	Parametertyp „integer“ (Rücksendung)	
			N 01	Parameterindex 1 (Rücksendung)	
6	01	-0000001	P 01	Prozess 1	Ende
7	20	-0100000	T 20	Parametertyp „integer“	
			P 00	Parameterindex (FBnr.) 0 (measure)	Ende
8	\r			Wagenrücklauf	
9	\n			Zeilenvorschub	
Gesamtstring		:06030401210120\r\n			

Antwort von Node 3 Prozess 1					
Nr.	Byte	Layout	Beschreibung		Beschreibung pro Block
0	:				Start
1	06		Länge 6		
2	03		Node 3		
3	02		Befehl Schreiben		Prozess 1
4	01	00000001	C 00	Prozess nicht verkettet	
			P 01	Prozess 1 (Empfangsprozess)	Parameterindex 1 measure
5	21	00100001	C 00	Parameter nicht verkettet	
			T 20	Parametertyp „integer“	
			N 01	Parameterindex 1	
6	3E		Wert 3E80h = 16000 = 50 %		Ende
7	80				
8	\r		Wagenrücklauf		
9	\n		Zeilenvorschub		
Gesamtstring		:06030201213E80\r\n			

3.9.4 Abfragen des Zählerwerts

Abfragen des Zählerwerts von Node 3 Prozess 104, Typ „float“						
Nr.	Byte	Layout	Beschreibung			Beschreibung pro Block
0	:					Start
1	06		Länge 6			
2	03		Node 3			
3	04		Befehl Lesen			Prozess 104
4	68	01101000	C 00	Prozess nicht verkettet (Rücksendung)		
			P 68	Prozess 104 (Rücksendung)		
5	41	01000001	C 00	Parameter nicht verkettet (Rücksendung)		Parameterindex 1 Prozess 104 Parameterindex 1 Counter value
			T 40	Parametertyp „float“ (Rücksendung)		
			N 01	Parameterindex 1 (Rücksendung)		
6	68	01101000	P 68	Prozess 104		
7	41	01000001	T 40	Parametertyp „float“		
			P 01	Parameterindex 1 (Rücksendung)		
8	\r			Wagenrücklauf		Ende
9	\n			Zeilenvorschub		
Gesamtstring			:06030468416841\r\n			

Antwort von Node 3 Prozess 104						
Nr.	Byte	Layout	Beschreibung			Beschreibung pro Block
0	:					Start
1	08		Länge 8			
2	03		Node 3			
3	02		Befehl Schreiben			Prozess 104
4	68	01101000	C 00	Prozess nicht verkettet		
			P 68	Prozess 104 (Empfangsprozess)		
5	41	01000001	C 00	Parameter nicht verkettet		Parameterindex 1 Counter value
			T 40	Parametertyp „float“		
			N 01	Parameterindex 1		
6	45		Parameterwert „float“ = 5023,96 dezimal			
7	9C					
8	FF					
9	AE					
10	\r			Wagenrücklauf		Ende
11	\n			Zeilenvorschub		
Gesamtstring			:0803026841459CFFAE\r\n			

3.9.5 Auflistung von ProPar ASCII Beispielen



hexadezimal 7D00 = dezimal 32000
 hexadezimal 3E80 = dezimal 16000
 Float numbers (Gleitkommazahlen) sind dargestellt im 32-bit Single-precision floating-point Format(IEEE-754), z.B. float 3F800000=dec 1
 Strings enthalten ASCII characters z.B. hex 41 is character A, hex 4D = M, hex 6D = m , etc



Es ist wichtig zu wissen, dass nicht alle Parameter an allen ProPar/Multibus-Geräten zur Verfügung stehen.
Weitere Informationen über Parameter und ihre Verwendung sowie eine Beschreibung der digitalen Instrumente finden Sie auch im Dokument Nr. 9.19.023. Wenn Sie das Programm FlowDDE haben, können Sie auch dort einen Überblick bekommen, welche Parameter an welchen Geräten verfügbar sind.

PARAMETER FÜR DEN NORMALBETRIEB

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 :050301010400\r\n

Answer(example) :0403000004 \r\n

Status : 00 (No error)

Control mode: write value = 1 , Process: 1, Parameter: 4, Type: character

Send :050301010401\r\n

Answer(example) :0403000004 \r\n

Status : 00 (No error)

Control mode: write value = 18 = hex 12 , Process: 1, Parameter: 4, Type: character

Send :050301010412\r\n

Answer(example) :0403000004 \r\n

Status : 00 (No error)

Slave factor: read, Process: 33, Parameter: 1, Type: float

Request :06030421412141\r\n

Answer(example) :080302214142C80000\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) :0F800201710A4169522020202020\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) :0803027241009DDDDDD\r\n
 Valve output = 009DDDDDD = 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-/STATUSPARAMETER

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) :06800261260000\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) :048000004 \r\n

Status : 00 (No error)

Reset alarm enable: read , Process: 97, Parameter: 9, Type: character

Request :06800461096109\r\n

Answer(example) :05800261090F\r\n

Value = 0F

Reset alarm enable: write value = 0F = dec 15 , Process: 97, Parameter: 9, Type: character

Send :05800161090F\r\n

Answer(example) :048000004 \r\n

Status : 00 (No error)

ZÄHLERPARAMETER

Counter value: read, Process: 104, Parameter: 1, Type: float

Request :06800468416841\r\n

Answer(example) :0880026841444A6E18\r\n

fmeasure= 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) :048000004\r\n

Status : 00 (No error)

Counter mode: write value = 2 , Process: 104, Parameter: 8, Type: character

Send :058001680802\r\n

Answer(example) :048000004\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) :048000004\r\n

Status : 00 (No error)

counter new setpoint: write value = 0 , Process: 104, Parameter: 6, Type: integer

Send :06800168260000\r\n

Answer(example) :048000005\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) :048000005\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) :058002680500\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) :058002680907 \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) :0480000007 \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) :0480000007 \r\n

Status : 00 (No error)

IDENTIFIKATIONPARAMETER

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 :0780047162716200\r\n

Answer(example) :1A8002716200462D32303143562D354B302D4141442D33332D5600\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

bhtmodel number = 56382E333700 = (ASCII characters) V8.37

usertag: Process: 113, Parameter: 6, Type: string

Request :0780047166716600\r\n

Answer(example) :0D80027166005553455254414700\r\n

usertag = 55534552544147 = (ASCII characters) USERTAG

customer model: read, Process: 113, Parameter: 4, Type: string

Request :0780047164716400\r\n

Answer(example) :0E80027164005354414E4441524400\r\n

Value = 5354414E4441524400 = (ASCII characters) STANDARD

device type: read, Process: 113, Parameter: 1, Type: string [6]

Request :0780047161716106\r\n

Answer(example) :0B8002716106434F52494643\r\n

bhtmodel number = 434F52494643 = (ASCII characters) CORIFC

SPEZIELLE PARAMETER

reset: write value = 0, Process: 115, Parameter: 8, Type: character

Send :058001730800\r\n

Answer(example) :0480000004\r\n

Status : 00 (No error)

reset: write value = 3, Process: 115, Parameter: 8, Type: character

Send :058001730803\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

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)

SPEZIELLE INSTRUMENTFUNKTIONEN

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) :058002010400\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 : 058001010409\r\n

Answer(example) : 0480000004 \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) : 0480000004 \r\n

Status : 00 (No error)

calibration mode: write value = 0 , Process: 115, Parameter: 1, Type: character

Send :058001730100\r\n

Answer(example) : 0480000004 \r\n

Status : 00 (No error)

calibration mode: write value = 9 , Process: 115, Parameter: 1, Type: character

Send :058001730109\r\n

Answer(example) : 0480000004 \r\n

Status : 00 (No error)

3.9.6 Abfragen verketteter Parameter

Schnittstelle sendet eine Abfrage folgender Parameter an das Gerät an Node 3:

Prozess 113: Serial number (3), USERTAG (6)

Prozess 1: Measure (0), Capacity (13), Capacity unit (31), Fluid name (17)

Abfrage verketteter Parameter an Node 3						
Nr.	Byte	Layout	Beschreibung			Beschreibung pro Block
0	:					Start
1	1A		Länge 26			
2	03		Node 3			
3	04		Befehl Lesen			
4	F1	11110001	C	80	Prozess verkettet (Rücksendung)	Prozess 113 (Rücksendung)
			P	71	Prozess 113 (Rücksendung)	
5	EC	11101100	C	80	Parameter verkettet (Rücksendung)	Parameterindex 12 (Rücksendung) Prozess 113 Parameterindex 3 Serial number
			T	60	Parametertyp „string“ (Rücksendung)	
			N	0C	Parameterindex 12 (Rücksendung)	
6	71	01110001	P	71	Prozess 113	Parameterindex 13 (Rücksendung) Prozess 113 Parameterindex 6 USERTAG
7	63	01100011	T	60	Parametertyp „string“	
			P	03	Parameterindex 3 – Serial number	
8	00	00000000		00	Stringlänge 00, Länge nicht definiert	
9	6D	01101101	C	00	Parameter nicht verkettet (Rücksendung)	Parameterindex 13 (Rücksendung) Prozess 113 Parameterindex 6 USERTAG
			T	60	Parametertyp „string“ (Rücksendung)	
			N	0D	Parameterindex 13 (Rücksendung)	
10	71	01110001	P	71	Prozess 113	Parameterindex 14 (Rücksendung) Prozess 1 Parameterindex 0 Measure
11	66	01100110	T	60	Parametertyp „string“	
			P	06	Parameterindex 6 – USERTAG	
12	00	00000000		00	Stringlänge 00, Länge nicht definiert	
13	01	00000001	C	00	Parameter nicht verkettet (Rücksendung)	Prozess 1 (Rücksendung)
			P	01	Prozess 1 (Rücksendung)	
14	AE	10101110	C	80	Parameter verkettet (Rücksendung)	Parameterindex 14 (Rücksendung) Prozess 1 Parameterindex 0 Measure
			T	20	Parametertyp „integer“ (Rücksendung)	
			N	0E	Parameterindex 14 (Rücksendung)	
15	01	00000001	P	00	Prozess 1	Parameterindex 15 (Rücksendung) Prozess 1 Parameterindex 15 Capacity
16	20	00100000	T	20	Parametertyp „integer“	
			P	00	Parameterindex 0 – Measure	
17	CF	11001111	C	80	Prozess verkettet (Rücksendung)	Parameterindex 15 (Rücksendung) Prozess 1 Parameterindex 15 Capacity
			T	40	Parametertyp „float“ (Rücksendung)	
			N	0F	Parameterindex 15 (Rücksendung)	
18	01	00000001	P	01	Prozess 1	Parameterindex 16 (Rücksendung) Prozess 1 Parameterindex 31 Capacity unit
19	4D	01001101	T	40	Parametertyp „float“	
			P	0D	Parameterindex 13 – Capacity	
20	F0	11110000	C	80	Parameter verkettet (Rücksendung)	Parameterindex 16 (Rücksendung) Prozess 1 Parameterindex 31 Capacity unit
			T	60	Parametertyp „string“ (Rücksendung)	
			N	10	Parameterindex 16 (Rücksendung)	
21	01	00000001	P	01	Prozess 1	Parameterindex 17 (Rücksendung)
22	7F	01111111	T	60	Parametertyp „string“	
			P	1F	Parameterindex 31 – Capacity unit	
23	07	00001110		07	Stringlänge 7	
24	71	01110001	C	00	Parameter nicht verkettet (Rücksendung)	Parameterindex 17 (Rücksendung)
			T	60	Parametertyp „string“ (Rücksendung)	

			N	11	Parameterindex 17 (Rücksendung)	Prozess 1 Parameternummer 17 Fluid name
25	01	00000001	P	01	Prozess 1	
26	71	01110001	T	60	Parametertyp „string“	
			P	11	Parameternummer (FBNr.) 17 – Fluid name	
27	0A				Stringlänge 10	
29	\n				Zeilenvorschub	Ende
28	\r				Wagenrücklauf	
Gesamtstring			:1A0304F1EC7163006D71660001AE0120CF014DF0017F077101710A\r\n			

Antwort von Node 3						
Nr.	Byte	Layout	Beschreibung			Beschreibung pro Block
0	:					Start
1	37		Anzahl der folgenden Bytes: 65 Byte			
2	03		Node 3			
3	02		Befehl Schreiben			
4	F1	11110001	C	80	Prozess verkettet	Prozess 113
			P	71	Prozess 113 (Empfangsprozess)	
5	EC	11101100	C	80	Parameter verkettet	Parameterindex 12 Serial number
			T	60	Parametertyp „string“	
			N	0C	Parameterindex 12	
6	00		Länge der Antwort 10 Byte			
7-26			4D 36 32 31 32 33 34 35 41 00 Parameterwert umgewandelt von hexadezimal in ASCII: M6212345A			
27	6D	01101101	C	00	Prozess nicht verkettet	Parameterindex 13 USERTAG
			T	60	Parametertyp „string“	
			N	0D	Parameterindex 13	
28	00		Stringlänge 00, Länge nicht definiert			
29-36			55 53 45 52 54 41 47 00 umgewandelt von hexadezimal in ASCII lautet der Parameterwert: USERTAG			
37	01	00000001	C	00	Prozess nicht verkettet	Prozess 1
			P	01	Prozess 1 (Empfangsprozess)	
38	AE	10101110	C	80	Parameter verkettet	Parameterindex 14 Measure
			T	20	Parametertyp „integer“	
			N	0E	Parameterindex 14	
39	1C		Parameterwert lautet: 1CD8 (hexadezimal)			
40	D8		Messwert lautet: 7384 (dezimal)			
41	CF	11001111	C	80	Parameter verkettet	Parameterindex 15 Capacity
			T	40	Parametertyp „float“	
			N	0F	Parameterindex 15	
42	3F		3F 80 00 00			
43	80		Parameterwert in IEEE-Gleitkommadarstellung, 32-Bit „single precision“			
44	00		umgewandelt von float in dezimal lautet der Parameterwert:			
45	00		1,0			
46	F0	11110000	C	80	Parameter verkettet	Parameterindex 16 Capacity unit
			T	60	Parametertyp „string“	
			N	10	Parameterindex 16	
47	07		Länge der Antwort 7 Byte			
48-54			6D 6C 6E 2F 6D 69 6E umgewandelt von hexadezimal in ASCII lautet der Parameterwert: mln/min			

55	71	01110001	C	00	Parameter nicht verkettet	Parameterindex 17 Fluid name
			T	60	Parametertyp „string“	
			N	11	Parameterindex 17	
56	0A	Länge der Antwort 10 Byte				
57-66			4E 32 20 20 20 20 20 20 20 20 Umgewandelt von hexadezimal in ASCII lautet der Parameterwert: N2			
Gesamtstring		:370302F1EC004D3632313233343541006D00555345525441470001AE1CD8CF3F800000F0076D6C6E2F6D696E710A4E32202020202020 \r\n				



Das ASCII-Zeichen „LEERZEICHEN“ (20 HEX) wird für jedes Leerzeichen im String ausgegeben.

3.9.7 Beispiel abfragen verketteter Parameter, setpoint und measure

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

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

Verkettete Anfrage (Request) von setpoint und measure:

Request :0A80048121012101210120\r\n

Answer(example) :0A800281213E8001213E80\r\n

setpoint = 3E80 = dec. 16000 = 50%

measure = 3E80 = dec. 16000 = 50%

3.9.8 Beispiel abfragen verketteter Parameter, measure und temperature

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

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

Verkettete Anfrage (Request) von measure und temperature:

Request :0A80048121012021472147\r\n

Answer(example) :0C800281213E80214742033089\r\n

measure = 3E80 = dec. 16000 = 50%

temperature = 42033089 = dec. 32.797398 (Degr.C)

Note: Temperature Parameter Wert ist ein 32-bit Single-precision floating-point (Gleitkommazahl) Format (IEEE-754)

3.9.9 Beispiel abfragen verketteter Parameter, fmeasure und temperature

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

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

Verkettete Anfrage (Request) von fmeasure und temperature:

Request :0A8004A140214021472147\r\n

Answer(example) :0E8002A14041000000214741F30956\r\n

fmeasure = 41000000 = dec. 8

temperature = 41F30956 = dec. 30.379559 (Degr.C)

3.10 BEISPIELE - PROPAR VERBESSERTES BINÄRPROTOKOLL

3.10.1 Übertragung Sollwert - ProPar verbessertes Binärprotokoll

Senden des Sollwerts = 50% an Node 3 Prozess: 1, Parameter: 1, typ: integer.

Sollwerte werden in einem Bereich von 0 bis 32000 gesendet, d.h. für dieses Beispiel muss 16000 gesendet werden.

Senden des Sollwerts = 50% an Node 3 Prozess 1 Parameter 1						
Nr	Byte	Layout	Beschreibung			Beschreibung pro Block
0	10		DLE, STX			Start
1	02					
3	01		Nachrichtensequenznummer			Sequenznummer
4	03		Node-Adresse des Empfängers			Node-Adresse
5	05		Länge des Datenfeldes in Bytes			Länge
6	01		Parameter mit Empfängeradresse senden, wird mit Befehl des Typs 00 beantwortet			Kommunikationsbefehl
7	01	c ppppppp 00000001	c	00	Prozess nicht verkettet	Prozessnummer
			p	01	Prozess 1	
8	21	cttppppp 00100001	c	00	Parameter nicht verkettet	Parameternummer
			t	20	Parametertyp 'integer'	
			p	01	Parameternummer (FBnr.) 1	
9	3E		Sollwert 16000 = 3E80h = 50%			Nachrichtenfeld
10	80					
11	10		DLE, ETX			Ende
12	03					
Gesamtstring			10020103050101213E801003			

Antwort von Node 3						
Nr	Byte	Beschreibung				Beschreibung pro Block
0	10	DLE, STX				Start
1	02					
2	01	Nachrichtensequenznummer, wie in der Anfrage				Sequenznummer
3	03	Node-Adresse des Senders, wie in der Anfrage				Node-Adresse
4	03	Länge des Datenfeldes in Bytes				Länge
5	00	Statusmeldung				Kommunikationsbefehl
6	00	Status ok				Nachrichtenfeld
7	05	Index zeigt auf das erste Byte in der Sendenachricht, für das der obige Status gilt. Der Wert dieses Byte kann vernachlässigt werden, wenn der Wert des Byte 6 = 0 ist.				
8	10	DLE, ETX				Ende
9	03					
Gesamtstring			10020103030000051003			

3.10.2 Abfrage Sollwert - ProPar verbessertes Binärprotokoll

Abfrage Sollwert = Node 3 Prozess 1 Parameter 1						
Nr	Byte	Layout	Beschreibung			Beschreibung pro Block
0	10		DLE, STX			Start
1	02					
3	01		Nachrichtensequenznummer			Sequenznummer
4	03		Node-Adresse des Empfängers			Node-Adresse
5	05		Länge des Datenfeldes in Bytes			Länge
6	04		Parameter abfragen, wird mit Befehl des Typs 02 oder 00 beantwortet			Kommunikationsbefehl
7	01	c p p p p p p p 00000001	c	00	Prozess nicht verkettet (Rücksendung)	Prozessnummer
			p	01	Prozess 1 (Rücksendung)	
8	21	c t t n n n n n 00100001	c	00	Parameter nicht verkettet (Rücksendung)	Parameternummer
			t	20	Parametertyp 'integer' (Rücksendung)	
			n	01	Parameterindex 1 (Rücksendung)	
9	01	- p p p p p p p p 00000001	p	01	Prozess 1	Prozessnummer
10	21	- t t p p p p p p 00100001	t	20	Parametertyp 'integer'	Parameternummer
			p	01	Parameternummer (FBnr.) 1 (Sollwert)	
11	10		DLE, ETX			Ende
12	03					
Gesamtstring			100201030504012101211003			

Antwort von Node 3						
Nr	Byte	Layout	Beschreibung			Beschreibung pro Block
0	10		DLE, STX			Start
1	02					
2	01		Nachrichtensequenznummer, wie in der Anfrage			Sequenznummer
3	03		Node-Adresse des Senders, wie in der Anfrage			Node-Adresse
4	05		Länge des Datenfeldes in Bytes			Länge
5	02		Parameter mit Empfängeradresse senden, kein Status abgefragt			Kommunikationsbefehl
6	01	c p p p p p p p 00000001	c	00	Prozess nicht verkettet	Prozess
			p	01	Prozess 1 (Empfangsprozess)	
7	21	c t t n n n n n 00100001	c	00	Parameter nicht verkettet	Parameterindex 1
			t	20	Parametertyp 'integer'	
			n	01	Parameterindex 1	
8	7D		Wert 7D00h = 32000 = 100%			Sollwert
9	00					
10	10		DLE, ETX			Ende
11	03					
Gesamtstring			10020103050201217D001003			

3.10.3 Abfrage Messwert - ProPar verbessertes Binärprotokoll

Abfrage Messwert = Node 3 Prozess 1, Parameter 0, typ integer						
Nr	Byte	Layout	Beschreibung			Beschreibung pro Block
0	10		DLE, STX			Start
1	02					
3	01		Nachrichtensequenznummer			Sequenznummer
4	03		Node-Adresse des Empfängers			Node-Adresse
5	05		Länge des Datenfeldes in Bytes			Länge
6	04		Parameter abfragen, wird mit Befehl des Typs 02 oder 00 beantwortet			Kommunikationsbefehl
7	01	c p p p p p p p 00000001	c	00	Prozess nicht verkettet(Rücksendung)	Prozessnummer
			p	01	Prozess 1 (Rücksendung)	
8	21	c t t n n n n n 00100001	c	00	Parameter nicht verkettet(Rücksendung)	Parameternummer
			t	20	Parametertyp 'integer' (Rücksendung)	
			n	01	Parameterindex 1 (Rücksendung)	
9	01	- p p p p p p p p 00000001	p	01	Prozess 1	Prozessnummer
10	20	- t p p p p p p 00100000	t	20	Parametertyp 'integer'	Parameternummer
			p	00	Parameternummer (FBnr.) 0 (Messwert)	
11	10		DLE, ETX			Ende
12	03					
Gesamtstring			100201030504012101201003			

Antwort von Node 3						
Nr	Byte	Layout	Beschreibung			Beschreibung pro Block
0	10		DLE, STX			Start
1	02					
2	01		Nachrichtensequenznummer, wie in der Anfrage			Sequenznummer
3	03		Node-Adresse des Senders, wie in der Anfrage			Node-Adresse
4	05		Länge des Datenfeldes in Bytes			Länge
5	02		Parameter mit Empfängeradresse senden, kein Status abgefragt			Kommunikationsbefehl
6	01	c p p p p p p p 00000001	c	00	Prozess nicht verkettet	Prozess
			p	01	Prozess 1 (Empfangsprozess)	
7	21	c t t n n n n n 00100001	c	00	Parameter nicht verkettet	Parameterindex 1
			t	20	Parametertyp 'integer'	
			n	01	Parameterindex 1	
8	7D		Wert 7D00h = 32000 = 100%			Messwert
9	00					
10	10		DLE, ETX			Ende
11	03					
Gesamtstring			100201030502012100001003			

3.10.4 Auflistung von ProPar verbesserten Binärbeispielen

PARAMETER FÜR DEN NORMALBETRIEB

measure: read, Process: 1, Parameter: 1, Type: integer

Request 100201800504012101201003

Answer(example) 10020180050101217D001003

measure = hex 7D00 = 32000 = 100%

fmeasure: read, Process: 33, Parameter: 0, Type: float

Request 100201800504214021401003

Answer(example) 1002018007022140417000001003

fmeasure = 41700000 = dec 15

setpoint: read, Process: 1, Parameter: 1, Type: integer

Request 100201800504012101211003

Answer(example) 10020180050201217D001003

measure = hex 7D00 = 32000 = 100%

Setpoint: write value = hex 7D00 = 32000 = 100% ,Process: 1, Parameter: 1, Type: integer

Send 10020180050101217D001003

Answer(example) 10020180030000051003

Status : 00 (No error)

setpoint: write value = hex 3E80 = 16000 = 50% ,Process: 1, Parameter: 1, Type: integer

Send 10020180050101213E801003

Answer(example) 10020180030000051003

Status : 00 (No error)

setpoint: write value = 0, Process: 1, Parameter: 1, Type: integer

Send 100201800501012100001003

Answer(example) 10020180030000051003

Status : 00 (No error)

fsetpoint: read, Process: 33, Parameter: 3, Type: float

Request 100201800504214121431003

Answer(example) 100201800702214140F000001003

fmeasure = 40F00000 = dec 7.5

fsetpoint: write value = float 3F800000 = dec 1, Process: 33, Parameter: 3, Type: float

Send 10020180070121433F8000001003

Answer(example) 10020180030000071003

Status : 00 (No error)

3.10.5 Beispiele verketteter Parameter - ProPar verbessertes Binärprotokoll

measure: Prozess: 1, Parameter: 0, Type: integer

setpoint: Prozess: 1, Parameter: 1, Type: integer

Individuelle Anfrage von measure und setpoint von Node 3 :

Read Measure 1002 01030504 01210120 1003

Read Setpoint 1002 01030504 01210121 1003

Verkettete Anfrage (Request) von measure und setpoint:

Hinweis: Zur Verkettung auf Prozessebene wird das erste Bit der Prozessnummer angegeben, wenn sich ein anderer Prozess anschließt.

Request 1002 01800904 81210120 01210121 1003

measure setpoint

Answer: 10020180090281213E801213E801003

measure = 3E80 = dec. 16000 = 50%

setpoint = 3E80 = dec. 16000 = 50%

4 BETRIEB MIT ZWEI SCHNITTSTELLEN

Für den einwandfreien Betrieb eines Reglers (Aufnahme des Messwerts und Ausgabe des Sollwerts) ist es wichtig, dass der vorgegebene Sollwert von der richtigen Quelle stammt. Sollwerte können verschiedene Quellen haben: analoger Eingang, Feldbus-Schnittstelle oder RS232; oder sie können durch Befehle zum Öffnen oder Schließen des Ventils (Spülen) aufgehoben werden. Deshalb ist es wichtig, die Sollwertquelle des Reglers zu kennen. Diese kann über den Parameter control mode (Prozess 1, Parameter 4) eingestellt werden.

In einigen Fällen ist es möglich, dass Sollwerte gleichzeitig von 2 Quellen kommen. Der letzte Sollwert ist gültig und wird an den Regler gesendet. Dies ist im control mode = 0 der Fall, in dem Sollwerte über eine Feldbus-Schnittstelle oder RS232 kommen können.

Es sind jedoch auch Situationen möglich, in denen eine Einflussnahme auf das Instrument unmöglich scheint. Dies ist der Fall, wenn das Gerät in einen sicheren Zustand versetzt ist, z.B. wenn die Feldbuskommunikation gestört oder unterbrochen ist. Das Ventil wird automatisch in einen sicheren Zustand versetzt: geschlossen (NC) oder voll geöffnet (NO).

Möchten Sie über RS232 wieder Einfluss nehmen können, müssen Sie den control mode ändern. Nimmt der control mode den Wert 18 an, wird der sichere Betriebszustand aufgehoben und über die RS232-Schnittstelle gesendete Sollwerte beeinflussen den Regler wieder.



Eine ausführlichere Beschreibung der Parameter digitaler Instrumente und ihres Verhaltens finden Sie auch im Dokument Nr. 9.19.023.

http://www.bronkhorst.com/en/downloads/instruction_manuals/

5 PARAMETERINFORMATIONEN

ProPar dient dem Austausch von Parameterwerten zwischen Instrumenten und Steuer Geräte (Tastatur oder PC-Schnittstelle). Die Parameterinformationen umfassen verschiedene Verhaltenseigenschaften. In der Tabelle „Parametereigenschaften“ finden Sie eine Liste der Parameter und ihrer Eigenschaften. In der Tabelle „Parameterwerte“ sind die Werte ausführlicher beschrieben. Diese Liste besteht vorwiegend aus Parametern für die Moduseinstellung.

Beschreibung der Eigenschaften in der Tabelle der Parametereigenschaften	
Item	Beschreibung
Parameter(DDE)	Eindeutige Parameternummer (wird auch für DDE-Kommunikation verwendet: P(x))
Name	Name des Parameters, dient zur Parameteridentifikation
process	Prozess, bei dem der Parameter am ProPar -Gerät für die Kommunikation direkt über RS232 verwendet wird; wenn in der Tabelle ein Wert eingetragen ist, muss dieser Wert verwendet werden (für in nur 1 Prozess angesiedelte Parameter); wenn in der Tabelle nichts eingetragen ist, muss der Prozess aus den ProPar -Systeminformationen bestimmt werden (für in mehreren Prozessen angesiedelte Parameter, z.B. setpoint, measure)
FBnr(parameter)	Parameternummer im Prozess am ProPar - Gerät, die für die Kommunikation direkt über RS232 verwendet wird
VarType	Variabler Typ zur Information über die Bytemenge <ul style="list-style-type: none"> • c Typ „(unsigned) char“ 1 Byte Wert 0..255 • l Typ „(unsigned) integer“ 2 Byte Wert 0..65535 • f Typ „float“ 4 Byte Wert +-1,18E-38..+-3,39E+38 (IEEE-Gleitkommadarstellung) • l Typ „(unsigned) long“ 4 Byte Wert 0..4294967295 Datentypen > 1 Byte sind MSB first.
VarLength	Variable Länge zur Angabe der Länge des Zeichenstrings, der in Kombination mit VarType c für die Übertragung von Strings über F ProPar verwendet wird: Wert 0..65535 VarLength gibt die Bytemenge für einen Parametertyp an <ul style="list-style-type: none"> • -2 weist darauf hin, dass ein String mit Null abgeschlossen wird, nicht für Länge definiert • X weist auf einen String mit einer Länge von X Byte (Zeichen) hin • 0 bedeutet, dass keine Informationen nötig sind, d.h. mit Null abgeschlossen.
Min	Zulässiger Mindestwert des Parameters, wenn der Parameter über RS232 gelesen/geschrieben wird; der Wert wird auf diesen Grenzwert hin geprüft (Fehler bei Grenzwertüberschreitung)
Max	Zulässiger Höchstwert des Parameters, wenn der Parameter über RS232 gelesen/geschrieben wird; der Wert wird auf diesen Grenzwert hin geprüft (Fehler bei Grenzwertüberschreitung)
Read	Angabe, ob der Parameter über ProPar gelesen werden darf
Write	Angabe, ob der Parameter über ProPar geschrieben werden darf
Poll	Angabe, ob der Parameter kontinuierlich durch die RS232-Anwendung abgefragt werden soll, damit die (sich ändernden) Parameterinformationen aktuell bleiben
Advanced	Angabe, ob der Parameter nur für fortgeschrittene Benutzer vorgesehen ist; dies sind hauptsächlich Parameter für Wartung/Service
Secured	Angabe, ob der Parameter für die Verwendung durch ProPar gesichert ist; dieser Parameter kann gelesen werden, aber seine Änderung erfordert eine spezielle Handhabung
High security	Angabe, ob der Parameter hoch gesichert ist (nur wenige Parameter); dieser Parameter kann gelesen werden, aber seine Änderung erfordert eine spezielle Handhabung
Description	Kurze Beschreibung der Bedeutung des Parameters oder wofür er verwendet wird
DDE str	Parameter-DDE-String (max. 10 Zeichen), DDE-Wert, wenn der Parameter nicht im Instrument verfügbar ist; dient bis FlowDDE V4.58 auch zur Parameteridentifikation



Parameterakzeptanz:

Parameterwerte, die nicht nur zur Anzeige und nicht abgesichert sind, können geändert werden. Bereiche und Typen der Parameter sind in den Tabellen beschrieben. Wenn Parameterwerte den jeweiligen Bereich überschreiten, werden sie entweder auf den nächsten erlaubten Wert „abgeschnitten“ oder es gibt eine Fehlermeldung: „parameter value error“ (Parameterwertfehler).

(FlowDDE)-Parameternummern:

Jede Parameterinformation bezieht sich auf die Parameternummer. Bei dieser handelt es sich um eine eindeutige Nummer für einen Parameter, um Redundanz zu vermeiden. Diese Nummern werden nur für DDE-Kommunikation benötigt.

Zur Kommunikation mit ProPar über andere Wege als DDE, also direkt über RS232-ASCII-Strings oder über C-Bibliotheken (DOS oder Windows), verwendet man die Parameternummern für die ProPar-Geräte (in der Spalte „FBnr“ der Tabelle der „Parametereigenschaften“). Dazu muss man immer die Node-Adresse“ des Instruments am ProPar, die Prozessnummer und die Parameternummer (FBnr.) kennen, die dem Instrument zugeordnet sind.

Die Prozessnummer wäre aus der Tabelle abzulesen oder ist zu bestimmen, wenn dort nichts ausgefüllt ist. In den meisten Fällen ist die Prozessnummer 1. Auch die Node-Adresse ist zu bestimmen. Dies ist die Node-Adresse des Instruments am ProPar. Neuere RS232-Protokolle an Multibus-Instrumenten akzeptieren Node = 128. An diese Node-Adresse gesendete Nachrichten werden immer akzeptiert, unabhängig von der Node-Adresse des Instruments am Bus.



Es ist wichtig zu wissen, dass nicht alle Parameter an allen ProPar/Multibus-Geräten zur Verfügung stehen.

Weitere Informationen über Parameter und ihre Verwendung sowie eine Beschreibung der digitalen Instrumente finden Sie auch im Dokument Nr. 9.19.023. Wenn Sie das Programm FlowDDE haben, können Sie auch dort einen Überblick bekommen, welche Parameter an welchen Geräten verfügbar sind.

ANHANG 1 UND 2 ENTHALTEN INFORMATIONEN ÜBER PARAMETER, IHRE EIGENSCHAFTEN UND IHRE MÖGLICHEN WERTE.

6 FEHLERSUCHE

6.1 LED-ANZEIGEN

Bei Problemen mit dem Instrument können LED-Anzeigen sehr hilfreich sein.

In der Regel wird die grüne LED für die Statusanzeige des Instruments verwendet, zum Beispiel für normalen Betrieb oder Spezialfunktionsmodus. Für DeviceNet ist auch die Anzeige möglich, dass sich das Instrument im Abbruchzustand und Ruhezustand befindet. Die rote LED dient normalerweise als Fehler-/Warnungsanzeige (je länger sie blinkt, desto gravierender ist die Störung).



Weitere Informationen sind im Handbuch „9.19.023 Betriebsanleitung für digitale Instrumente“ enthalten.

Dieses Dokument finden Sie unter:

http://www.bronkhorst.com/en/downloads/instruction_manuals/

6.2 TIPPS UND HINWEISE ZUR FEHLERSUCHE

SITUATION	Beschreibung
RS232-Kommunikationsprobleme	<p>Kabel prüfen. Sicherstellen, dass die richtigen Kabel für den jeweiligen Zweck verwendet werden.</p> <p>Adresse der Schnittstelle (Slave) prüfen. An Node-Adresse 128 gesendete Nachrichten werden meistens von der Schnittstelle akzeptiert.</p> <p>Reset des Instruments und/oder Neustart Ihres PCs/Ihrer SPS versuchen.</p> <p>Sicherstellen, dass die Nachrichten der Beschreibung des ProPar -Protokolls entsprechen.</p> <p>Sicherstellen, dass die Parameterwerte, die gelesen/geschrieben werden sollen, verfügbar sind und im richtigen Bereich liegen (mit Tabellen vergleichen).</p> <p>Der Regler reagiert nicht auf Sollwerte:</p> <ul style="list-style-type: none"> • Regelmodus prüfen. Wenn 0 und ein anderer Feldbus einen Fehler meldet: sicherer Zustand wird aktiviert, woraus der sichere Sollwert resultiert. Kann durch control mode = 18 umgangen werden (nur bei RS232-Betrieb). • Alarm- oder Zählermodul schaltet Sollwert auf Alarm. Alarm oder Zähler zurücksetzen und fortfahren. • Sollwertkurve (setpoint slope) ist zu steil. Neue Sollwerte werden erreicht, wenn die entsprechende Anstiegszeit abgelaufen ist. Steilheit der Sollwertkurve verringern. • Der Regelmodus hat einen anderen Wert als 0 oder 18. Funktion prüfen, wenn der Wert abweicht. • Wenn sich die Messung sich nicht ändert, Vordruck und Verrohrung prüfen (ggf. auch Absperrventile). • Sicherstellen, dass die Sollwerte im zulässigen Bereich liegen: 0...32000 (= 0...100 %). • Sicherstellen, dass die Sollwerte an das richtige Instrument, den richtigen Prozess (meistens = 1) und Parameter (FBnr für Sollwert = 1) gesendet werden und dass der Datentyp richtig ist (short integer = 2 Byte MSB first).

7 SERVICE

Aktuelle Informationen über Bronkhorst® und Serviceadressen finden Sie auf unserer Website:

 <http://www.bronkhorst.com>

Haben Sie Fragen zu unseren Produkten? Unsere Verkaufsabteilung wird Ihnen gerne helfen, das richtige Produkt für Ihre Anwendung auszuwählen. Wenden Sie sich per E-Mail an den Verkauf:

 sales@bronkhorst.com

oder an Ihren lokalen Vertriebspartner.

Für Kundendienstfragen steht unsere Serviceabteilung mit Hilfe und Beratung zur Verfügung. Kontaktieren Sie den Service per E-Mail:

 support@bronkhorst.com

Ungeachtet der Zeitzone stehen unsere Experten im Betreuungsbereich Ihnen zur Verfügung, um Ihre Fragen umgehend zu beantworten oder für geeignete weitere Maßnahmen zu sorgen. Unsere Experten sind erreichbar unter:

 **+31 859 02 18 66**

8 ANHANG

8.1 TABELLE DER PARAMETERWERTE

Siehe unten.

8.2 TABELLE DER PARAMETEREIGENSCHAFTEN

Siehe unten.

Parameter values table



FlowDDE database version V3.64

2011-05-18 Page 1 of 5

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 subject 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 steering (valve = setpoint)
12	Control mode			21 analog valve steering (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 l/min)
60	Monitor mode			12 volume flow in l/min
60	Monitor mode			13 temperature in °C
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	<AlarmEnable>			0 disable
69	<AlarmEnable>			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/Confrroller 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



FlowDDE database version V3.64

2011-05-18 Page 1 of 5

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	secondary 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		0	1	i		0	32767	Yes	Yes	Yes	No	No	0	setpoint	setpoint: wanted value (100% = 32000)
10	Setpoint slope	18			0	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. polyname, 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 (0xFF 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		alarmmsgTA	alarm message string with unit type information
34	Alarm message number	5			10	1	c	16			Yes	No	Yes	No	No		alarmmsgnr	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	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	0	99235959	Yes	Yes	No	No	No	00000000	duratime	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
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	DSPRegFit	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		CnUnitTypNam	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	1000000000	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	1000000000	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	1000000000	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