Automotive
Energy & Power Analysis
Aerospace & Defense
Transportation
General Test & Measurement

DEWE-Modules

Programmers reference manual

ISO9001

Re-inventing Data Acquisition

worldwide
DEWETRON
www.dewetron.com

Notice

The information contained in this document is subject to change without notice.

DEWETRON elektronische Messgeraete Ges.m.b.H. (DEWETRON) shall not be liable for any errors contained in this document. DEWETRON MAKES NO WARRANTIES OF ANY KIND WITH REGARD TO THIS DOCUMENT, WHETHER EXPRESS OR IMPLIED. DEWETRON SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. DEWETRON shall not be liable for any direct, indirect, special, incidental, or consequential damages, whether based on contract, tort, or any other legal theory, in connection with the furnishing of this document or the use of the information in this document.

Warranty Information:

A copy of the specific warranty terms applicable to your DEWETRON product and replacement parts can be obtained from your local sales and service office.

Restricted Rights Legend:

Use austrian law for duplication or disclosure.

DEWETRON GesmbH
Parkring 4
8074 Graz-Grambach
Austria

Printing History:

Version 2.1.2
Released March 28, 2006

Notice

Safety symbols in the manual

	*Indicates hazardous voltages.*

WARNING	*Calls attention to a procedure, practice, or condition that could cause bodily injury or death.*

CAUTION	*Calls attention to a procedure, practice, or condition that could possibly cause damage to equipment or permanent loss of data.*

WARNINGS

The following general safety precautions must be observed during all phases of operation, service, and repair of this product. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the product. DEWETRON Elektronische Messgeraete Ges.m.b.H. assumes no liability for the customer's failure to comply with these requirements.

All accessories shown in this document are available as option and will not be shipped as standard parts.

Safety instructions for DEWETRON amplifiers

- The DEWETRON data acquisition systems and amplifiers may only be installed by experts.
- Read your manual carefully before operating.
- Observe local laws when using the amplifiers.
- Ground the equipment: For Safety Class 1 equipment (equipment having a protective earth terminal), a non interruptible safety earth ground must be provided from the mains power source to the product input wiring terminals or supplied power cable.
- DO NOT operate the product in an explosive atmosphere or in the presence of flammable gases or fumes.
- DO NOT operate damaged equipment: Whenever it is possible that the safety protection features built into this product have been impaired, either through physical damage, excessive moisture, or any other reason, REMOVE POWER and do not use the product until safe operation can be verified by service-trained personnel. If necessary, return the product to a DEWETRON sales and service office for service and repair to ensure that safety features are maintained.
- Keep away from live circuits: Operating personnel must not remove equipment covers or shields. Procedures involving the removal of covers or shields are for use by service-trained personnel only. Under certain conditions, dangerous voltages may exist even with the equipment switched off. To avoid dangerous electrical shock, DO NOT perform procedures involving cover or shield removal unless you are qualified to do so.
- No modifications are allowed at the amplifiers.
- DO NOT service or adjust alone. Do not attempt internal service or adjustment unless another person, capable of rendering first aid and resuscitation, is present.
- DO NOT substitute parts or modify equipment: Because of the danger of introducing additional hazards, do not install substitute parts or perform any unauthorized modification to the product. Return the product to a DEWETRON sales and service office for service and repair to ensure that safety features are maintained.
- DO NOT touch internal wiring!
- DO NOT use higher supply voltage than specified!
- Use only original plugs and cables for harnessing.
- Safety of the operator and the unit depend on following these rules.

Support

For any support please contact your local distributor first or DEWETRON directly.

For Asia and Europe, please contact:

DEWETRON Ges.m.b.H.
Parkring 4
A-8074 Graz-Grambach
AUSTRIA
Tel.: +43 316 3070
Fax: +43 316 307090
Email: support@dewetron.com
Web: http://www.dewetron.com

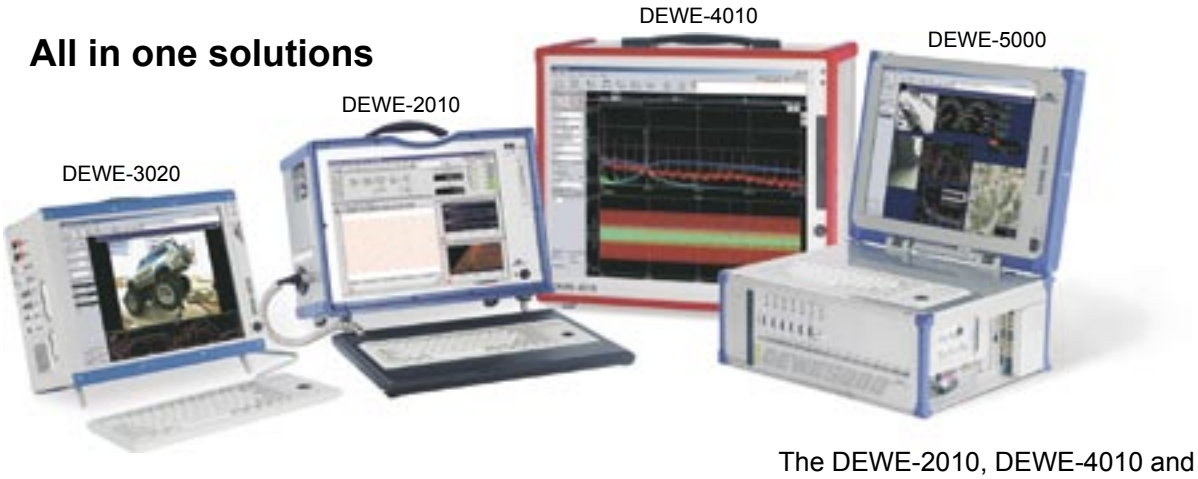
The telephone hotline is available Monday to Friday between 08:00 and 12:00 CET (GMT -1:00) and Monday to Thursday between 13:00 and 17:00 CET.

For the Americas, please contact:

DEWETRON, Inc.
10 High Street, Suite K
Wakefield, RI 02879
U.S.A.
Tel.: +1 401 284 3750
Toll-free: +1 877 431 5166
Fax: +1 401 284 3755
Email: support@dewamerica.com
Web: http://www.dewamerica.com

The telephone hotline is available Monday to Friday between 08:00 and 17:00 GST (GMT +5:00)

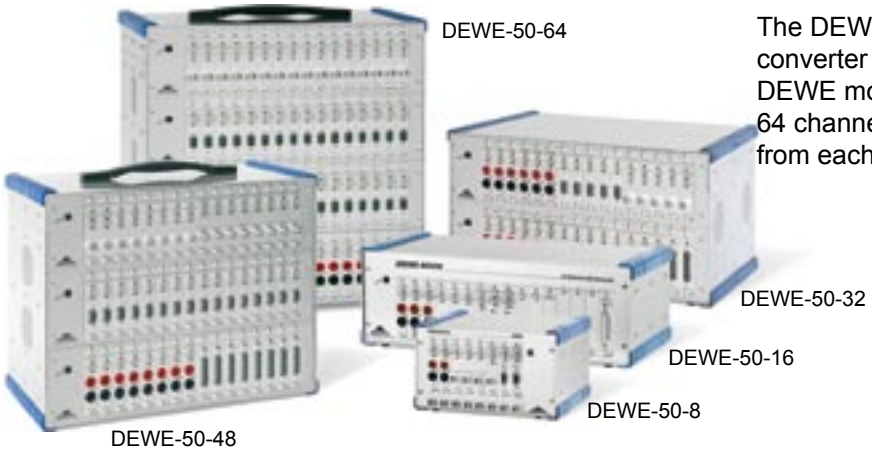
All in one solutions



The DEWE-2010, DEWE-4010 and DEWE-5000 offers 16 slots, the DEWE-3020 8 slots for DEWE modules. All systems are expandable up to several hundred channels.

Signal conditioning solutions

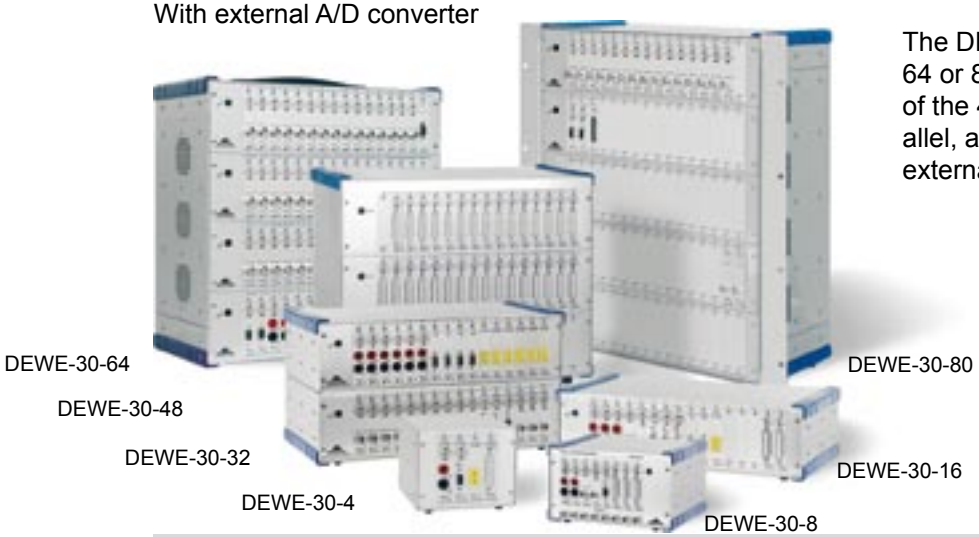
With internal A/D converter



The DEWE-50 series with integrated A/D converter offers 8, 16, 32, 48 or 64 slots for DEWE modules and except of the 48 and 64 channel version, in parallel, a ±5 V output from each module to an external device.

Signal conditioning solutions

With external A/D converter



The DEWE-30 family offers 4, 8, 16, 32, 48, 64 or 80 slots for DEWE modules and except of the 48, 64 and 80 channel version, in parallel, a ±5 V output from each module to an external device.

DEWE-Modules Overview

Analog input amplifiers (DAQx series)										
Module	**Input connector**	**# CH**	**Prog. Ranges & Filter**	**TEDS**	**Ranges**	**Filters**	**Bandwidth**	**Isolation**	**Output**	**Details on page**
High voltage amplifier										
DAQN-DMM	Banana plugs	1			±10, ±40, ±100, ±200, ±400, ±1000V	10, 100 Hz, 1, 3 kHz	3 kHz	1.5 kV_{RMS}	±5 V	15
DAQP-DMM	Banana plugs	1	✓		±10, ±40, ±100, ±200, ±400, ±1000V	10, 100 Hz, 1, 3, 20 / 30 kHz	3 kHz	1.5 kV_{RMS}	±5 V	15
DAQP-HV	Banana plugs	1	✓		±20, ±50, ±100, ±200, ±400, ±800, ±1400 V	10, 30, 100, 300 Hz 1, 3, 10, 30, 100, 300 kHz	180 kHz	1.8 kV_{RMS}	±5 V	17
Voltage amplifier										
DAQN-V-B	Banana plugs	1			±0.01, ±0.1, ±1, ±5, ±10, ±50 V	10, 100 Hz, 1, 10 kHz	10 kHz	1 kV_{RMS}	±5 V	19
DAQN-V-BNC	BNC							1 kV_{RMS}		
DAQN-V-D	9-pin SUB-D							350 V_{DC}		
DAQP-V-B	Banana plugs	1	✓		±0.01, ±0.1, ±1, ±5, ±10, ±50 V	10, 100 Hz, 1, 10, 50 kHz	50 kHz	1 kV_{RMS}	±5 V	19
DAQP-V-BNC	BNC							1 kV_{RMS}		
DAQP-V-D	9-pin SUB-D							350 V_{DC}		
DAQP-V-LEMO	7-pin LEMO							350 V_{DC}		
DAQP-LV-B-B	Banana plugs	1	✓	✓	±10, ±20, ±50, ±100, ±200, ±500 mV ±1, ±2.5, ±5, ±10, ±25, ±50 V	10, 30, 100, 300 Hz 1, 3, 10, 30, 100, 180 kHz	300 kHz	350 V_{DC}	±5 V	23
DAQP-LV-B-BNC	BNC									
DAQP-LV-B-D	9-pin SUB-D									
DAQP-LV-B-LEMO	7-pin LEMO									
Current amplifier										
DAQP-LA-B	Banana plugs	1	✓		0.1A, 0.3 A, 1 A, 3 A, (10 A, 30 A peak)	10, 30, 100, 300, 1000 Hz, 3, 10, 30, 100, 180, 300[1] kHz	300 kHz	1.4 kV_{RMS}	±5 V	33
DAQP-LA-SC	Screw terminals				2, 6, 20, 60, 200, 600 mA					
Bridge / strain gage amplifier										
DAQP-STG-D	9-pin SUB-D	1	✓	✓	±0.5 , ±1, ±2.5, ±5, ±10, ±25, ±50, ±100, ±250, ±500 mV, ±1 V, ±2V, ±5 V,±10 V	10, 30, 100, 300 Hz, 1, 3, 10, 30, 100, 300 kHz	300 kHz	350 V_{DC}	±5 V	37
DAQP-BRIDGE-A	9-pin SUB-D	1	✓		±1, ±2, ±5, ±10, ±20, ±50 mV/V (@ 5 V_{DC})	10 Hz, 100 Hz, 1 kHz, 5 kHz	20 kHz	350 V_{DC}	±5 V	45
DAQP-BRIDGE-A-LEMO	8-pin LEMO									
DAQP-BRIDGE-B	9-pin SUB-D	1	✓	✓	±0.1, ±0.2, ±0.5, ±1, ±2, ±5, mV/V ±10, ±20, ±50, ±100 mV/V (@ 5 V_{DC})	10 Hz, 100 Hz, 1 kHz, 5 kHz	20 kHz	-	±5 V	49
DAQP-BRIDGE-B-LEMO	8-pin LEMO									
Carrier frequency amplifier										
DAQP-CFB	9-pin SUB-D	1	✓		±0.1 to ±1000 mV/V	10, 30, 100, 300 Hz, 1 kHz	2.3 kHz	-	±5 V	57
Charge / IEPE® amplifier for vibration measurement										
DAQP-ACC-A	BNC	1	✓		IEPE®: ±50, ±166, ±500 mV, ±1.66, ±5 V	1, 10, 100, 300 kHz	0.5 Hz to 300 kHz	-	±5 V	63
DAQP-CHARGE-A	BNC	1	✓		Charge: 5, 50, 500, 5000, 50000 pC IEPE®: ±5, ±50, ±500 mV, ±5 V	1 kHz, 5 kHz, 10 kHz, 20 kHz	0.1 Hz to 50 kHz	-	±5 V	65
DAQP-CHARGE-B	BNC	1	✓		Charge: ±100, ±500, ±2 000, ±10 000, ±40 000, ±200 000, ±1 000 000 pC	10, 30, 100, 300 Hz, 1, 3, 10, 30, 100 kHz	DC to 100 kHz	350 V_{DC}	±5 V	69
Frequency to voltage converter										
DAQP-FREQ-A	9-pin SUB-D	1	✓		100 Hz, 1, 5, 20, 100, 200 kHz	100 Hz, 1, 5, 20, 100, 200 kHz	according to range	350 V_{DC}	±5 V	73
Multifunctional amplifier										
DAQP-MULTI	9-pin SUB-D Mini-TC	1	✓	✓	Min. to max. of the input range is free programmable within the full thermocouple input span	6 progr. low pass filter (3Hz to 3 kHz) and progr. filter orders (2^{nd}, 4^{th}, 6^{th}, 8^{th})	3 kHz	1 kV_{RMS}	±5 V; 0 to ±5 V[2]	77
Thermocouple amplifier										
DAQP-THERM	Mini-TC	1	✓	✓	Min. to max. of the input range is free programmable within the full thermocouple input span	6 progr. low pass filter (3Hz to 3 kHz) and progr. filter orders (2^{nd}, 4^{th}, 6^{th}, 8^{th})	3 kHz	1 kV_{RMS}	±5 V; 0 to ±5 V[2]	77

[1] 300 kHz exclusively for Bessel filter characterisic

[2] ±10 V and 0 to 10 V with special DEWE-30

Analog input amplifiers, continued (DAQx series)										
Module	**Input connector**	**# CH**	**Prog. Ranges & Filter**	**TEDS**	**Ranges**	**Filters**	**Bandwidth**	**Isolation**	**Output**	**Details on page**
Thermocouple amplifier										
DAQN-THERM-1	Mini-TC	1			‚K': -30 °C to 170 °C	-	4 Hz	1 kV_{RMS}	±5 V	85
DAQN-THERM-2					‚K': -30 °C to 370 °C					
DAQN-THERM-3					‚K': 0 °C to 1000 °C					
DAQN-THERM-4					‚K': -100 °C to 1350 °C					
DAQN-THERM-5					‚J': -100 °C to 760 °C					
DAQN-THERM-SPEC	Mini-TC	1			customer defined range and type	-	up to 300 Hz			
RTD amplifier										
DAQN-RTD-1	9-pin SUB-D	1			Pt100 -30 °C to 170 °C	-	10 Hz	-	±5 V	89
DAQN-RTD-2					Pt100 -100 °C to 200 °C					
DAQN-RTD-3					Pt100 -50 °C to 600 °C					
DAQN-RTD-SPEC	9-pin SUB-D	1			customer defined range	-	on request	-	±5 V	89
Potentiometric and ohmic amplifier										
DAQN-OHM	9-pin SUB-D	1			0 to 100 % (resistors from 100 Ω to 10 kΩ)	-	100 Hz	350 V_{DC}	±5 V	91
1:1 analog voltage input										
DAQN-AIN-B	Banana plugs	1			depending on A/D board (1:1 input)	-		overvoltage protection (< ±500 V)	max. ±10 V	93
DAQN-AIN-BNC	BNC									
DAQN-AIN-D	9-pin SUB-D									
Customer defined modules										
DAQN-CUSTOM-B	Banana plugs	customer defined, prototype board inside							max. ±10 V	95
DAQN-CUSTOM-BNC	BNC									
DAQN-CUSTOM-D	9-pin SUB-D									

Analog output amplifiers (DAQx series)										
Voltage output module										
DAQN-V-OUT-B	Banana plugs	1			1:1 output module with isolation Input voltage: ±10 V Output voltage: ±10 V	-	400 Hz	240 V_{RMS}	max. ±10 V	97
DAQN-V-OUT-BNC	BNC									
DAQN-V-OUT-D	9-pin SUB-D									

Amplifiers with integrated A/D converter and DIO modules (PAD series)										
Voltage / current amplifier										
PAD-V8-P	25-pin SUB-D	8	✓		±100, ±150, ±500 mV, -150 mV to +1.5 V, ±1, ±2.5, ±5, ±10, ±50 V	1 / 4 / 8 values averaging	3 Hz	350 V_{DC}	RS232/485	99
High accuracy thermocouple and RTD amplifier										
PAD-TH8-P	25-pin SUB-D	8	✓		±15, ±50, ±100, ±150 mV, -150 mV to +1.5 V, Thermocouple type J, K and T	1 / 4 / 8 values averaging	3 Hz	350 V_{DC}	RS232/485	103
PAD-TH8-P + CB8-RTD	9-pin SUB-D (8x)	8	✓		Pt100, Pt200, Pt500, Pt1000, Pt2000, Ni120	1 / 4 / 8 values averaging	3 Hz	350 V_{DC}	RS232/485	106
Analog output module										
PAD-AO1	25-pin SUB-D	1	✓		0 to 20 mA, 4 to 20 mA, 0 to 10 V	-		300 V_{DC}	RS232/485	109
Frequency / counter module										
PAD-CNT2	25-pin SUB-D	2	✓		32 bit counter; low: 0 to 1 V, high: 3.5 to 30 V	-	1 Hz to 100 kHz	300 V_{DC}	RS232/485	111
Digital input / output module										
PAD-DI8	25-pin SUB-D	8	✓		Opto input low: 0 to 1 V, high: 3.5 to 30 V	-		300 V_{DC}	RS232/485	113
PAD-DO7	25-pin SUB-D	7	✓		Relay outputs (dry contacts)	-		300 V_{DC}	RS232/485	115

Notes

Calibration information

All DEWETRON modules are calibrated at 25 °C and meet their specifications when leaving the factory. The time interval for recalibration depends on environmental conditions. Typically, the calibration should be checked once a year.

Calibration certificates are available from DEWETRON as an option. DEWETRON offers two types:

- ISO traceable DEWETRON certificate
- Calibration certificate according to ÖKD (equivalent to DKD)

This manual contains no calibration information. For self calibration, there is a separate calibration kit for the DAQ series modules available. The CAL-KIT contains the required cables, software and instructions.

Adjustment information are only mentioned if they are required for operation (e.g. DAQP-TRQ).

General module specifications

Module dimensions:	20 x 65 x 105 mm (W x H x D without front cover and connectors)	(0.79 x 2.56 x 4.13 in.)
Frontcover:	20 x 87 x 2 mm (W x H x D without connector)	(0.79 x 3.43 x 0.08 in.)
Environmental:		
Temp. range storage:	-30 °C to +85 °C	(-30 °F to 185 °F)
Temp. range operating:	-5 °C to +60 °C	(-4 °F to 140 °F)
Relative humidity (MIL202):	0 to 95 % at 60 °C, non-condensing	
RFI susceptibility:	±0.5 % span error at 400 MHz, 5 W, 3 m	

All specifications within this manual are valid at 25 °C!

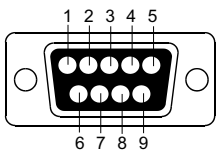
All modules are produced according ISO9001 and ISO14001.

Module connectors

Frontpanel connector: Accessable to the user. The connector type and pin assignment varies from module to module. Detailed pin assignment of each module is shown in the appropriate module description.

Rear connector: 9-pin male SUB-D, interface to the DEWE-System, not accessable to the user.

DAQx and PAD module rear view



9-pin male SUB-D connector

Interface pin assignment:

1 Module output (±5 V)
2 RS-485 (A)
3 RS-485 (B)
4 GND
5 +9 V power supply
6 +12 V power / sensor supply
7 Module input (from D/A converter of the A/D board)[1)]
8 reserved
9 -9 V power supply

[1)] Triggerout at DAQP-FREQ-A

General Module Information

RS-232/485 interface

DAQP modules can be configured via RS-485 interface, PAD modules require this interface for all data transfers.

For all DEWETRON systems, an internal RS-232/485 converter is available (standard with DEWE-800, DEWE-2000, DEWE-2500, DEWE-3000, DEWE-4000 and DEWE-5000 systems). This converter allows communication with DAQ and PAD modules.

To communicate with the modules, the RS-232 interface has to be set to the following parameters:

baud rate: 9600 bps
data bits: 8
parity: no
stop bits: 1
handshake: not required

DAQP module configuration

The DAQP modules offer two possibilities to change the measurement range and the input filters:

1.) Push button selection

All ranges and filters can be selected directly by pressing the push buttons on the module. Approx. 15 sec. after changing range and/or filter, the range and filter information is stored in an EEPROM. This procedure increases the livetime of the EEPROM.

The current input range setting is shown all the time by LED. To change the range just press RANGE button a few times until the required range is displayed.

To sse the current filter setting just press the FILTER button once. The corresponding LED is flashing for approx. 3 seconds. Within this time, the filter can be selected by pressing the FILTER button again. 10 to 15 seconds after the last key activity, the information will be stored, the LED stops flashing and shows the input range again.

CAUTION: Power loss during this time leaves the module in the former settings!

2.) RS-232/485 programming

All ranges and filters also can be selected via RS-232/485 interface. All DEWE-800, DEWE-2000, DEWE-2500, DEWE-3000, DEWE-4000 and DEWE-5000 systems are prepared as a standard to work with DAQP modules.

The easiest way to change the configuration is to use the DEWE-CONFIG software, which comes as a standard with the DEWETRON data acquisition system.

Detailed information about DAQP modules programming for customer application is available in the corresponding module section.

CAUTION: ALL range and filter changes whcih are done via RS-232/485 interface are not stored in the EEPROM of the DAQP modules. You have to store this information in a seperate initialisation file to keep settings information for next system start.

DAQP programming commands

The following commands are available for all DAQP modules. Detailed command description is available in the corresponding module section.

1.) Set module address

Change the current module address to a new address. To change address sent the 'set module address' command while the filter button on the module gets pressed. After that, the module sends back an acknowladge response, which contains the current module settings.

The module address always has to be a 2 digit hex value. We recommend addresses between 0x00 and 0xFE. Some commands uses 0xFF to act all modules in rack.

2.) Read module configuration

The response contains the currrent module setting.

3.) Set module configuration

This command allows the change of the current module setting.

4.) Lock/Unlock module buttons

Lock or unlock the push button access to range and filter selection.

5.) TEDS programming commands

Read TEDS chip type

SomeModule supports Transducer Electronic Data Sheet. TEDS is based on 1-wire E²PROM. For more detailed information about this technology please refer to the datasheet of the specific E²Prom (TEDS Chip).

Command: ??(Addr)TEDS\r

??:	Command leading code
Addr:	Module address (2 characters hex)
TEDS:	Command
\r:	Carriage return (0x0D)

Response: !(Addr)(FamilyCode)(Serial)(CRC)\r

!:	Response leading code
Addr:	Channel Number
FamilyCode:	(2 characters hex)

12h:	DS2406
14h:	DS2430A
23h:	DS2433
33h:	DS2432

Serial:	(12 characters hex) Unique Serial Number
CRC:	(2 characters hex) For CRC calculation refer to the TEDS Datasheet
\r:	Carriage return (0x0D)

Error Message: !(Channel)TEDER(ERR)\r

Channel:	Channel number (2 characters hex from 0x00 to 0xFE)
TEDER:	Response
ERR:	Supported Error codes: 00, 07; refer to ‚Error messages‘

Example:

Command:	??04TEDS
Response:	!042300000025DD905A
Explanation:	Channel 4; DS2433; SN.: 2481552; CRC: 5A

TEDS Error Messages

ERR:	Error code
00:	No 1-Wire chip connected to the module.
01:	1-Wire chip not supported – see list above for supported types.
02:	Error during reading data from the E²PROM.
03:	Wrong command length from host received.
04:	Check sum error detected (data transfer error from host to module).
05:	Application register is already written (only at DS2430A!).
06:	Write error to E²PROM: Check error after writing the data.
07:	Data line short circuit.
08:	Memory area not found.
09:	Device not supported for programming.
10:	1-wire ATR not OK; no answer from 1-wire device
11:	1-wire master in fail state.

Read TEDS E²PROM

Command: ??(Addr)TEDS(M) or (MM)\r

??:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
TEDS:	Command
M or MM:	Memory Register

0..3	DS2406
0..1	DS2430A Memory Register 1 only one time programmable
0..F	DS2433
0..3	DS2432
0..50	DC28EC20 (DAQP-MULTI/-THERM; 0..F for other modules)

\r: Carriage return (0x0D)

Response: !(Addr)(M) or (MM)(DATA)\r

!:	Response leading code
Addr:	Module address
M or MM:	Memory address
DATA:	64 characters ASCII codes (32 ASCII characters)
CHS:	Check sum: last two digits of Memory Register + ASCII [0]+ ASCII[1]…+ ASCII[64]
\r:	Carriage return (0x0D)

Error Message: !(Addr)TEDER(ERR)\r

Addr:	Module address (2 characters hex from 0x00 to 0xFE)
TEDER:	Response
ERR:	Supported Error codes: 00;01;02;07; refer to ‚Error Messages‘

Example: Command: ??04TEDS1

Response: !0448656C6C6F20776F726C64202020202020202020202020202020202020202020202020DD\r

Explanation: Address 4;
48=> H
65=> e
:
Hello world Check Sum: 1+48+65+6C+..= 6DD

Write TEDS E²PROM

Command: ##(Addr)TEDS(M) or (MM)(DATA)(CHS)\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
TEDS:	Command
M or MM:	Memory Register

0..3	DS2406
0..1	DS2430A Memory Register 1 only one time programmable
0..F	DS2433
0..3	DS2432
0..50	DC28EC20 (DAQP-MULTI/-THERM; 0..F for other modules)

DATA:	64 characters ASCII codes (32 ASCII characters)
CHS:	Check sum: last two digits of Memory Register + ASCII [0]+ ASCII[1]…+ ASCII[64]
\r:	Carriage return (0x0D)

Response: !(Addr)\r

!:	Response leading code
Addr:	Module address
\r:	Carriage return (0x0D)

Error message: !(Addr)TEDER(ERR)\r

Addr:	Module Address (2 characters hex from 0x00 to 0xFE)
TEDER:	Response
ERR:	Supported Error codes: 00;01;03;04;05;06;07; refer to ‚Error messages‘

Example: Command: ##04TEDS148656C6C6F20776F726C64202020202020202020202020202020202020202020DD\r

Response: !04\r

Explanation: Addr 4;

48=> H
65=> e
:
Hello world Check Sum: 1+48+65+6C+..= 6DD

Read TEDS I/O state (only DS2406 & DC28EC20)

This function is only supported by the DS2406. Refer to the supplier datasheet for further information.

Command: ??(Addr)TEDSOUT\r

??:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
TEDSOUT:	Command
\r:	Carriage return (0x0D)

Response: !(Addr)(D)(S)(CHS)\r

!:	Response leading code
Addr:	Module address
D:	GPIO Bit (1 Byte ASCII Hex; representing the state of the GPIO ports A & B. Bits 5 & 6 of the SRAM S. B.) 0: both ports LOW 1: Port A = HIGH and Port B = LOW 2: Port A = LOW and Port B = HIGH 3: Port A = HIGH and Port B = HIGH
S:	Channel info Byte (2 Characters ASCII hex): Refer to the supplier datasheet
CHS:	Check Sum: last two digits of D + S
\r:	

Error Message: !(Addr)TEDER(ERR)\r

Addr:	Module address (2 characters hex from 0x00 to 0xFE)
TEDER:	Response
ERR:	Supported Error codes: 00;01;02;07; refer to ‚Error messages‘

Example: Command: ??K04TEDSOUT\r

Response: !0704343\r

Explanation: Addr 0x04; Port A LOW; Port B LOW; CHS: 0+43 = 43;

Write TEDS I/O pin (only DS2406 & DC28EC20)

This function is only supported by the DS2406. Refer to the supplier datasheet for further information.

Command: ##(Addr)TEDSOUT(D)\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
TEDSOUT:	Command
D:	GPIO Bit (1 Byte ASCII Hex; representing the state of the GPIO ports A & B. Bits 5 & 6 of the SRAM S. B.) 0: both ports LOW 1: Port A = HIGH and Port B = LOW 2: Port A = LOW and Port B = HIGH 3: Port A = HIGH and Port B = HIGH
\r:	Carriage return (0x0D)

Response: !(Addr)ACK\r

Error Message: !(Addr)TEDER(ERR)\r

Addr:	Module address (2 characters hex from 0x00 to 0xFE)
TEDER:	Response
ERR:	Supported Error codes: 00;01;06;07; refer to ‚Error messages‘

Example: Command: # #04TEDSOUT0\r

Response: !04ACK\r

Explanation: Addr 0x04; Port A LOW; Port B LOW;

PAD module programming

Due to different commands, detailed information about module programming is available for each module in the corresponding section.

Be aware that the module address always has to be a 2 digit hex value! We recommend addresses between 0x00 and 0xFE. Some commands uses 0xFF to act all modules in rack.

CAUTION: DO NOT MIX UP '#' and '$' COMMAND SIGNS!
TO GET DATA FROM A MODULE, USE ONLY THE '#' COMMAND SIGN,
OTHERWISE A MODULE-RECALIBRATION IS POSSIBLE.
THIS REQUIRES A NEW MODULE CALIBRATION!

Setting communication speed

A temporary short circuit between 21 and 22 (GND) offers the possibility to change the baud rate. When changing the baud rate of TH8 and V8 modules, additional press the ID button on the module front. After sending the mentioned command turn power off and remove the short circuit from pin 21. At next powering on the baud rate changes become effective.

Communication speed can be up to 115200 bps for PAD modules. You have to set all modules in a rack to the same speed. Also the rack has to be jumpered at the same speed (DEWE-RACK-4 has no jumper to change the communication speed of 9600 bps.) When using PAD and DAQ modules in one rack, you have to set the communication speed to 9600 bps.

Be aware that the module address always has a 2 digit hex value!

Module reset

A short circuit between pin 21 and 22 (GND) during power on the module effect a reset of the module default settings (9600 baud, 8 data bits, 1 stop bit, no parity, module address 0x00). To reset TH8 and V8 modules, additional press the ID button on the module front during power on the module.

General Module Information

CPAD2 module general information

Valid range for extended identifier: 29 Bit = 0 .. 2^29-1 = 0 .. 536870911

Possible EPAD SNo Range: 200 000 to 16 777 215

SNoIdent (= Identifier for access to individual modules):

0x1C000000 + DataBit (=Bit 25) + SNo + ChnBit (Bit 0)

DataBit: “0”, if host writes commando to Module
“0”, if module answers on request
“1” if modules sends data

ChnBit: “1”: if measurement results or configuration is send from CH4 to Ch7
“0”: on all other conditions

GlobIdent (= Identifier for access to all modules) = 0x1C000000

SyncIdent (= Identifier send by the master model if active) = 0x1C000002

DataIdent (= Identifier if CAN_Address is used for sending the data)

a) Extended Identifier Format
Bit 29 .. Bit 25 = 0
Bit 24 .. Bit 1 = CAN_ADDRESS
Bit 0 = ChnBit

b) Standard Identifier Format
Bit 10 .. Bit 1 = CAN_ADDRESS (lower 10 Bits)
Bit 0 = ChnBit

All configurations are sent in Intel Data Format!

DLC (=DataLenghtCode) = Number of data byte to be send

Data output format: 16 Bit scaled to full range

Examples:

Type K (-270 to 1372°C): 0x0000 = -270°C; 0xFFFF = 1372 °C
Type J (-210 to 1200°C): 0x0000 = -210°C; 0xFFFF = 1200 °C
Voltage range: Offset binary: 0x0000 = min Range, 0xFFFF = max. Range
Example: -150mV to +1500 mV: 0x0000 = -150mV; 0xFFFF = 1500 mV
Example: -50mV to +50 mV: 0x0000 = -50mV; 0xFFFF = +50 mV

Conversion table: decimal to hexadecimal and binary

Dec	Hex	Bin	Dec	Hex	Bin	Dec	Hex	Bin	Dec	Hex	Bin
0	00	00000000	64	40	01000000	128	80	10000000	192	C0	11000000
1	01	00000001	65	41	01000001	129	81	10000001	193	C1	11000001
2	02	00000010	66	42	01000010	130	82	10000010	194	C2	11000010
3	03	00000011	67	43	01000011	131	83	10000011	195	C3	11000011
4	04	00000100	68	44	01000100	132	84	10000100	196	C4	11000100
5	05	00000101	69	45	01000101	133	85	10000101	197	C5	11000101
6	06	00000110	70	46	01000110	134	86	10000110	198	C6	11000110
7	07	00000111	71	47	01000111	135	87	10000111	199	C7	11000111
8	08	00001000	72	48	01001000	136	88	10001000	200	C8	11001000
9	09	00001001	73	49	01001001	137	89	10001001	201	C9	11001001
10	0A	00001010	74	4A	01001010	138	8A	10001010	202	CA	11001010
11	0B	00001011	75	4B	01001011	139	8B	10001011	203	CB	11001011
12	0C	00001100	76	4C	01001100	140	8C	10001100	204	CC	11001100
13	0D	00001101	77	4D	01001101	141	8D	10001101	205	CD	11001101
14	0E	00001110	78	4E	01001110	142	8E	10001110	206	CE	11001110
15	0F	00001111	79	4F	01001111	143	8F	10001111	207	CF	11001111
16	10	00010000	80	50	01010000	144	90	10010000	208	D0	11010000
17	11	00010001	81	51	01010001	145	91	10010001	209	D1	11010001
18	12	00010010	82	52	01010010	146	92	10010010	210	D2	11010010
19	13	00010011	83	53	01010011	147	93	10010011	211	D3	11010011
20	14	00010100	84	54	01010100	148	94	10010100	212	D4	11010100
21	15	00010101	85	55	01010101	149	95	10010101	213	D5	11010101
22	16	00010110	86	56	01010110	150	96	10010110	214	D6	11010110
23	17	00010111	87	57	01010111	151	97	10010111	215	D7	11010111
24	18	00011000	88	58	01011000	152	98	10011000	216	D8	11011000
25	19	00011001	89	59	01011001	153	99	10011001	217	D9	11011001
26	1A	00011010	90	5A	01011010	154	9A	10011010	218	DA	11011010
27	1B	00011011	91	5B	01011011	155	9B	10011011	219	DB	11011011
28	1C	00011100	92	5C	01011100	156	9C	10011100	220	DC	11011100
29	1D	00011101	93	5D	01011101	157	9D	10011101	221	DD	11011101
30	1E	00011110	94	5E	01011110	158	9E	10011110	222	DE	11011110
31	1F	00011111	95	5F	01011111	159	9F	10011111	223	DF	11011111
32	20	00100000	96	60	01100000	160	A0	10100000	224	E0	11100000
33	21	00100001	97	61	01100001	161	A1	10100001	225	E1	11100001
34	22	00100010	98	62	01100010	162	A2	10100010	226	E2	11100010
35	23	00100011	99	63	01100011	163	A3	10100011	227	E3	11100011
36	24	00100100	100	64	01100100	164	A4	10100100	228	E4	11100100
37	25	00100101	101	65	01100101	165	A5	10100101	229	E5	11100101
38	26	00100110	102	66	01100110	166	A6	10100110	230	E6	11100110
39	27	00100111	103	67	01100111	167	A7	10100111	231	E7	11100111
40	28	00101000	104	68	01101000	168	A8	10101000	232	E8	11101000
41	29	00101001	105	69	01101001	169	A9	10101001	233	E9	11101001
42	2A	00101010	106	6A	01101010	170	AA	10101010	234	EA	11101010
43	2B	00101011	107	6B	01101011	171	AB	10101011	235	EB	11101011
44	2C	00101100	108	6C	01101100	172	AC	10101100	236	EC	11101100
45	2D	00101101	109	6D	01101101	173	AD	10101101	237	ED	11101101
46	2E	00101110	110	6E	01101110	174	AE	10101110	238	EE	11101110
47	2F	00101111	111	6F	01101111	175	AF	10101111	239	EF	11101111
48	30	00110000	112	70	01110000	176	B0	10110000	240	F0	11110000
49	31	00110001	113	71	01110001	177	B1	10110001	241	F1	11110001
50	32	00110010	114	72	01110010	178	B2	10110010	242	F2	11110010
51	33	00110011	115	73	01110011	179	B3	10110011	243	F3	11110011
52	34	00110100	116	74	01110100	180	B4	10110100	244	F4	11110100
53	35	00110101	117	75	01110101	181	B5	10110101	245	F5	11110101
54	36	00110110	118	76	01110110	182	B6	10110110	246	F6	11110110
55	37	00110111	119	77	01110111	183	B7	10110111	247	F7	11110111
56	38	00111000	120	78	01111000	184	B8	10111000	248	F8	11111000
57	39	00111001	121	79	01111001	185	B9	10111001	249	F9	11111001
58	3A	00111010	122	7A	01111010	186	BA	10111010	250	FA	11111010
59	3B	00111011	123	7B	01111011	187	BB	10111011	251	FB	11111011
60	3C	00111100	124	7C	01111100	188	BC	10111100	252	FC	11111100
61	3D	00111101	125	7D	01111101	189	BD	10111101	253	FD	11111101
62	3E	00111110	126	7E	01111110	190	BE	10111110	254	FE	11111110
63	3F	00111111	127	7F	01111111	191	BF	10111111	255	FF	11111111

General Module Information

General module types

CODE		
DEC	HEX	MODULE
0	0	DAQP-V
1	1	DAQP-POT
2	2	DAQP-BRIDGE
3	3	DAQN-RTD
4	4	DAQP-DMM
5	5	DAQP-CHARGE
6	6	DAQP-µV
7	7	DAQP-TRQ
8	8	DAQP-FREQ
9	9	DAQP-ACC
10	A	
11	B	
15	F	
16	10	PAD-TH8-P
17	11	PAD-V8-P
18	12	
19	13	
20	14	
21	15	
22	16	DAQ-AAF
23	17	DAQP-CHARGE-A
24	18	DAQP-BRIDGE-A
25	19	DAQN-OHM
26	1A	DAQP-FREQ-A
27	1B	DAQP-ACC-A
28	1C	
29	1D	CAL-SCANN
30	1E	DAQP-CHARGE-B
31	1F	DAQP-BRIDGE-B
32	20	CAL-BRIDGE
33	21	MDAQ-V*
34	22	DAQP-V-A
35	23	DAQP-V-B
36	24	MDAQ-BASE-x
37	25	MDAQ-SUB-ACC
38	26	MDAQ-SUB-ACC-A
39	27	MDAQ-SUB-V-200-BNC
40	28	MDAQ-SUB-BRIDGE
41	29	MDAQ-SUB-V200-D
42	2A	DAQP-BRIDGE-S
43	2B	DAQP-HV
44	2C	DAQP-CFB
45	2D	MDAQ-STG
46	2E	MDAQ-BASE-xA
47	2F	DAQP-LV
48	30	PQL-BASE
49	31	PQL-HV
50	32	PQL-LV
51	33	PQL-CURR
52	34	DAQP-LA-SC
53	35	DAQP-STG
54	36	MDAQ-DIFF-OUT
55	37	DAQP-THERM
56	38	DAQP-MULTI

Button lock/unlock status

Code	Function
0	Buttons unlock (range and filter button active)
1	Buttons lock (range and filter button not active)

Notes

Programming commands DAQP-DMM

Instruction set

Command	Syntax
Set module address	##(NewAddr)SETD\r
Read module configuration	??(Addr)\r
Set module configuration	##(Addr)(Range)(Filter)(Button Lock)\r
Set buttons locked/unlocked	##(Addr)Rx\r
Activate Power on default [(1)]	##(Addr)P(Range)(Filter)(Button Lock)\r
Deactivate Power on default [(1)]	##(Addr)P\r
Read serial number [(1)]	##(Addr)SETB\r

[(1)] Commands applied in 08 - 2003

Commands in detail

Set module address

This command has to be sent while the filter button gets pressed on the module. After that, the new address is stored in an EEPROM and the module sends a response string to the system.

Command: ##(NewAddr)SETD\r

##:	Command leading code
NewAddr:	New module address (2 characters hex from 0x00 to 0xFE)
SETD:	Set address command
\r:	Carriage return (0x0D)

Response: !(NewAddr)(ModuleType)(Range)(Filter)(Button Lock)\r

!:	Response leading code
NewAddr:	Confirmed new module address (2 characters hex)
ModuleType:	Type of module (2 characters hex, 0x04 = DAQP-DMM)
Range:	Measuring range position (2 characters hex, according table)
Filter:	Filter position (2 characters hex, according table)
Button Lock:	Lock/Unlock module (1 character, according general prog. info)

Example: Command: ##01SETD\r
Response: !010401020\r
(Address 0x01, DAQP-DMM, 400 V, 1 kHz, buttons unlocked)

Range and filter selection table

Range	Code	Filter	Code
1000 V	0x00	20 kHz	0x00
400 V	0x01	3 kHz	0x01
200 V	0x02	1 kHz	0x02
100 V	0x03	100 Hz	0x03
40 V	0x04	10 Hz	0x04
10 V	0x05	-	-

Read module configuration

Command:	??(Addr)\r	
	??:	Command leading code
	Addr:	Module address (2 characters hex from 0x00 to 0xFE)
	\r:	Carriage return (0x0D)
Response:	!(Addr)(ModuleType)(Range)(Filter)(Button Lock)\r	
		Response details see also ‘Set module address’
Example:	Command:	??01\r
	Response:	!010401020\r
		(Address 0x01, DAQP-DMM, 400 V, 1 kHz, buttons unlocked)

Set module configuration

Command:	##(Addr)(Range)(Filter)(Button Lock)\r	
		Command details see also ‘Set module address’
Response:	!ACK\r	(Notice: an incorrect command gets no response from module!)
	!:	Response leading code
	ACK:	Acknowledge
	\r:	Carriage return (0x0D)
Example:	Command:	##0101020\r
		(Address 0x01, 400 V, 1 kHz, buttons unlocked)
	Response:	!ACK\r

Lock/Unlock module buttons

Command:	##(Addr)Rx\r	
	##:	Command leading code
	Addr:	Module address (2 characters hex from 0x00 to 0xFE)
		Address FF sets all modules with one command
	Rx:	Module buttons lock/unlock command
	R0:	buttons unlocked
	R1:	buttons locked
	\r:	Carriage return (0x0D)
Response:	no response	
Example:	Command:	##01R0\r
		(Module address 0x01, buttons unlocked)
	Command:	##01R1\r
		(Module address 0x01, buttons locked)
	Command:	##FFR1\r
		(All modules, buttons locked)

Activate power on default

Not available for modules produced before 08 - 2003

Command:	##(Addr)P(Range)(Filter)(Button Lock)\r	
		Command details see also ‘Set module address’
Response:	!ACK\r	Valid command
Example:	Command:	##01P01020\r
	Response:	!ACK\r

Deactivate power on default

Not available for modules produced before 08 - 2003

Deactivates the power on default mode.
The last pushbutton selected range and filter is adjusted at power on time.

Command:	##(Addr)P\r	
	##:	Command leading code
	Addr:	Acknowledged new module address
	P:	Command
	\r:	Carriage return (0x0D)
Response:	!ACK\r	Valid command
	?(Addr)\r	Invalid command, syntax error
Example:	Command:	##01P\r
	Response:	!ACK\r

Read serial number

Not available for modules produced before 08 - 2003

Command:	##(Addr)SETB\r	
	Command details see also ‘Set module address’	
Response:	!(Serial)xx(Revision)xxx\r	
	(Serial):	Module Serial Number (6 digits)
	x:	Space
	(Revision):	Revision Number (e.g. V200)
	\r:	Carriage return (0x0D)

DAQP-DMM Module

DAQP-DMM

Nr:	Command					Response		Function
1	##	AA	SETD		CR	!AA04RRFFk	CR	Program address
2	??	AA			CR	!AA04RRFFk	CR	Read configuration
3	##	AA		RRFFk	CR	!ACK	CR	Set configuration
4	##	AA+	R1		CR			Lock buttons
5	##	AA+	R0		CR			Unlock buttons
6	##	AA	SETB		CR	!{16*ASCII}	CR	Read serial number
7	##	AA	P	RRFFk	CR	!ACK	CR	Write power on default to module

AA	Address
AA+	Address
	FF = all modules
MM	Module type
RR	Range
FF	Filter code
k	Button lock

RR	Range
0x00	1000 V
0x01	400 V
0x02	200 V
0x03	100 V
0x04	40 V
0x05	10 V

FF	Filter
0x00	20 kHz
0x01	3 kHz
0x02	1 kHz
0x03	100 Hz
0x04	10 Hz

Notes

Programming commands DAQP-HV

Instruction Set

Command	Syntax
Set module address	##(NewAddr)SETD\r
Read module configuration	??(Addr)\r
Set module configuration	##(Addr)(Range)(Filter)(Filter Type)(Button Lock)\r
Lock/Unlock module buttons	##(Addr)Rx\r
Read serial number	##(Addr)SETB\r
Activate Power on default	##(Addr)P(Range)(Filter)(Filter type)(Button Lock)\r
Deactivate Power on default	##(Addr)P\r

Commands in detail

Set module address

Command: ##(NewAddr)SETD\r

##:	Command leading code
NewAddr:	New module address (2 characters hex from 0x00 to 0xFE)
SETD:	Set address command
\r:	Carriage return (0x0D)

Response: !(NewAddr)(ModuleType)(Range)(Filter)(FilterType)(Button Lock)(Special)\r

!:	Response leading code
NewAddr:	Confirmed new module address (2 characters hex)
ModuleType:	Type of module (2 characters hex, 2B = DAQ-DMA Module)
Range:	Measuring range position (2 characters hex, according table)
Filter:	Filter position (2 characters hex, according table)
Filter type:	0: Bessel filter characteristic 1: Butterworth filter characteristic
Button Lock:	Lock/Unlock module (1 character, according general prog. info)
Special:	0: Standard module 1: Special module
\r:	Carriage return (0x0D)

Warning: *All other values will cause a refusal of the commands! Only ASCII characters between 20 and 7F Hex are allowed. No binary data will be sent! From 0x30 and 0x3F Hex.*

Example: Command: ##01SETD\r

Response: !012B0102000\r
(Address 0x01, DAQ-DMA Module, 800 V, 30 kHz, Bessel Filter, buttons unlocked, Standard Module)

This command has to be sent as long as the user pushes one of the two buttons on the front of the module. Afterwards, the assigned module address will be written in a permanent storage and the answer string is sent back to the host for confirmation. The module waits until the user unhands the button.

Range and filter selection table

Range	Code	Filter	Code
1400 V	0x00	180 kHz	0x00
800 V	0x01	100 kHz	0x01
400 V	0x02	30 kHz	0x02
200 V	0x03	10 kHz	0x03
100 V	0x04	3 kHz	0x04
50 V	0x05	1 kHz	0x05
20 V	0x06	300 Hz	0x06
-	-	100 Hz	0x07
-	-	10 Hz	0x08
-	-	3 Hz	0x09

Read Module configuration

Command: ??(Addr)\r

??: Command leading code
Addr: Module Address (2 characters hex from 0x00 to 0xFE)
\r: Carriage return

Response: !(Addr)(Module type)(Range)(Filter)(Filter Type)(Button Lock)(Special)\r

!: Response leading code
Addr: Module address (2 characters hex)
ModuleType: Type of module (2 characters hex, 2B = DAQ-DMA Module)
Range: Measuring range position (2 characters hex, according table)
Filter: Filter position (2 characters hex, according table)
Filter type: 0: Bessel filter characteristic
1: Butterworth filter characteristic
Button Lock: Lock/Unlock module (1 character, according general prog. info)
Special: 0: Standard module
1: Special module
\r: Carriage return (0x0D)

Example: Command: ##01SETD\r

Response: !012B0102000\r
(Address 0x01, DAQ-DMA Module, 800 V, 30 kHz, Bessel Filter, buttons unlocked, Standard Module)

Warning: All other values will cause a refusal of the commands! Only ASCII characters between 0x20 and 0x7F Hex are allowed. No binary data will be sent!

Set module configuration

Command: ##(Addr)(Range)(Filter)(Filter type)(Button Lock)\r

##: Command leading code
Addr: Module Address (2 characters hex from 0x00 to 0xFE)
Range: Measuring range position (2 characters hex, according table)
Filter: Filter position (2 characters hex, according table)
Filter type: 0: Bessel filter characteristic
1: Butterworth filter characteristic
Button Lock: Lock/Unlock module (1 character, according general prog. info)

Response: !ACK\r (Notice: an incorrect command gets no response from module!)

!: Response leading code
ACK: Acknowledge
\r: Carriage return (0x0D)

Warning: *All other values will cause a refusal of the commands! Only ASCII characters between 0x20 and 0x7F Hex are allowed. No binary data will be sent!*

Example: Command: ##01010200\r
(Address 0x01, 800 V, 30 kHz, Bessel filter, buttons unlocked)
Response: !ACK\r

Lock/Unlock module buttons

Command: ##(Addr)Rx\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE) Address FF sets all modules with one command
Rx:	Module buttons lock/unlock command R0: buttons unlocked R1: buttons locked
\r:	Carriage return (0x0D)

Response: no response

Example: Command: ##01R0\r
(Module address 0x01, buttons unlocked)
Command: ##01R1\r
(Module address 0x01, buttons locked)

Read serial number

Command: ##(Addr)SETB\r

##:	Command leading code
Addr:	Module address (2 characters hex)
SETB:	Command
\r:	Carriage return (0x0D)

Response: !xxxxxxxxxxxxxxxx\r

!:	Response leading code
x:	16 characters long ASCII string
\r:	Carriage return (0x0D)

Activate Power-on default

Command: ##aaPBBFFBR\r

##:	Command leading code
Addr:	Module Address (2 characters hex from 0x00 to 0xFE)
P:	Command
Range:	Measuring range position (2 characters hex, according table)
Filter:	Filter position (2 characters hex, according table)
Filter type:	0: Bessel filter characteristic 1: Butterworth filter characteristic
Button Lock:	Lock/Unlock module (1 character, according general prog. info)

Response: !ACK\r (Notice: an incorrect command gets no response from module!)

!:	Response leading code
ACK:	Acknowledge
\r:	Carriage return (0x0D)

Example: Command: ##01P01020\r
Response: !ACK\r

Deactivate Power-on default

Deactivates the power on default mode.
The last pushbutton selected range and filter is adjusted at power on time.

Command:	##(Addr)P\r	
	##:	Command leading code
	Addr:	Acknowledged new module address
	P:	Command
	\r:	Carriage return (0x0D)
Response:	!ACK\r	Valid command
Example:	Command:	##01P\r
	Response:	!ACK\r

Notes

DAQP-HV

Nr:	Command					Response		Function
1	##	AA	SETD		CR	!AAMMBBFFBRZ\r	CR	Program address
2	??	AA			CR	!AAMMBBFFBRZ\r	CR	Read configuration
3	##	AA		BBFFBR	CR	!ACK	CR	Set configuration
4	##	AA+	R1		CR			Lock buttons
5	##	AA+	R0		CR			Unlock buttons
6	##	AA	SETB		CR	!{16*ASCII}	CR	Read serial number
7	##	AA	P	PBBFFBR	CR	!ACK	CR	Write power on default to module

AA	Address
AA+	Address FF = all modules
MM	Module type
BB	Range
FF	Filter code
R	Button lock/unlock

RR	Range
0x00	1400 V
0x01	800 V
0x02	400 V
0x03	200 V
0x04	100 V
0x05	50 V
0x06	20 V

FF	Filter
0x00	180 kHz
0x01	100 kHz
0x02	30 kHz
0x03	10 Hz
0x04	3 Hz
0x05	1 Hz
0x06	300 Hz
0x07	100 Hz
0x08	30 Hz
0x09	10 Hz

Programming commands DAQP-V

Instruction set

Command	Syntax
Set module address	##(NewAddr)SETD\r
Read module configuration	??(Addr)\r
Set module configuration	##(Addr)(Range)(Filter)(Button Lock)\r
Lock/Unlock module buttons	##(Addr)Rx\r
Activate Power on default (1)	##(Addr)P(Range)(Filter)(Button Lock)\r
Deactivate Power on default (1)	##(Addr)P\r
Read serial number (1)	##(Addr)SETB\r
(1) Commands applied in 08 - 2003	

Commands in detail

Set module address

This command has to be sent while the filter button gets pressed on the module. After that, the new address is stored in an EEPROM and the module sends a response string to the system.

Command: ##(NewAddr)SETD\r

##:	Command leading code
NewAddr:	New module address (2 characters hex from 0x00 to 0xFE)
SETD:	Set address command
\r:	Carriage return (0x0D)

Response: !(NewAddr)(ModuleType)(Range)(Filter)(Button Lock)\r

!:	Response leading code
NewAddr:	Confirmed new module address (2 characters hex)
ModuleType:	Type of module (2 characters hex, 00 hex for DAQP-V according general prog. info)
Range:	Measuring range position (2 characters hex, according table)
Filter:	Filter position (2 characters hex, according table)
Button lock:	Lock/Unlock module (1 character, according general prog. info)

Example:
Command: ##01SETD\r
Response: !010001020\r
(Address 0x01, DAQP-V, 10 V, 1 kHz, buttons unlocked)

Range and filter selection table

Range	Code	Filter	Code
50 V	0x00	50 kHz	0x00
10 V	0x01	10 kHz	0x01
5 V	0x02	1 kHz	0x02
1V	0x03	100 Hz	0x03
0,1 V	0x04	10 Hz	0x04
0,01 V	0x05	-	-

DAQP-V Module

Read module configuration

Command: ??(Addr)\r

??: Command leading code
Addr: Module address (2 characters hex from 0x00 to 0xFE)
\r: Carriage return (0x0D)

Response: !(Addr)(ModuleType)(Range)(Filter)(Button Lock)\r
Response details see also ‘Set module address’

Example: Command: ??01\r
Response: !010001020\r
(Address 0x01, DAQP-V, 10 V, 1 kHz, buttons unlocked)

Set module configuration

Command: ##(Addr)(Range)(Filter)(Button Lock)\r

Command details see also ‘Set module address’

Response: !ACK\r (Notice: an incorrect command gets no response from module!)

!: Response leading code
ACK: Acknowledge
\r: Carriage return (0x0D)

Example: Command: ##0101020\r
(Address 0x01, 10 V, 1 kHz, buttons unlocked)
Response: !ACK\r

Lock/Unlock module buttons

Command: ##(Addr)Rx\r

##: Command leading code
Addr: Module address (2 characters hex from 0x00 to 0xFE)
Address FF sets all modules with one command
Rx: Module buttons lock/unlock command
R0: buttons unlocked
R1: buttons locked
\r: Carriage return (0x0D)

Response: no response

Example: Command: ##01R0\r
(Module address 0x01, buttons unlocked)
Command: ##01R1\r
(Module address 0x01, buttons locked)

Activate power on default

Not available for modules produced before 08 - 2003

Command: ##(Addr)P(Range)(Filter)(Button Lock)\r

Command details see also ‘Set module address’

Response: !(Addr)\r Valid command
?(Addr)\r Invalid command, syntax error

Example: Command: ##01P01020\r
Response: !ACK\r

Deactivate power on default

Not available for modules produced before 08 - 2003

Deactivates the power on default mode.
The last pushbutton selected range and filter is adjusted at power on time.

Command:	##(Addr)P\r	
	##:	command leading code
	Addr:	Acknowledged new module address
	P:	Command
	\r:	Carriage return (0x0D)
Response:	!ACK\r	Valid command
Example:	Command:	##01P\r
	Response:	!ACK\r

Read serial number

Not available for modules produced before 08 - 2003

Command:	##(Addr)SETB\r	
	Command details see also ‘Set module address’	
Response:	!(Serial)xx(Revision)xxx\r	
	(Serial):	Module Serial Number (6 digits)
	x:	space
	(Revision):	Revision Number (e.g. V200)
	\r:	Carriage return (0x0D)

DAQP-V

Nr:	Command					Response		Function
1	##	AA	SETD		CR	!AA00RRFFk	CR	Program address
2	??	AA			CR	!AA00RRFFk	CR	Read configuration
3	##	AA		RRFFk	CR	!ACK	CR	Set configuration
4	##	AA+	R1		CR			Lock buttons
5	##	AA+	R0		CR			Unlock buttons
6	##	AA	SETB		CR	!{16*ASCII}	CR	Read serial number
7	##	AA	P	RRFFk	CR	!ACK	CR	Write power on default to module

AA	Address
AA+	Address FF = all modules
MM	Module type
RR	Range
FF	Filter code
k	Button lock

RR	Range
0x00	50 V
0x01	10 V
0x02	5 V
0x03	1 V
0x04	0.1 V
0x05	0.01 V

FF	Filter
0x00	50 kHz
0x01	10 kHz
0x02	1 kHz
0x03	100 Hz
0x04	10 Hz

Programming commands DAQP-V-A, -B and -LV

Instruction set

Command	Syntax
Set module address	##(NewAddr)SETD\r
Read module configuration	??(Addr)\r
Set module configuration	##(Addr)(Range)(Filter)(UniBi)(Coupling)(FilterType)(Remote)\r
Set remote control	##(Addr)R(Remote)\r
Activate Power On Default	##(Addr)SETP(Range)(Filter)(UniBipolar)(Coupling)(FilterType)(Remote)\r
Deactivate PON	##(Addr)SETP\r
Read Serial Number	##(Addr)SETB\r
TEDS programming commands[1)]	-
[1)] please refere to chapter: ‚TEDS programming commands' in the ‚General module overview'.	

Range and filter selection table

DAQP-V-A

Range	Code	Filter	Code
50 V	0x00	300 kHz	0x00
25 V	0x01	100 kHz	0x01
10 V	0x02	30 kHz	0x02
5 V	0x03	10 kHz	0x03
2 V	0x04	3 kHz	0x04
1 V high	0x05	1 kHz	0x05
1 V	0x06	300Hz	0x06
500 mV	0x07	100Hz	0x07
200 mV	0x08	30Hz	0x08
100 mV	0x09	10Hz	0x09
40 mV	0x0A	-	-
20 mV	0x0B	-	-

DAQP-V-B & DAQP-LV

Range	Code	Filter	Code
50 V	0x00	180 kHz*	0x00
25 V	0x01	100 kHz	0x01
10 V	0x02	30 kHz	0x02
5 V	0x03	10 kHz	0x03
2,5 V	0x04	3 kHz	0x04
1 V	0x05	1 kHz	0x05
500 mV	0x06	300Hz	0x06
200 mV	0x07	100Hz	0x07
100 mV	0x08	30Hz	0x08
50	0x09	10Hz	0x09
20 mV	0x0A	-	-
10 mV	0x0B	-	-

*) DAQP-LV: 300 kHz

Commands in detail

Set module address

This command has to be send while the filter button gets pressed on the module. After that, the new address is stored in an EEPROM, and the module sends a response string to the system.

Command: ##(NewAddr)SETD\r

##:	Command leading code
NewAddr:	New module address (2 characters hex)
SETD:	Command
\r:	Carriage return (0x0D)

Response: !(NewAddr)(ModuleType)(Range)(Filter)(UniBipolar)(Coupling)(FilterType)(Remote)(Overflow)(Special)\r

!:	Response leading code	
NewAddr:	Module address	
ModuleType:	22 DAQP-V-A; 23 DAQP-V-B; 2F DAQP-LV;	
Range:	Amplifier Range (2 characters hex, according to table)	
Filter:	Low Pass Filter (2 characters hex, according to table)	
UniBipolar:	0:	Bipolar Mode (for example ±5 V)
	1:	Unipolar Mode (for example +5 V)
Coupling :	0:	DC Coupling
	1:	AC Coupling
	2,3:	reserved
FilterType:	0:	Bessel filter characteristic
	1:	Butterworth filter characteristic
Remote :	Remote or local access to the module (1 character, according to general prog info.)	
Overflow:	0:	Module Okay
	1:	Module exceeds Common mode Range. (DAQP-V-A only)
	The overflow Flag is cleared after one readout.	
Special:	0:	Standard module
	1:	Special module

Example: Command: ##04SETD\r

Response: !04220201000010\r

Module address 04; DAQP-V-A; 10 V Range; 100 kHz LP; Bipolar; DC-Coupled; Bessel characteri stic; Local Mode; the module has been out of the common mode range; Standard Module;

Read module Configuration

Command: ??(Addr)\r

??: Command leading code
Addr: Module address (2 characters hex)
\r: Carriage return (0x0D)

Response: !(Addr)(ModuleType)(Range)(Filter)(UniBipolar)(Coupling)(FilterType)(Remote)(reserved)(Special)\r

!: Response leading code
Addr: Module address
ModuleType: 22 DAQP-V-A; 23 DAQP-V-B; 2F DAQP-LV
Range: Amplifier Range (2 characters hex, according to table)
Filter: Low Pas Filter (2 characters hex, according to table)
UniBipolar: 0: Bipolar Mode (for example ±5 V)
1: Unipolar Mode (for example +5 V)
Coupling: 0: DC Coupling
1: AC Coupling
2,3: reserved
FilterType: 0: Bessel filter characteristic
1: Butterworth filter characteristic
Remote: Remote or local access to the module (1 character, according to general prog info.)
Reserved: 0
Special: 0: Standard module
1: Special module

Example: Command: ??04\r

Response: !04220201000000\r

Module address 0x04; DAQP-V-A; 10V Range; 100 kHz LP; Bipolar; DC-Coupled; Besel characteristic; Local mode; reserved; Standard Module;

Set module Configuration

Command: ##(Addr)(Range)(Filter)(UniBipolar)(Coupling)(FilterType)(Remote)\r

##: Command leading code
Addr: Module address (2 characters hex)
Range: Amplifier Range (2 characters hex, according to table)
Filter: Low Pas Filter (2 characters hex, according to table)
UniBipolar: 0: Bipolar Mode (for example ±5 V)
1: Unipolar Mode (for example +5 V)
Coupling: 0: DC Coupling
1: AC Coupling
2,3: reserved
FilterType: 0: Bessel filter caracteristic
1: Butterworth filter caracteristic
Remote: Remote or local access to the module (1 character, according to general prog info.)
\r: Carriage return (0x0D)

Response: !ACK\r

Example: Command: ##0405021011\r

Module address 0x04; 2V Range; 1 kHz LP; Unipolar; DC-Coupled; Butterworth characteristic; Remote mode;

Response: !ACK\r

Set Remote Control

Command: ##(Addr)R(remote)\r

##:	Command leading code	
Addr:	Module address (2 characters hex; address FF takes effect on all modules)	
R:	Command	
Remote:	0:	Local mode
	1:	Remote Mode (Front Panel Buttons locked)
\r:	Carriage return (0x0D)	

Response: !ACK\r

Example: Command: ##0405021011\r

Module address 04; 2 V Range; 1 kHz LP; Unipolar; DC-Coupled; Butterworth characteristic; Remote Mode;

Response: !ACK\r

Activate Power On Default

Command: ##(Addr)SETP(Range)(Filter)(UniBipolar)(Coupling)(FilterType)(Remote)\r

##:	Command leading code	
Addr:	Module address (2 characters hex)	
SETP:	Command	
Range:	Amplifier Range (2 characters hex, according to table)	
Filter:	Low Pas Filter (2 characters hex, according to table)	
UniBipolar:	0:	Bipolar Mode (for example ±5 V)
	1:	Unipolar Mode (for example +5 V)
Coupling:	0:	DC Coupling
	1:	AC Coupling
	2,3:	reserved
FilterType:	0:	Bessel filter caracteristic
	1:	Butterworth filter caracteristic
Remote:	Remote or local access to the module (1 character, according to general prog info.)	
\r:	Carriage return (0x0D)	

Response: !ACK\r

Example: Command: ##04SETP05021011\r

Module address 0x04; 2V Range; 1 kHz LP; Unipolar; DC-Coupled; Butterworth characteristic; Remote Mode

Response: !ACK\r

Deactivate Power On Default

Command: ##(Addr)SETP\r

##:	Command leading code
Addr:	Module address (2 characters hex)
SETP:	Command
\r:	Carriage return (0x0D)

Response: !ACK\r

Read Serial Number

Command: ##(Addr)GETN\r

##:	Command leading code
Addr:	Module address (2 characters hex)
GETN:	Command
\r:	Carriage return (0x0D)

Response: !(SERAL)xx(REVISION)xxx\r

(SERIAL):	Module Serial Number (6 digits)
xx:	2 characters space
(Revision):	Revision Number V100
xxx:	3characters space
\r:	Carriage return (0x0D)

Notes

Programming commands DAQP-µV

Instruction set

Command	Syntax
Set module address	##(NewAddr)SETD\r
Read module configuration	??(Addr)\r
Set module configuration	##(Addr)(Range)(Filter)(Button Lock)\r
Lock/Unlock module buttons	##(Addr)Rx\r

Commands in detail

Set module address

This command has to be sent as long as the filter button has been pressed on the module. After that, the new address is stored in an EEPROM and the module sends a response string to the system.

Command: ##(NewAddr)SETD\r

##:	Command leading code
NewAddr:	New module address (2 characters hex from 0x00 to 0xFE)
SETD:	Set address command
\r:	Carriage return (0x0D)

Response: !(NewAddr)(ModuleType)(Range)(Filter)(Button Lock)\r

!:	Response leading code
NewAddr:	Confirmed new module address (2 characters hex)
ModuleType:	Type of module (2 characters hex, 06 hex according general prog. info)
Range:	Measuring range position (2 characters hex, according table)
Filter:	Filter position (2 characters hex, according table)
Button lock:	Lock/Unlock module (1 character, according generalprog. info)

Example: Command: ##01SETD\r

Response: !010605020\r
(Address 0x01, DAQP-µV, 500 µV, 1 kHz, buttons unlocked)

Range and filter selection table

Range	Code	Filter	Code
25 mV	0x00	20 kHz	0x00
10 mV	0x01	5 kHz	0x01
5 mV	0x02	1 kHz	0x02
2.5 mV	0x03	100 Hz	0x03
1 mV	0x04	10 Hz	0x04
500 µV	0x05	-	-

DAQP-µV Module

Read module configuration

Command: ??(Addr)\r
- ??: Command leading code
- Addr: Module address (2 characters hex from 0x00 to 0xFE)
- \r: Carriage return (0x0D)

Response: !(NewAddr)(ModuleType)(Range)(Filter)(Button Lock)\r
Response details see also ‘Set module address’

Example: Command: ??01\r

Response: !010605020\r
(Address 0x01, DAQP-µV, 500 µV, 1 kHz, buttons unlocked)

Set module configuration

Command: ##(Addr)(Range)(Filter)(Button Lock)\r
Command details see also ‘Set module address’

Response: !ACK\r (Notice: an incorrect command gets no response from module!)
- !: Response leading code
- ACK: Acknowledge
- \r: Carriage return (0x0D)

Example: Command: ##0105020\r
(Address 0x01, 500 µV, 1 kHz, buttons unlocked)

Response: !ACK\r

Lock/Unlock module buttons

Command: ##(Addr)Rx\r
- ##: Command leading code
- Addr: Module address (2 characters hex from 0x00 to 0xFE)
  Address FF sets all modules with one command
- Rx: Module buttons lock/unlock command
  - 0: buttons unlocked
  - 1: buttons locked
- \r: Carriage return (0x0D)

Response: no response

Example: Command: ##01R0\r
(Module address 0x01, buttons unlocked)

Command: ##01R1\r
(Module address 0x01, buttons locked)

Notes

DAQP-µV

Nr:	Command					Response		Function
1	##	AA	SETD		CR	!AA06RRFFk	CR	Program address
2	??	AA			CR	!AA06RRFFk	CR	Read configuration
3	##	AA		RRFFk	CR	!ACK	CR	Set configuration
4	##	AA+	R1		CR			Lock buttons
5	##	AA+	R0		CR			Unlock buttons

AA	Address
AA+	Address
	FF = all modules
MM	Module type
RR	Range
FF	Filter code
k	Button lock

RR	Range
0x00	25 mV
0x01	10 mV
0x02	5 mV
0x03	2.5 mV
0x04	1 mV
0x05	500 µV

FF	Filter
0x00	20 kHz
0x01	5 kHz
0x02	1 kHz
0x03	100 Hz
0x04	10 Hz

Programming commands DAQP-LA

Instruction set

Command	Syntax
Set module address	##(NewAddr)SETD\r
Read module configuration	??(Addr)\r
Set module configuration	##(Addr)(Range)(Filter)(FilterType)(Remote)\r
Set remote control	##(Addr)Rx\r
Activate Power On Default	##(Addr)SETP(Range)(Filter)(FilterType)(Remote)\r
Deactivate Power On Default	##(Addr)SETP\r
Read Serial Number	##(Addr)SETB\r

Range and filter selection table

Range DAQP-LA-SC	Code	Range DAQP-LA-B-S1		Filter	Code
30 A	0x00	600 mA	0x00	300 kHz	0x00
10 A	0x01	200 mA	0x01	100 kHz	0x01
3 A	0x02	60 mA	0x02	30 kHz	0x02
1 A	0x03	20 mA	0x03	10 kHz	0x03
0.3 A	0x04	6 mA	0x04	3 kHz	0x04
0.1 A	0x05	2 mA	0x05	1 kHz	0x05
-	0x06		0x06	300Hz	0x06
-	0x07		0x07	100Hz	0x07
-	0x08		0x08	30Hz	0x08
-	0x09		0x09	10Hz	0x09

Commands in detail

Set module address

This command has to be send while the filter button gets pressed on the module. After that, the new address is stored in an EEPROM, and the module sends a response string to the system.

Command: ##(NewAddr)SETD\r

##:	Command leading code	
NewAddr:	New module address (2 characters hex)	
SETD:	Command	
\r:	Carriage return (0x0D)	

Response: !(NewAddr)(ModuleType)(Range)(Filter)(FilterType)(Remote)(Special)\r

!:	Response leading code	
NewAddr:	Module address	
ModuleType:	34 DAQP-LV	
Range:	Amplifier Range (2 characters hex, according to table)	
Filter:	Low Pass Filter (2 characters hex, according to table)	
FilterType:	0:	Bessel filter characteristic
	1:	Butterworth filter characteristic
Remote :	Remote or local access to the module (1 character, according to general prog info.)	
Special:	0:	Standard module
	1:	Special module

Example: Command: ##04SETD\r

Response: !04340201001\r

(Module address 0x04; DAQP-LA; 3000/60 mA Range; 100 kHz LP; Buterworth characteristic; Local Mode; Special Module)

Read module configuration

Command: ??(Addr)\r

??:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
\r:	Carriage return (0x0D)

Response: !(Addr)(ModuleType)(Range)(Filter)(FilterType)(Remote)(Special)\r
Response details see also ‘Set module address’

Example: Command: ??01\r

Response: !01340201001\r

(Address 0x01, DAQP-LA; 3000/60 mA Range; 100 kHz LP; Buterworth characteristic; Local Mode; Special Module)

Set module Configuration

Command: ##(Addr)(Range)(Filter)(FilterType)(Remote)\r

##:	Command leading code	
Addr:	Module address (2 characters hex)	
Range:	Amplifier Range (2 characters hex, according to table)	
Filter:	Low Pas Filter (2 characters hex, according to table)	
FilterType:	0:	Bessel filter caracteristic
	1:	Butterworth filter caracteristic
Remote:	Remote or local access to the module (1 character, according to general prog info.)	
\r:	Carriage return (0x0D)	

Response: !ACK\r

Example: Command: ##04050201\r

Module address 0x04; 100/2 mV Range; 30 kHz LP; Butterworth characteristic; Remote mode;

Response: !ACK\r

Set Remote Control

Command: ##(Addr)R(remote)\r

##:	Command leading code	
Addr:	Module address (2 characters hex; address FF takes effect on all modules)	
R:	Command	
Remote:	0:	Local mode
	1:	Remote Mode (Front Panel Buttons locked)
\r:	Carriage return (0x0D)	

Response: !ACK\r

Example: Command: ##04050201\r

Module address 0x04; 100/2 mA Range; 30 kHz LP; Butterworth characteristic; Remote mode;

Response: !ACK\r

Activate Power On Default

Command: ##(Addr)SETP(Range)(Filter)(FilterType)(Remote)\r

##:	Command leading code	
Addr:	Module address (2 characters hex)	
SETP:	Command	
Range:	Amplifier Range (2 characters hex, according to table)	
Filter:	Low Pas Filter (2 characters hex, according to table)	
FilterType:	0:	Bessel filter caracteristic
	1:	Butterworth filter caracteristic
Remote:	Remote or local access to the module (1 character, according to general prog info.)	
\r:	Carriage return (0x0D)	

Response: !ACK\r

Example: Command: ##04SETP050201\r

Module address 0x04; 100/2 mA Range; 30 kHz LP; Butterworth characteristic; Remote mode;

Response: !ACK\r

Deactivate Power On Default

Command: ##(Addr)SETP\r

##:	Command leading code
Addr:	Module address (2 characters hex)
SETP:	Command
\r:	Carriage return (0x0D)

Response: !ACK\r

Read serial number

Command: ##(Addr)SETB\r

Command details see also ‘Set module address’

Response: !(Serial)xx(Revision)xxx\r

(Serial):	Module Serial Number (6 digits)
x:	space
(Revision):	Revision Number (e.g. V200)
\r:	Carriage return (0x0D)

Programming commands DAQP-BRIDGE

Instruction set

Command	Syntax
Set module address	##(NewAddr)SETD\r
Read module configuration	??(Addr)\r
Set module configuration	##(Addr)(Range)(Filter)(Button Lock)\r
Lock/Unlock module buttons	##(Addr)Rx\r
Read serial [1]	##(Addr)SETB\r

[1] Commands applied in 08-2003

Commands in detail

Set module address

This command has to be sent as long as the filter button has been pressed on the module. After that, the new address is stored in an EEPROM and the module sends a response string to the system.

Command: ##(NewAddr)SETD\r

##:	Command leading code
NewAddr:	New module address (2 characters hex from 0x00 to 0xFE)
SETD:	Set address command
\r:	Carriage return (0x0D)

Response: !(NewAddr)(ModuleType)(Range)(Filter)(Button Lock)\r

!:	Response leading code
NewAddr:	Confirmed new module address (2 characters hex)
ModuleType:	Type of module (2 characters, 02 hex according general prog. info)
Range:	Current measuring range position (2 characters hex, according table)
Filter:	Current filter position (2 characters hex, according table)
Button lock:	Lock/Unlock module (1 character, according general prog. info)

Example: Command: ##01SETD\r

Response: !010204010\r
(Address 0x01, DAQP-BRIDGE, 1 mV/V, 5 kHz, buttons unlocked)

Range and filter selection table

Range [1]	Code	Filter	Code
5 mV/V	0x00	20 kHz	0x00
2 mV/V	0x01	5 kHz	0x01
1 mV/V	0x02	1 kHz	0x02
0.5 mV/V	0x03	100 Hz	0x03
0.2 mV/V	0x04	10 Hz	0x04
0.1 mV/V	0x05	-	-
[1] @ 5V Excitation		-	-

DAQP-BRIDGE Module

Read module configuration

Command: ??(Addr)\r

??: Command leading code
Addr: Module address (2 characters hex from 0x00 to 0xFE)
\r: Carriage return (0x0D)

Response: !(Addr)(ModuleType)(Range)(Filter)(Button Lock)\r
Response details see also ‘Set module address’

Example: Command: ??01\r

Response: !010204010\r
(Address 0x01, DAQP-BRIDGE, 1 mV/V, 5 kHz, buttons unlocked)

Set module configuration

Command: ##(Addr)(Range)(Filter)(Button Lock)\r

Command details see also ‘Set module address’

Response: !ACK\r (Notice: an incorrect command get no response from module!)

!: Response leading code
ACK: Acknowledge
\r: Carriage return (0x0D)

Example: Command: ##0104010\r
(Address 0x01, 1 mV/V, 5 kHz, buttons unlocked)

Response: !ACK\r

Lock/Unlock module buttons

Command: ##(Addr)Rx\r

##: Command leading code
Addr: Module address (2 characters hex from 0x00 to 0xFE)
Address FF sets all modules with one command
Rx: Module buttons lock/unlock command
R0: buttons unlocked
R1: buttons locked
\r: Carriage return (0x0D)

Response: no response

Example: Command: ##01R0\r
(Module address 0x01, buttons unlocked)

Command: ##01R1\r
(Module address 0x01, buttons locked)

Read serial number

Command: ##(Addr)SETB\r

Command details see also ‘Set module address’

Response: !(Serial)xx(Revision)xxx\r

(Serial): Module Serial Number (6 digits)
x: space
(Revision): Revision Number (i.e. V200)
\r: Carriage return (0x0D)

Notes

DAQP-BRIDGE

Nr:	Command					Response		Function
1	##	AA	SETD		CR	!AA02RRFFk	CR	Program address
2	??	AA			CR	!AA02RRFFk	CR	Read configuration
3	##	AA		RRFFk	CR	!ACK	CR	Set configuration
4	##	AA+	R1		CR			Lock buttons
5	##	AA+	R0		CR			Unlock buttons
6	##	AA	SETB		CR	!{16*ASCII}	CR	Read serial number

AA	Address
AA+	Address
	FF = all modules
MM	Module type
RR	Range
FF	Filter code
k	Button lock

RR	Range
0x00	5 mV/V
0x01	2 mV/V
0x02	1 mV/V
0x03	0.5 mV/V
0x04	0.2 mV/V
0x05	0.1 mV/V

FF	Filter
0x00	20 kHz
0x01	5 kHz
0x02	1 kHz
0x03	100 Hz
0x04	10 Hz

Programming commands DAQP-BRIDGE-A

Instruction set

Command	Syntax
Set module address	##(NewAddr)SETD\r
Read module configuration	??(Addr)\r
Set module configuration	##(Addr)(Range)(Filter)(Excitation)(ShortCircuit)(Shunt)(Mode)(FilterType)(Button Lock)\r
Automatic sensor offset adjustment	##(Addr)R7\r
Internal amplifier zero	##(Addr)R8\r
Lock/Unlock module buttons	##(Addr)R(Button Lock)\r
Input short circuit on	##(Addr)R2\r
Input short circuit off	##(Addr)R3\r
Activate shunt resistor	##(Addr)R4\r
Deactivate shunt resistor	##(Addr)R5\r
Activate Power on default	##(Addr)P(InputRange)(Filter)(Excitation)(Mode)(FilterType)(Button Lock)\r
Deactivate Power on default	##(Addr)P\r
Read serial number	##(Addr)SETB\r

Commands in detail

Set module address

This command has to be sent as long as the filter button has been pressed on the module. After that, the new address is stored in an EEPROM and the module sends a response string to the system.

Command: ##(Addr)SETD\r

##:	Command leading code	
Addr:	Module address (2 characters hex from 0x00 to 0xFE)	
\r:	Carriage return (0x0D)	

Response: !(Addr)18(InputRange)(Filter)(Excitation)(ShortCircuit)(Shunt)(Mode)(FilterType)(Button Lock)(Special)\r

!:	Response leading code	
Addr:	Acknowledged new module address	
18:	Module type (DAQP-BRIDGE-A)	
InputRange:	Measuring range (2 characters hex, according to table)	
Filter:	Low pass filter (2 characters hex, according to table)	
Excitation:	Excitation voltage (2 characters hex, according to table)	
ShortCircuit:	0	Input activated
	1	Input short circuit
Shunt:	0	Shunt deactivated
	1	Shunt activated
Mode:	Bridge type (1 character hex, according to table)	
FilterType:	0	Bessel characteristic
	1	Butterworth characteristic
Button lock:	0	Buttons unlocked
	1	Buttons locked
Special:	0	Standard module
	1	Special module

Example: Command: ##01SETD\r

Response: !0118000103001000\r

(DAQP-BRIDGE-A; 50 mV/V; 5kHz; 5V; no short circuit; no shunt; half bridge; Bessel filter; key's on; standard module)

DAQP-BRIDGE-A Module

Range and filter selection table

Range [1]	Code	Filter	Code
50 mV/V	0x00	20 kHz	0x00
20 mV/V	0x01	5 kHz	0x01
10 mV/V	0x02	1 kHz	0x02
5 mV/V	0x03	1 00Hz	0x03
2 mV/V	0x04	10 Hz	0x04
1 mV/V	0x05	-	-
[1] @ 5 V Excitation		-	-

Bridge type

Code	Mode	Shunt
0x00	Full Bridge	59.88 kOhm
0x01	Half Bridge	59.88 kOhm
0x02	Quarter Bridge 120 Ohm	-
0x03	Quarter Bridge 350 Ohm	-
0x04	Half Bridge	175 kOhm
0x05	Full Bridge	175 kOhm

Excitation

Code	Excitation
0x00	0 V
0x01	1 V
0x02	2.5 V
0x03	5 V
0x04	10 V
0x05	0.25 V [2]
0x06	0.5 V [2]
[2] Rev. 2.00 or higher	

Read module configuration

Command: ??(Addr)\r

??: Command leading code
Addr: Module address (2 characters hex from 0x00 to 0xFE)
\r: Carriage return (0x0D)

Response: !(Addr)18(InputRange)(Filter)(Excitation)(ShortCirc)(Shunt)(Mode)(FilterType)(Button Lock)(Special)\r

!: Response leading code
Addr: Module address
18: Module Type (DAQP-BRIDGE-A)
InputRange: Measuring range (2 characters hex, according to table)
Filter: Low pass filter (2 characters hex, according to table)
Excitation: Excitation Voltage (2 characters hex, according to table)
ShortCircuit: 0 Input activated
1 Input short circuit
Shunt: 0 Shunt deactivated
1 Shunt activated
Mode: Bridge type (1 character hex, according to table)
FilterType: 0 Bessel characteristic
1 Butterworth characteristic
Button lock: 0 Buttons unlocked
1 Buttons locked
Special: 0 Standard module
1 Special module

Example: Command: ??01\r

Response: !0118000103001000\r

(DAQP-BRIDGE-A; 50 mV/V; 5 kHz; 5 V; no short circuit; no shunt; half bridge; Bessel filter; key's on; standard module) Rev. 2.00 or higher

Set module configuration

Command: ##(Addr)(InputRange)(Filter)(Excitation)(ShortCircuit)(Shunt)(Mode)(FilterType)(Button Lock)\r

!: Response leading code
Addr: Module address
InputRange: Measuring range (2 characters hex, according to table)
Filter: Low pass filter (2 characters hex, according to table)
Excitation: Excitation voltage (2 characters hex, according to table)
ShortCircuit: 0 Input activated
1 Input short circuit
Shunt: 0 Shunt deactivated
1 Shunt activated
Mode: Bridge type (1 character hex, according to table)
Filter Type: 0 Bessel Characteristic
1 Butterworth Characteristic

	Button lock:	0	Buttons unlocked
		1	Buttons locked
Response:	!(Addr)\r	Valid command	
	?(Addr)\r	Invalid command, syntax error	

Automatic sensor offset adjustment

Sets the actual sensor offset to zero. The maximum adjustment range is +/-200% of the input range.

Command:	##(Addr)R7\r	
	##:	Command leading code
	Addr:	Module address (2 characters hex from 0x00 to 0xFE) Sending the address FF takes affect on all connected bridge modules.
	R7:	Command
	\r:	Carriage return (0x0D)
Response:	There is no response on this command.	
Warning:	The module is approximately 2 seconds off-line after sending this command. That means it cant receive any command.	

Internal amplifier zero

This function short circuit the module input, and measures the offset values from the different input ranges. The module automatically corrects the output voltage with this offset values.
Also the values are permanently stored in to the modules memory.

Command:	##(Addr)R8\r	
	##:	Command leading code
	Addr:	Module address (2 characters hex from 0x00 to 0xFE) Sending the address FF takes affect on all connected bridge modules.
	R8:	Command
	\r:	Carriage return (0x0D)
Response:	There is no response on this command.	
Warning:	The Module is approximately 15 seconds off-line after sending this command. That means it cant receive any command. If there is no sensor connected, activate the half bridge mode before sending this command. Otherwise you will get an output overflow.	

Lock/Unlock module buttons

Command: ##(Addr)R(Button Lock)\r

##:	Command leading code	
Addr:	Module address (2 characters hex from 0x00 to 0xFE) Sending the Address FF takes affect on all connected bridge modules.	
R:	Command	
Button lock:	0	buttons unlocked
	1	buttons locked
\r:	Carriage return (0x0D)	

Response: There is no response on this command.

Input short circuit

This function short circuit the module input.

Command:	##(Addr)R2\r	
	##:	Command leading code
	Addr:	Module address (2 characters hex from 0x00 to 0xFE) Sending the Address FF takes effect on all connected bridge modules.
	R2:	Command
	\r:	Carriage return (0x0D)
Response:	There is no response on this command.	

Input activate

This function is the inverse function of previous.

Command: ##(Addr)R3\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE) Sending the address FF takes effect on all connected bridge modules.
R3:	Command
\r:	Carriage return (0x0D)

Response: There is no response on this command.

Warning: The module is approximately 0.25 seconds off-line after sending this command.

Activate shunt resistor

Activates the shunt resistor, depending on the selected bridge completion type.

Standard shunts: 350 Ohm bridge completion: 175 kOhm
120 Ohm bridge completion: 59.88 kOhm

Command: ##(Addr)R4\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE) Sending the address FF takes effect on all connected bridge modules.
R4:	Command
\r:	Carriage return (0x0D)

Response: There is no response on this command.

Warning: The module is approximately 0.25 seconds off-line after sending this command.

Deactivate shunt resistor

Deactivates the shunt resistor.

Command: ##(Addr)R5\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE) Sending the address FF takes effect on all connected bridge modules.
R5:	Command
\r:	Carriage return (0x0D)

Response: There is no response on this command.

Warning: The Module is approximately 0.25 seconds off-line after sending this command.

Activate power on default

Command: ##(Addr)P(InputRange)(Filter)(Excitation)(Mode)(FilterType)(Button Lock)\r

##:	Command leading code	
Addr:	Acknowledged new module address	
InputRange:	Current measuring range (2 characters hex, according to table)	
Filter:	Current low pass filter (2 characters hex, according to table)	
Excitation:	Current Excitation Voltage (2 characters hex, according to table)	
Mode:	Current bridge type (1 character hex, according to table)	
FilterType:	0	Bessel characteristic
	1	Butterworth characteristic
Button lock:	0	Buttons unlocked
	1	Buttons locked
\r:	Carriage return (0x0D)	

Response:	!(Addr)\r	Valid command
	?(Addr)\r	Invalid command, syntax error
Example:	Command:	##01P000103010\r
	Response:	!ACK\r

(Set the module to 50 mV/V; 5 kHz; 5 V; full bridge; Butterworth; local buttons on; at power on)

Deactivate power on default

Deactivates the power on default mode.
The last pushbutton selected range and filter is adjusted at power on time.

Command: ##(Addr)P\r

##:	Command leading code
Addr:	Acknowledged new module address
P	Command
\r:	Carriage return (0x0D)

Response:	!(Addr)\r	Valid command
	?(Addr)\r	Invalid command, syntax error
Example:	Command:	##01P\r
	Response:	!ACK\r

Read Serial Number

Command: ##(Addr)SETB\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE) Sending the address FF takes effect on all connected bridge modules.
SETB:	Command
\r:	Carriage return (0x0D)

Response: !(SERAL)xx(REVISION)xxx\r

(SERIAL):	Module serial number (6 digits)
x:	space
(Revision):	Revision number (i.e. V200)
\r:	Carriage return (0x0D)

DAQP-BRIDGE-A

Nr	Command					Response		Function
1	##	AA	SETD		CR	!AA18RRFFESCMTRZ	CR	Program address
2	??	AA			CR	!AA18RRFFESCMTRZ	CR	Read configuration
3	##	AA+	R7		CR			Auto balance
4	##	AA+	R8		CR			Internal amplifier calibration
5	##	AA		RRFFESCMTR	CR	!ACK	CR	SET configuration
6	##	AA+	R2		CR			Short circuit input on
7	##	AA+	R3		CR			Short circuit input off
8	##	AA+	R4		CR			Shunt on
9	##	AA+	R5		CR			Shunt off
10	##	AA+	R1		CR			Lock buttons
11	##	AA+	R0		CR			Unlock buttons
12	##	AA	SETB		CR	!{16*ASCII}	CR	Read serial number
13	##	AA	P	RRFFESCMTR	CR	!ACK	CR	Write power on default to module

AA	Address
AA+	Address FF = all modules
MM	Module type (18 hex)
S	Short circuit 0/1
C	Shunt resistor 0/1
Z	Special module (2)

RR	Range (1)
0x00	50 mV/V
0x01	20 mV/V
0x02	10 mV/V
0x03	5 mV/V
0x04	2 mV/V
0x05	1 mV/V

(1) @ 5 V Excitation

(2) Vers. 2.00 or higher

FF	Filter
0x00	20 kHz
0x01	5 kHz
0x02	1 kHz
0x03	100 Hz
0x04	10 Hz

M	Mode	Shunt
0x00	Full bridge	59.88 kOhm
0x01	Half bridge	59.88 kOhm
0x02	Quarter bridge 120	-
0x03	Quarter bridge 350	-
0x04	Half bridge	175 kOhm
0x05	Full bridge	175 kOhm

E	Excitation
0x00	0 V
0x01	1 V
0x02	2.5 V
0x03	5 V
0x04	10 V
0x05	0.25 V (2)
0x06	0.5 V (2)

T	Filter Type
0	Butterworth characteristic
1	Bessel characteristic

Programming commands DAQP-BRIDGE-B

Instruction set

Command	Syntax
Set module address	##(NewAddr)SETD\r
Read module configuration	??(Addr)\r
Set module configuration	##(Addr)(InputRange)(Filter)(Excitation)(ShortCircuit)(ShuntFunction)(Shunt)(Mode)(FilterType)(Button Lock)\r
Automatic sensor offset adjustment	##(Addr)R7\r
Internal amplifier zero	##(Addr)R8\r
Lock/Unlock module buttons	##(Addr)R(Button Lock)\r
Input short circuit	##(Addr)R2\r
Input activate	##(Addr)R3\r
Activate shunt resistor	##(Addr)R4\r
Deactivate shunt resistor	##(Addr)R5\r
Activate Power on default	##(Addr)P(InputRange)(Filter)(Excitation)(ShortCircuit)(ShuntFunction)(Shunt)(Mode)(FilterType)(Button Lock)\r
Deactivate Power on default	##(Addr)P\r
Read serial number	##(Addr)SETB\r
TEDS programming commands[1)]	-
[1)] Please refere to chapter: ‚TEDS programming commands‘ in the ‚General module overview‘.	

Commands in detail

Set module address

This command has to be sent as long as the filter button has been pressed on the module. After that, the new address is stored in an EEPROM and the module sends a response string to the system.

Command: ##(Addr)SETD\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
\r:	Carriage return (0x0D)

Response: !(Addr)1F(InputRange)(Filter)(Excitation)(ShortCircuit)(ShuntFunction)(Shunt)(Mode)(FilterType)(Button Lock)(Special)\r

!:	Response leading code	
Addr:	Acknowledged new module address	
1F:	Module type (DAQP-BRIDGE-B)	
InputRange:	Measuring range (2 characters hex, according to table)	
Filter:	Low pass filter (2 characters hex, according to table)	
Excitation:	Excitation voltage (1 character hex, according to table)	
ShortCircuit:	0	Input activated
	1	Input short circuit
ShuntFunct.:	Shunt function (1 character, according to table)	
Shunt:	0	Shunt activated
	1	Shunt deactivated
Mode:	Bridge type (1 character hex, according to table)	
Filter Type:	0	Butterworth characteristic
	1	Bessel characteristic
Button lock:	0	Buttons unlocked
	1	Buttons locked
Special:	0	Standard module
	1	Special module

Example: Command: ##01SETD\r

Response: !011F000130211100\r

(DAQP-BRIDGE-B; 100 mV/V; 100 kHz; 1V; no short circuit; shunt 2; shunt activated; half bridge; Bessel filter; key’s on; standard module)

DAQP-BRIDGE-B Module

Range and filter selection table

Range	Code	Filter	Code
100 mV/V	0x00	Off	0x00
50 mV/V	0x01	100 kHz	0x01
20 mV/V	0x02	30 kHz	0x02
10 mV/V	0x03	10 kHz	0x03
5 mV/V	0x04	3 kHz	0x04
2 mV/V	0x05	1 kHz	0x05
1 mV/V	0x06	300 Hz	0x06
0.5 mV/V	0x07	100 Hz	0x07
0.2 mV/V	0x08	30 Hz	0x08
0.1 mV/V	0x09	10 Hz	0x09

Bridge type

Code	Mode
0x00	Full Bridge
0x01	Half Bridge
0x02	Quarter Bridge 120 Ohm (3-wire)
0x03	Quarter Bridge 350 Ohm (3-wire)
0x04	Quarter Bridge 120 Ohm (4-wire)
0x05	Quarter Bridge 350 Ohm (4-wire)

Excitation

Code	Excitation
0x00	0 V
0x01	0.25 V
0x02	0.5 V
0x03	1 V
0x04	2.5 V
0x05	5 V
0x06	10 V

Shunt function

Code	Shunt function
0x00	No shunt
0x01	Shunt 1
0x02	Shunt 2
0x03	External shunt
0x04	+ 9 V output

Read module configuration

Command: ??(Addr)\r

??: Command leading code
Addr: Module address (2 characters hex from 0x00 to 0xFE)
\r: Carriage return (0x0D)

Response: !(Addr)1F(InputRange)(Filter)(Excitation)(ShortCircuit)(ShuntFunction)(Shunt)(Mode)(FilterType)(Button Lock)(Special)\r

!: Response leading code
Addr: Module address
1F: Module type (DAQP-BRIDGE-B)
InputRange: Measuring range (2 characters hex, according to table)
Filter: Low pass filter (2 characters hex, according to table)
Excitation: Excitation voltage (1 character hex, according to table)
ShortCircuit: 0 Input activated
1 Input short circuit
Shunt: 0 Shunt deactivated
1 Shunt activated
ShuntSelect: Current shunt (1 character, according to table)
Mode: Bridge type (1 character hex, according to table)
Filter Type: 0 Butterworth characteristic
1 Bessel characteristic
Button lock: 0 Buttons unlocked
1 Buttons locked
Special: 0 Standard module
1 Special module

Example: Command: ??01\r

Response: !011F000130021000\r

(DAQP-BRIDGE-B; 100 mV/V; 100 kHz; 1V; no short circuit; shunt activated; shunt 2; half bridge; Bessel filter; key's on; standard module)

Set module configuration

Command: ##(Addr)(InputRange)(Filter)(Excitation)(ShortCircuit)(ShuntFunction)(Shunt)(Mode)(Filter type)(Button Lock)\r

!:	Response leading code	
Addr:	Module address	
Input Range:	Measuring range (2 characters hex, according to table)	
Filter:	Low pass filter (2 characters hex, according to table)	
Excitation:	Excitation voltage (1 character hex, according to table)	
ShortCircuit:	0	Input activated
	1	Input short circuit
Shunt:	0	Shunt deactivated
	1	Shunt activated
ShuntSelect:	Current shunt (1 character, according to table)	
Mode:	Bridge type (1 character hex, according to table)	
FilterType:	0	Bessel Characteristic
	1	Butterworth Characteristic
Button lock:	0	Buttons unlocked
	1	Buttons locked

Response:	!(Addr)\r	Valid command
	?(Addr)\r	Invalid command, syntax error

Automatic sensor offset adjustment

Sets the actual Sensor offset to Zero. The maximum adjustment range is +/-200% of the input Range.

Command: ##(Addr)R7\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE) Sending the Address FF takes affect on all connected Bridge modules.
R7:	Command
\r:	Carriage return (0x0D)

Response: There is no response on this command.
Warning: The Module is approximately 2 seconds off-line after sending this command. That means it cant receive any command.

Internal amplifier zero

This function short circuit the Module input, and measures the offset values from the different input ranges. The module automatically corrects the output voltage with this offset values.
Also the values are permanently stored in to the modules Memory.

Command: ##(Addr)R8\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE) Sending the Address FF takes affect on all connected Bridge modules.
R8:	Command
\r:	Carriage return (0x0D)

Response: There is no response on this command.
Warning: The Module is approximately 15 seconds off-line after sending this command. That means it cant receive any command.

If there is no sensor connected, activate the half bridge mode before sending this command. Otherwise you will get an output overflow.

Lock/Unlock module buttons

Command: ##(Addr)R(Button Lock)\r

##:	Command leading code	
Addr:	Module address (2 characters hex from 0x00 to 0xFE)	
	Sending the Address FF takes affect on all connected Bridge modules.	
R:	Command	
Button lock:	0	buttons unlocked
	1	buttons locked
\r:	Carriage return (0x0D)	

Response: There is no response on this command.

Input short circuit on

This function short circuit the Module input.

Command: ##(Addr)R2\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
	Sending the Address 0xFF takes effect on all connected Bridge modules.
R2	Command
\r:	Carriage return (0x0D)

Response: There is no response on this command.

Warning: The Module is approximately 0.25 seconds off-line after sending this command.

Input short circuit off

This function is the inverse function of previous.

Command: ##(Addr)R3\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
	Sending the Address 0xFF takes effect on all connected Bridge modules.
R3	Command
\r:	Carriage return (0x0D)

Response: There is no response on this command.

Warning: The module is approximately 0.25 seconds off-line after sending this command.

Activate shunt resistor

Activates the selected shunt function refered to shunt function table.

Shunt 1:	175 kOhm
Shunt 2:	59.88 kOhm

Command: ##(Addr)R4\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
	Sending the address 0xFF takes effect on all connected bridge modules.
R4:	Command
\r:	Carriage return (0x0D)

Response: There is no response on this command.

Warning: The module is approximately 0.25 seconds off-line after sending this command.

Deactivate shunt resistor

Deactivates the shunt resistor.

Command: ##(Addr)R5\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE) Sending the address 0xFF takes effect on all connected bridge modules.
R5	Command
\r:	Carriage return (0x0D)

Response: There is no response on this command.

Warning: The Module is approximately 0.25 seconds off-line after sending this command.

Activate power on default

Command: ##(Addr)P(InputRange)(Filter)(Excitation)(ShuntSelect)(Mode)(FilterType)(Button Lock)\r

##:	Command leading code	
Addr:	Acknowledged new module address	
InputRange:	Current measuring range (2 characters hex, according to table)	
Filter:	Current low pass filter (2 characters hex, according to table)	
Excitation:	Current Excitation Voltage (1 character hex, according to table)	
ShuntSelect:	Current shunt (1 character, according to table)	
Mode:	Current bridge type (1 character hex, according to table)	
FilterType:	0	Butterworth characteristic
	1	Bessel characteristic
Button lock:	0	Buttons unlocked
	1	Buttons locked
\r:	Carriage return (0x0D)	

Response:	!(Addr)\r	Valid command
	?(Addr)\r	Invalid command, syntax error

Example: Command: ##01P000103010\r

Response: !ACK\r

(Set the module to 50 mV/V; 5 kHz; 5 V; full bridge; Butterworth; local buttons on; at power on)

Deactivate power on default

Deactivates the power on default mode.
The last pushbutton selected range and filter is adjusted at power on time.

Command: ##(Addr)P\r

##:	Command leading code
Addr:	Acknowledged new module address
P:	Command
\r:	Carriage return (0x0D)

Response:	!(Addr)\r	Valid command
	?(Addr)\r	Invalid command, syntax error

Example: Command: ##01P\r

Response: !ACK\r

Read Serial Number

Command: ##(Addr)SETB\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE) Sending the address 0xFF takes effect on all connected bridge modules.
SETB:	Command
\r:	Carriage return (0x0D)

Response: !(SERAL)xx(REVISION)xxx\r

(SERIAL):	Module serial number (6 digits)
x:	space
(Revision):	Revision number (i.e. V200)
\r:	Carriage return (0x0D)

Notes

DAQP-BRIDGE-B

Nr:	Command					Response		Function	REV
1	##	AA	SETD		CR	!AA1FRRFFESCsCMTRZ	CR	Program address	2+
2	??	AA			CR	!AA1FRRFFESCsCMTRZ	CR	Read configuration	2+
3	##	AA+	R7		CR			Auto balance	0
4	##	AA+	R8		CR			Internal amplifier calibration	0
5	##	AA		RRFFESCsCMTR	CR	!ACK	CR	Set configuration	0
6	##	AA		RRFFESCsCMTR	CR	!NoACK	CR	Set configuration	0
7	##	AA+	R2		CR			Short circuit input on	0
8	##	AA+	R3		CR			Short circuit input off	0
9	##	AA+	R4		CR			Shunt on	0
10	##	AA+	R5		CR			Shunt off	0
11	##	AA+	R1		CR			Lock buttons	0
12	##	AA+	R0		CR			Unlock buttons	0
13	##	AA	SETB		CR	!{16*ASCII}	CR	Read serial number	0
14	##	AA	P	RRFFESCsCMTR	CR	!ACK	CR	Write power on default	0
15	??	AA	SETD		CR	!AA1FRRFFESCsCMTR	CR	Program address	1
16		AA			CR	!AA1FRRFFESCsCMTR	CR	Read configuration	1

AA	Address
AA+	Address
	FF = all modules
R	Button Lock
S	Short circuit 0/1
C	Shunt resistor 0/1
Z	Special module

Cs	Shunt function
0x00	No shunt
0x01	Shunt I
0x02	Shunt II
0x03	External shunt
0x04	+ 9 V output

M	Mode
0x00	Full bridge
0x01	Half bridge
0x02	Quarter bridge 120 (3-Wire)
0x03	Quarter bridge 350 (3-Wire)
0x04	Quarter bridge 120 (4-Wire)
0x05	Quarter bridge 350 (4-Wire)

T	Filter type
0	Butterworth characteristic
1	Bessel characteristic

E	Excitation
0x00	0 V
0x01	0.25 V
0x02	0.5 V
0x03	1 V
0x04	2.5 V
0x05	5 V
0x06	10 V

FF	Filter
0x00	OFF
0x01	100 kHz
0x02	30 kHz
0x03	10 kHz
0x04	3 kHz
0x05	1 kHz
0x06	300 Hz
0x07	100 Hz
0x08	30 Hz
0x09	10 Hz

RR	Range
0x00	100 mV/V
0x01	50 mV/V
0x02	20 mV/V
0x03	10 mV/V
0x04	5 mV/V
0x05	2 mV/V
0x06	1 mV/V
0x07	0.5 mV/V
0x08	0.2 mV/V
0x09	0.1 mV/V

Programming commands DAQP-CFB

Instruction set

Command	Syntax
Set module address	##(NewAddr)SETD\r
Read module configuration	??(Addr)\r
Set module configuration	##(Addr)(InputRange)(Filter)(Excitation)(ExcMode)(ShortCircuit)(ShuntFct)(Shunt)(Mode)(FilterType)(Remote)\r
Set remote control	##(Addr)Rx\r
Input short curcuit	##(Addr)Rx(Short)\r
Shunt resistor	##(Addr)Rx(Shunt)\r
Automatic sensor offset adjustment	##(Addr)R7\r
Internal amplifier zero	##(Addr)R8\r
Automatic C-balance adjustment	##(Addr)RB\r
Set CAL-signal	##(Addr)R(CAL)\r
Read/Write offset values	##(Addr)SETZx[13(±yyyy)][S]\r
Write sensor offset value	##(Addr)SETS±CCCC(S)\r
Read/Write phase shift value	##(Addr)SETLxx\r
Read gain calibration values	##(Addr)SETKx\r
Read/Write custom range	##(Addr)SETX A ±BBBB ±CCCC ±DDDD EEEE F (S)\r
Activate Power On Default	##(Addr)SETP(InputRange)(Filter)(Excitation)(ExcMode)(ShuntFct)(Mode)(FilterType)(Remote)\r
Deactivate Power On Default	##(Addr)SETP\r
Read Serial Number	##(Addr)SETB\r

Ranges overview

			1 Vrms excitation			2 Vrms excitation			5 Vrms excitation			
Range	Software scaling factor	Range code	Bridge mode	Inductive bridge mode	Max offset adjustment [%]	Bridge mode	Inductive bridge mode	Max offset adjustment [%]	Bridge mode	Inductive bridge mode	Max offset adjustment [%]	
1000 mV	200	0x00	■	■	200	■	■	■	■	■	■	
500 mV	100	0x01	■	■	200	■	■	400	■	■	■	
200 mV	40	0x02	■	■	200	■	■	400	■	■	500	Low gain
100 mV	20	0x03	■	■	200	■	■	400	■	■	1000	
50 mV	10	0x04	■	■	200	■	■	400	■	■	1000	
20 mV	4	0x05	■	■	200	■	■	400	■	■	1000	
10 mV	2	0x06	■	■	200	■	■	400	■	■	1000	
5 mV	1	0x07	■	■	200	■	■	400	■	■	1000	
2 mV	0.4	0x08	■	■	200	■	■	400	■	■	500	High gain
1 mV	0.2	0x09	■	■	200	■	■	400	■	■	1000	
0.5 mV	0.1	0x0A	■	■	200	■	■	400	■	■	1000	
0.2 mV	0.04	0x0B	■	■	200	■	■	400	■	■	1000	
0.1 mV	0.02	0x0C	■	■	■	■	■	400	■	■	1000	
Custom		0xCC										

■ supported
■ not supported

Commands in detail

Set module address

This command has to be sent as long as the filter button has been pressed on the module. After that, the new address is stored in an EEPROM and the module sends a response string to the system.

Command: ##(Addr)SETD\r

##:	Command leading code	
Addr:	Module address (2 characters hex from 0x00 to 0xFE)	
\r:	Carriage return (0x0D)	

Response: !(Addr)2C(InputRange)(Filter)(Excitation)(ExcMode)(ShortCircuit)(ShuntFct)(Shunt)(Mode)(FilterType)(CBalance)(Remote)(Special)\r

!:	Response leading code	
Addr:	Acknowledged new module address	
2C:	Module type (DAQP-CFB)	
InputRange:	Measuring range (2 characters hex, according to table)	
Filter:	Low pass filter (2 characters hex, according to table)	
Excitation:	Excitation voltage (1 character hex, according to table)	
Exc Mode:	Excitation mode (1 character hex, according to table)	
ShortCircuit:	0	Input activated
	1	Input short circuit
ShuntFunct.:	Shunt function (1 character, according to table)	
Shunt:	0	Shunt deactivated
	1	Shunt activated
Mode:	Bridge type (1 character hex, according to table)	
Filter Type:	0	Butterworth characteristic
	1	Bessel characteristic
CBalance:	0	Amplifier C-balance
	1	Amplifier unbalanced
Remote:	0	Local mode (Module button activated)
	1	Remote mode (Module button locked)
Special:	0	Standard module
	1	Special module

Example: Command: ##01SETD\r
Response: !012C0001000101010\r

(DAQP-CFB; 1000 mV/V; 1000 Hz; 1 Vrms; internal, no sync output; Input activated; shunt disabled; shunt activated; full bridge (strain); C-balance; Bessel filter; Remote mode; standard module)

Range and filter selection table

Range	Code	Filter	Code
1000 mV	0x00	OFF	0x00
500 mV	0x01	1000 Hz	0x01
200 mV	0x02	300 Hz	0x02
100 mV	0x03	100 Hz	0x03
50 mV	0x04	30 Hz	0x04
20 mV	0x05	10 Hz	0x05
10 mV	0x06	-	0x06
5 mV	0x07	-	0x07
2 mV	0x08	-	0x08
1 mV	0x09	-	0x09
0.5 mV	0x0A	-	0x0A
0.2 mV	0x0B	-	0x0B
0.1 mV	0x0C	-	0x0C
CUSTOM	0xCC	-	0xCC

Bridge type

Code	Mode
0x00	Full bridge (strain)
0x01	Half bridge (strain)
0x02	Quarter bridge 120 Ohm
0x03	Quarter bridge 350 Ohm
0x04	Full bridge (inductive) (refere to range overview table)
0x05	Half bridge (inductive) (refere to range overview table)

Shunt function

Code	Shunt funtion
0x00	Shunt disabled
0x01	Shunt resistor 1 (100 kOhm default)
0x02	Shunt resistor 1 (50 kOhm default)

Excitation

Code	Excitation
0x00	1 Vrms
0x01	2 Vrms
0x02	5 Vrms

Excitation mode

Code	Excitation mode
0x00	Internal, no sync output
0x01	Internal, with sync output
0x02	External

Read module configuration

Command: ??(Addr)\r

??: Command leading code
Addr: Module address (2 characters hex from 0x00 to 0xFE)
\r: Carriage return (0x0D)

Response: !(Addr)2C(InputRange)(Filter)(Excitation)(ExcMode)(ShortCircuit)(ShuntFct)(Shunt)(Mode)(FilterType)(CBalance)(Remote)(Special)\r

!: Response leading code
Addr: Acknowledged new module address
2C: Module type (DAQP-CFB)
InputRange: Measuring range (2 characters hex, according to table)
Filter: Low pass filter (2 characters hex, according to table)
Excitation: Excitation voltage (1 character hex, according to table)
Exc Mode: Excitation mode (1 character hex, according to table)
ShortCircuit: 0 Input activated
1 Input short circuit
ShuntFunct.: Shunt function (1 character, according to table)
Shunt: 0 Shunt deactivated
1 Shunt activated
Mode: Bridge type (1 character hex, according to table)
Filter Type: 0 Butterworth characteristic
1 Bessel characteristic
CBalance: 0 Amplifier C-balance
1 Amplifier unbalanced
Remote: 0 Local mode (Module button activated)
1 Remote mode (Module button locked)
Special: 0 Standard module
1 Special module

Example: Command: ##01SETD\r
Response: !012C0001000101010\r

(DAQP-CFB; 1000 mV/V; 1000 Hz; 1 Vrms; internal, no sync output; Input activated; shunt disabled; shunt activated; full bridge (strain); C-balance; Bessel filter; Remote mode; standard module)

Set module configuration

Command: ##(Addr)(InputRange)(Filter)(Excitation)(ExcMode)(ShortCircuit)(ShuntFct)(Shunt)(Mode)(Filter Type)(Remote)\r

##:	Command leading code	
Addr:	Module address (2 characters hex from 0x00 to 0xFE)	
\r:	Carriage return (0D hex)	
InputRange:	Measuring range (2 characters hex, according to table)	
Filter:	Low pass filter (2 characters hex, according to table)	
Excitation:	Excitation voltage (1 character hex, according to table)	
Exc Mode:	Excitation mode (1 character hex, according to table)	
ShortCircuit:	0	Input activated
	1	Input short circuit
ShuntFunct.:	Shunt function (1 character, according to table)	
Shunt:	0	Shunt deactivated
	1	Shunt activated
Mode:	Bridge type (1 character hex, according to table)	
Filter Type:	0	Butterworth characteristic
	1	Bessel characteristic
CBalance:	0	Amplifier C-balance
	1	Amplifier unbalanced
Remote:	0	Module button activated
	1	Module button locked
Special:	0	Standard module
	1	Special module

Response:	!ACK\r	Valid command
	!N0ACK\r	If custom range is selected and custom gain is set to 0000h

Set Remote Control

Command: ##(Addr)R(remote)\r

##:	Command leading code	
Addr:	Module address (2 characters hex; address 0xFF takes effect on all modules)	
R:	Command	
Remote:	0:	Local mode
	1:	Remote Mode (Front Panel Buttons locked)
\r:	Carriage return (0x0D)	

Response: !ACK\r

Example: Command: ##04050201\r

(DAQP-CFB; 1000 mV/V; 1000 Hz; 1 Vrms; internal, no sync output; Input activated; shunt disabled; shunt deactivated; full bridge (strain); C-balance; Bessel filter; Remote mode; standard module)

Response: !ACK\r

Input short circuit on/off

This function short circuit the Module input.

Command: ##(Addr)Rx\r

##:	Command leading code	
Addr:	Module address (2 characters hex from 0x00 to 0xFE) Sending the Address 0xFF takes effect on all connected Bridge modules.	
Rx	2:	input is short cut
	3:	input is in standard measurement mode
\r:	Carriage return (0x0D)	

Response: There is no response on this command.

Warning: The Module is approximately 0.25 seconds off-line after sending this command.

Activate/Deactivate shunt resistor

Activates the selected shunt function refered to shunt function table.

	Shunt 1:	175 kOhm
	Shunt 2:	59.88 kOhm
Command:	##(Addr)Rx\r	
	##:	Command leading code
	Addr:	Module address (2 characters hex from 0x00 to 0xFE) Sending the address 0xFF takes effect on all connected bridge modules.
	Rx:	Command 4: Activates the selected shunt (refere to ShuntFunct) 5: Switching off the shunt resistor
	\r:	Carriage return (0x0D)
Response:	There is no response on this command.	
Warning:	The module is approximately 0.25 seconds off-line after sending this command.	

Automatic sensor offset adjustment

Sets the actual Sensor offset to Zero. The maximum adjustment range depends on the excitation voltage:
5 Vrms: ±400 % of range
2 Vrms: ±400 % of range
1 Vrms: ±200 % of range
This function is not available in Custom Range.

Command:	##(Addr)R7\r	
	##:	Command leading code
	Addr:	Module address (2 characters hex from 0x00 to 0xFE) Sending the Address 0xFF takes affect on all connected modules.
	R7:	Command
	\r:	Carriage return (0x0D)
Response:	!ACK\r	Valid command. There is no response on this command.
Warning:	The Module is approximately 2 seconds off-line after sending this command. That means it can't receive any command.	

Internal amplifier zero

This function short circuit the Module input, and measures the offset values from the different input ranges. The module automatically corrects the output voltage with this offset values.
Also the values are permanently stored in to the modules Memory.

Command:	##(Addr)R8\r	
	##:	Command leading code
	Addr:	Module address (2 characters hex from 0x00 to 0xFE) Sending the Address FF takes affect on all connected modules.
	R8:	Command
	\r:	Carriage return (0x0D)
Response:	!ACK\r:	Valid command. There is no response on this command.
Warning:	The Module is approximately 2 seconds off-line after sending this command. That means it cant receive any command.	

Automatic C-Balance adjustment (strain mode)

The input parallel capacitors are set to optimise the range of the C-Balance circuit.

Command: ##(Addr)RB\r

##:	Command leading code
Addr:	Module address (2 characters from 0x00 to 0xFE)

Sending the Address FF takes affect on all connected modules.

RB:	Command
\r:	Carriage return (0x0D hex)

Response:

!ACK\r	C-Balance successful
!NOACK\r	C-Balance not successful or module is currently in inductive mode (mode 4 or 5)

There is no response on this command if address was set to 0xFF!

Warning: The Module is approximately 2 seconds off-line after sending this command. That means it can't receive any command.

Set CAL-signal on/off

Apply a 80 % Signal to the Output.

Command: ##(Addr)R(CAL)\r

##:	Command leading code	
Addr:	Module address (2 characters hex from 0x00 to 0xFE)	

Sending the Address FF takes affect on all connected modules.

R:	Command	
CAL:	D:	switching calibration signal ON
	E:	switching calibration signal OFF
\r:	Carriage return (0x0D)	

Response:

!ACK\r	Valid command. There is no response on this command if address was set to 0xFF!

Read/Write offset values

This command is used to write additional offset values to the module.

Maximum offset adjustment ranges:
1 V OffsetAdjustmentRange: 200% of Range
2 V OffsetAdjustmentRange: 400% of Range
5 V OffsetAdjustmentRange: 1000% of Range

Command: ##(Addr)SETZx[13(±yyyy)][S]\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
SETZ:	Command
x:	0: Amplifier Zero offset DAC values for 1 Vrms excitation (read only) 1: Amplifier Zero offset DAC values for 2 Vrms excitation (read only) 2: Amplifier Zero offset DAC values for 5 Vrms excitation (read only) 3: Sensor offset value (read only) 4: Customer offset value (have to be recalculated for excitation voltage change)
[13(±yyyy)]:	optional: Offset DAC values for the ranges beginning from 1000 mV/V down to 0.1 mV/V
[S]:	optional: Sending “S” stores the values also in the internal EEPROM – without “S” it is just stored in the internal RAM.
\r:	Carriage return (0x0D)

Response: !ACK\r Valid command.
Sending without optional strings returns the actual values.

Example: ##(Addr)SETZ0\r -> Response: !13x(±yyyy)\r

Setting custom offset:

Calculate the appropriate Hex value:

Sign:	positive or negative offset
Offset:	wanted offset in % of actual range
Ext:	actual excitation voltage
HexVal:	Value that has to be sent to the module to get the desired offset. (4 digits) HexVal = [sign]convert to hex (65535(Offset/OffsetAdjustmentRange[excitation]))

Example: minus 100 % offset desired, 5 V excitation, 50 mV/V
Hex Val = [-]convert to hex (65535(100/1000))=-199A

Typical module commands for setting offset of a DAQP-CFB module on address 08 to -100 %:

Set Module to 50 mV/V 5 V Excitation:
##08040320001001
!ACK

Perform Amplifier Zero:
##08R8
!ACK

Perform Sensor Zero:
##08R7
!ACK

Set offset of all input ranges to -100 %:
##08SETZ4-199A-199A-199A-199A-199A-199A-199A-199A-199A-199A-199A-199A-199A S\r

Write sensor offset value

Caution: This command is only available in Firmware revision 1.44 (module revision V105) or higher.

This command read/writes the sensor offset value in % of the actual range.

Command: ##(Addr)SETS±CCCC(S)\r

##:	Command leading code
Addr:	Module address (2 characters hex from 00 to FE)
SETS:	Command
±CCCC:	Sensor offset in ASCII hex
(S):	Sending ‚S' stores the calues also in the internal E^2PROM. Without ‚S' it is just stored in the internal RAM
\r:	Carriage return (0D hex)

Response: !ACK\r Valid command.

Sending ‚\r' instead of the calibration values, the module responses the stored values from the E^2PROM.
Example: ##(Addr)SETS\r -> response: !±CCCC\r

The resolution of the output offset = 0.005%

Example table:

+CCCC = 0x0000	No output offset
+CCCC = 0x0001	Output offset = 0.005% of range
+CCCC = 0x000A	Output offset = 0.05% of range
+CCCC = 0x0064	Output offset = 0.5% of range
-CCCC = 0x0001	Output offset = -0.005% of range
-CCCC = 0x000A	Output offset = -0.05% of range

Note: For all ranges higher then 1 V (or for all gains <5) the maximum sensor offset is 20 %.
For the 1 V ranges or lower (or gain 5 and higher) the offset range is 200 %.

Command: ##(Addr)SETS±CCCC(S)\r
Response: !(Addr)ACK\r or !(Addr)NOACK\r

Read/Write phase shift value (inductive mode)

Writing will automatically store the ideal phase shift value of the potentiometer (Ideal hase). This is the potentiometer value having no phase shift at the input.

Command: ##(Addr)SETLxx\r

##:	Command leading code
Addr:	Module address (2 characters hex from 00 to FE)
	Sending the Address FF takes affect on all connected Bridge modules.
SETS:	Command
xx:	8 Bit ASCII Hex value (IdealPhase)
yy:	Actual phase shift potentiometer value (read only)
\r:	Carriage return (0x0D)

Response: !ACK\r Valid command or
!xx yy\r

Sending \r instead of the ADC value the amplifier responses IdealPhase and the actual phasse shift potentiometer value.

Example: ##(Addr)SETL\r -> Response: !xx yy\r

Note: 00 ~ = 51 °C, FF ~ -48 °C

Read gain calibration values

This command reads out the values of the gain DAC related to the associated selected ranges and excitation voltage.

Command: ##(Addr)SETKx\r

##:	Command leading code
Addr:	Module address (2 characters hex from 00 to FE)
SETK:	Command
x:	0: Calibrated Gain DAC values for 1 Vrms excitation
	1: Calibrated Gain DAC values for 2 Vrms excitation
	2: Calibrated Gain DAC values for 5 Vrms excitation
	3: Reserved
	4: Reserved
\r:	Carriage return (0D hex)

Response: !13*EEEE\r

EEEE internal gain DAC values in ASCII-Hex beginning from 1000mV/V range to 0.1mV/V .

Example:

Send: ##0FSETK0

Response: !FC0A 7E05 3268 1934 0C9A 050A FCDB 7E6D 3292 1949 0CA5 050F 0287

1000 mV/V DAC value pre amplifier gain Low < | > 0.1 mV/V DAC value pre amplifier gain High

Read/write custom range

If in the command: “Set module configuration” the custom range is used the values for the gain and offset DAC are used depending on this values. It could be used to scale the module output signal to the physical sensor input signal. Sending this command automatically activates the custom range (refer to InputRange).

Command: ##(Addr)SETX A ±BBBB ±CCCC ±DDDD EEEE F (S)\r

##:	Command leading code
Addr:	Module address (2 characters hex from 00 to FE)
	Sending the Address FF takes affect on all connected Bridge modules.
SETX:	Command
A:	Excitation voltage
	0: 1 Vrms
	1: 2 Vrms
	2: 5 Vrms
±BBBB:	Amplifier offset value (Bit value of DAC)
	Revision up to firmware 1.40: The value could be set from software or determined with the amplifier balance function.
	Firmware 1.50 or higher: The value is automatically calculated from stored amplifier zero values when switching to custom range. The value will be determined exactly if an amplifier balance is applied in the custom range. The value is read only.
±CCCC:	Sensor offset Value: Automatically measured during automatic sensor offset adjustment. ±CCCC=DecToHex(65536*(Sensor offset[%]/OffAdjRange[%]))
±DDDD:	Output offset value (percent of range). ±DDDD=DecToHex(65536*(Sensor offset[%]/OffAdjRange[%]))
EEEE:	Gain DAC value
	For the calculation of the gain DAC value of a certain range, it is required to read out the internal gain calibration values first. The new gain DAC value is calculated by using the nearest calibration point in the formula: CustomGainDACValue=(Custom Range/CalibratedRange)*CalibratedRangeDACValue
F:	0: Input amplifier set to gain 1 (low Gain)
	1: Input amplifier set to gain 100 (high Gain)
(S):	Sending “S” stores the values also in the internal E2Prom – without “S” it is just stored in the internal RAM.
\r:	Carriage return (0D hex)

Response: !ACK\r Valid command.
Sending \r instead of the calibration values the module responses the stored values.
Example: ##(Addr)SETX\r -> Response: !A ±BBBB ±CCCC ±DDDD EEEE F\r

Example 1:

Setting the DAQP-CFB to an input range of 1.043 mV/V at 2 V excitation:

1.Read out the 2V gain calibration values

Send: ##0FSETK1
Receive: !FFFF FC0A 64D1 3268 1934 0A15 050B FCE1 6527 3293 194A 0A1D 050F

2.Calculate the new gain DAC value

Calibrated Range: 1.00 mV/V
CalibratedRangeDACValue: 3293 hexadecimal = 12947 decimal

CustomGainDACValue=round(CustomRange/CalibratedRange)*CalibratedRangeDACValue
=(1.043/1)*12947=13504dec= 34BFhex

3. Activate the custom range by sending set module configuration

Send: ##0FCC0011011000
Receive: !ACK

4. Setup the Custom Range:

Send: ##0FSETX 1 +0000 +0000 +0000 34BF 1 S

Example 2:

Setting the DAQP-CFB to 50% output offset; 2 V excitation, 1 mV/V

1. Calculate the offset

Customer offset value:= [sign]convert to hex (65535(Offset/OffsetAdjustmentRange[excitation]))
Customer offset value:= [sign]convert to hex (50*65535/400)
Customer offset value:= [sign]convert to hex 8192
Customer offset value:= +2000

2. Read the actual setting:

Send: ##03SETX
Receive: !1 +0029 +0000 +0000 34BF 1

3. Write new offset:

Send: ##03SETX1 +0029 +000 +2000 34BF 1
Receive: !ACK

Activate power on default

Command: ##(Addr)P(InputRange)(Filter)(Excitation)(ExcMode)(ShuntFct)(Mode)(FilterType)(Remote)\r

##:	Command leading code	
Addr:	Acknowledged new module address	
P:	Command	
InputRange:	Current measuring range (2 characters hex, according to table)	
Filter:	Current low pass filter (2 characters hex, according to table)	
Excitation:	Current Excitation Voltage (1 character hex, according to table)	
ExcMode:	Excitation mode (1 character hex, according to table)	
ShuntFunct.:	Shunt function (1 character, according to table)	
Mode:	Current bridge type (1 character hex, according to table)	
FilterType:	0	Butterworth characteristic
	1	Bessel characteristic
Remote:	0	Local mode (Module button activated)
	1	Remote mode (Module button locked)
\r:	Carriage return (0x0D)	

Response: !ACK\r Valid command
!N0ACKr If custom range is selected and custom gain is set to 0000h

Example: Command: ##01P0001030100\r

Response: !ACK\r

(Set the module to 1000 mV/V; 1000 Hz; 1 V; external; Shunt disabled; Half bridge; Butterworth; Local mode; at power on)

Deactivate power on default

Deactivates the power on default mode.
The last pushbutton selected range and filter is adjusted at power on time.

Command: ##(Addr)P\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
	Sending the address 0xFF takes affect on all connected modules.
P:	Command
\r:	Carriage return (0x0D)

Response: !(Addr)\r Valid command

Example: Command: ##01P\r
Response: !ACK\r

Read Serial Number

Command: ##(Addr)SETB\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
	Sending the address 0xFF takes effect on all connected bridge modules.
SETB:	Command
\r:	Carriage return (0x0D)

Response: !xxxxxxxxxxxxxxxx\r
16 characters for the serial number, not used characters are filled with blank (20h)

Notes

Programming commands DAQP-STG

Instruction set

Command	Syntax
Set module address	##(NewAddr)SETD\r
Read module configuration	??(Addr)\r
Set module configuration	##(Addr)(InputRange)(Filter)(Excitation)(ShortCircuit)(ShuntFct)(Shunt)(Mode)(FilterType)(ExcMode)\r
Input short curcuit	##(Addr)Rx\r
Shunt resistor	##(Addr)Rx\r
Automatic sensor offset adjustment	##(Addr)R7\r
Internal amplifier zero	##(Addr)R8\r
Set CAL-signal	##(Addr)R(CAL)\r
Set Excitation on/off	##(Addr)R(CAL)\r
Read/Write custom range (CS)	##(Addr)SETX A ±BBBB ±CCCC ±DDDD EEEE F (S)\r
Read/Write output offset value (CS)	##(Addr)SETO±yyyy(S)\r
Read/Write sensor offset value (CS)	##(Addr)SETS±CCCC(S)\r
Read/Reset sensor offset (CS)	##(Addr)SETVxy\r
Write input CAL-Mux mode (CS)	##(Addr)RSTG(Mux)(InputRange)(Filter)(Excitation)(ExcMode)\r
Activate Power On Default	##(Addr)SETPO\r
Read text from module (CS)	##(Addr)SETRx\r
TEDS programming commands[1)]	-
Disable checksum mode	##(Addr)R0\r
Read Serial Number	##(Addr)SETB\r
[1)] please refere to chapter: ‚TEDS programming commands' in the ‚General module overview'.	

Range and filter selection table

Range[1)]	Code	Range[2)]	Code	Filter	Code
NA	0x00	10000 mV	0x00	300 kHz	0x00
1000 mV/V	0x01	5000 mV	0x01	100 kHz	0x01
500 mV/V	0x02	2000 mV	0x02	30 kHz	0x02
200 mV/V	0x03	1000 mV	0x03	10 kHz	0x03
100 mV/V	0x04	500 mV	0x04	3 kHz	0x04
50 mV/V	0x05	250 mV	0x05	1 kHz	0x05
20 mV/V	0x06	100 mV	0x06	300 Hz	0x06
10 mV/V	0x07	50 mV	0x07	100 Hz	0x07
5 mV/V	0x08	25 mV	0x08	30 Hz	0x08
2 mV/V	0x09	10 mV	0x09	10 Hz	0x09
1 mV/V	0x0A	5 mV	0x0A	-	0x0A
0.5 mV/V	0x0B	2.5 mV	0x0B	-	0x0B
0.2 mV/V	0x0C	1 mV	0x0C	-	0x0C
0.1 mV/V	0x0D	0.5 mV	0x0D	-	0x0D
CUSTOM	0xCC	CUSTOM	0xCC	-	0xCC

[1)] in mv/V @ 5 V xcitation

[2)] in V @ Voltage mode

Bridge type

Code	Mode
0x00	Full bridge
0x01	Half bridge
0x02	Quarter bridge 120 Ohm 3-wire
0x03	Quarter bridge 350 Ohm 3-wire
0x04	Quarter bridge 120 Ohm 4-wire
0x05	Quarter bridge 350 Ohm 4-wire
0x06	Voltage
0x07	Resistance
0x08	PT100
0x09	PT200
0x0A	PT500
0x0B	PT1000
0x0C	PT2000
0x0D	CUSTOM 1
0x0E	CUSTOM 2
0x0F	CUSTOM 3

Excitation

Code	Excitation (Voltage)	(Current)
0x00	0 V	(0.1 mA)
0x01	0.25 V	(0.2 mA)
0x02	0.5 V	(0.5 mA)
0x03	1 V	(1 mA)
0x04	2.5 V	(2 mA)
0x05	5 V	(5 mA)
0x06	10 V	(10 mA)
0x07	12 V	(20 mA)
0x0C	CUSTOM	CUSTOM

Shunt function

Code	Shunt function
0x00	Shunt disabled
0x01	Shunt resistor 1 (175 kOhm default)
0x02	Shunt resistor 2 (59.88 kOhm default)
0x03	Shunt resistor 3 (not installed default)

Commands in detail

Set module address

This command has to be sent as long as the filter button has been pressed on the module. After that, the new address is stored in an EEPROM and the module sends a response string to the system.

Command: ##(Addr)SETD\r

##:	Command leading code	
Addr:	Module address (2 characters hex from 0x00 to 0xFE)	
\r:	Carriage return (0x0D)	

Response: !(Addr)35(InputRange)(Filter)(Excitation)(ShortCircuit)(ShuntFct)(Shunt)(Mode)(FilterType)(ExcMode)(Special)(CS)\r

!:	Response leading code	
Addr:	Acknowledged new module address	
35:	Module type (DAQP-STG)	
InputRange:	Measuring range (2 characters hex, according to table)	
Filter:	Low pass filter (2 characters hex, according to table)	
Excitation:	Excitation voltage (1 character hex, according to table)	
ShortCircuit:	0	Input activated
	1	Input short circuit
ShuntFunct.:	Shunt function (1 character, according to table)	
Shunt:	0	Shunt deactivated
	1	Shunt activated
Mode:	Bridge type (1 character hex, according to table)	
Filter Type:	0	Butterworth characteristic
	1	Bessel characteristic
Exc Mode:	0	Voltage
	1	Current
Special:	0	Standard module
	1	Special module
CS:	CheckSum (2 characters hex)	

Example: Command: ##01SETD\r
Response: !0135000100010101000\r

(DAQP-STG; 100000 mV/V; 1000 Hz; 100 kHz; 0V (0.1 mA); Input activated; shunt disabled; shunt activated; full bridge; C-balance; Bessel filter; Voltage; special module; CheckSum)

Read module configuration

Command: ??(Addr)\r

??:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
\r:	Carriage return (0x0D)

Response: !(Addr)35(InputRange)(Filter)(Excitation)(ShortCircuit)(ShuntFct)(Shunt)(Mode)(FilterType)(ExcMode)(Special)(CS)\r

!:	Response leading code	
Addr:	Acknowledged new module address	
35:	Module type (DAQP-STG)	
InputRange:	Measuring range (2 characters hex, according to table)	
Filter:	Low pass filter (2 characters hex, according to table)	
Excitation:	Excitation voltage (1 character hex, according to table)	
ShortCircuit:	0	Input activated
	1	Input short circuit
ShuntFunct.:	Shunt function (1 character, according to table)	
Shunt:	0	Shunt deactivated
	1	Shunt activated
Mode:	Bridge type (1 character hex, according to table)	
Filter Type:	0	Butterworth characteristic
	1	Bessel characteristic
Exc Mode:	0	Voltage
	1	Current
Special:	0	Standard module
	1	Special module
CS:	CheckSum (2 characters hex)	

Example: Command: ##01SETD\r
Response: !013500010001010100\r

(DAQP-STG; 100000 mV/V; 1000 Hz; 100 kHz; 0V (0.1 mA); Input activated; shunt disabled; shunt activated; full bridge; C-balance; Bessel filter; Voltage; special module; CheckSum)

Set module configuration (CheckSum available)

Command: ##(Addr)(InputRange)(Filter)(Excitation)(ShortCircuit)(ShuntFct)(Shunt)
(Mode)(FilterType)(ExcMode)\r

##:	Command leading code	
Addr:	Acknowledged new module address	
InputRange:	Measuring range (2 characters hex, according to table)	
Filter:	Low pass filter (2 characters hex, according to table)	
Excitation:	Excitation voltage (1 character hex, according to table)	
ShortCircuit:	0	Input activated
	1	Input short circuit
ShuntFunct.:	Shunt function (1 character, according to table)	
Shunt:	0	Shunt deactivated
	1	Shunt activated
Mode:	Bridge type (1 character hex, according to table)	
Filter Type:	0	Butterworth characteristic
	1	Bessel characteristic
Exc Mode:	0	Voltage
	1	Current

Response: !ACK\r Valid command
?NOACK\r Wrong paramter or communication between PIC fails or custom range/excitation is selected but not defined.

Command with CS:
$#(Addr)(InputRange)(Filter)(Excitation)(ShortCircuit)(ShuntFct)(Shunt)
(Mode)(FilterType)(ExcMode)(CS)\r

Response: !(Addr)ACK(CS)\r
!(Addr)NOACK(CS)\r

Input short circuit on/off

This function short circuit the Module input.

Command: ##(Addr)Rx\r

##:	Command leading code	
Addr:	Module address (2 characters hex from 0x00 to 0xFE)	
	Sending the Address 0xFF takes effect on all connected Bridge modules.	
Rx:	2:	input is short cut
	3:	input is in standard measurement mode
\r:	Carriage return (0x0D)	

Response: !ACK\r Valid command.
!NOACK\r Wrong parameter or communication between PIC fails.

Warning: The Module is approximately 0.25 seconds off-line after sending this command.

Activate/Deactivate shunt resistor

Command: ##(Addr)Rx\r

##:	Command leading code	
Addr:	Module address (2 characters hex from 0x00 to 0xFE)	
	Sending the address 0xFF takes effect on all connected bridge modules.	
Rx:	Command	
	4:	Activates the selected shunt (refere to ShuntFunct)
	5:	Switching off the shunt resistor
\r:	Carriage return (0x0D)	

Response: !(Addr)ACK(CS)\r
!(Addr)NOACK(CS)\r

Warning: The module is approximately 0.25 seconds off-line after sending this command.

Automatic sensor offset adjustment

This function is not available in Custom Range.

Command: ##(Addr)R7\r

##:	Command leading code	
Addr:	Module address (2 characters hex from 0x00 to 0xFE)	
	Sending the Address 0xFF takes affect on all connected modules.	
R7:	Command	
\r:	Carriage return (0x0D)	

Response:

!ACK\r	Valid command.
!NOACK\r	Wrong parameter or communication between PIC fails.

If the address is set to FF then
there is no response
the sensor offset adjustment is only performed, if the module is in bridge mode.

Warning: The Module is approximately 2 seconds off-line after sending this command. That means it cant receive any command.

Internal amplifier zero

This function short circuit the Module input, and measures the offset values from the different input ranges. The module automatically corrects the output voltage with this offset values.
Also the values are permanently stored in to the modules Memory.

Command: ##(Addr)R8\r

##:	Command leading code	
Addr:	Module address (2 characters hex from 0x00 to 0xFE)	
	Sending the Address FF takes affect on all connected Bridge modules.	
R8:	Command	
\r:	Carriage return (0x0D)	

Response:

!ACK\r:	Valid command. There is no response on this command if address is set to 0xFF.
!NOACK\r:	Wrong parameter or communication between PIC fails.

Warning: The Module is approximately 8 seconds off-line after sending this command. That means it cant receive any command.

Set CAL-signal on/off

Apply a 80 % Signal to the Output.

Command: ##(Addr)R(CAL)\r

##:	Command leading code	
Addr:	Module address (2 characters hex from 0x00 to 0xFE)	
Sending the Address FF takes affect on all connected modules.		
R:	Command	
CAL:	D:	switching calibration signal ON
	E:	switching calibration signal OFF
\r:	Carriage return (0x0D)	

Response:

!ACK\r	Valid command. There is no response on this command if address was set to 0xFF!

Set Excitation on/off

Command: ##(Addr)R(CAL)\r

##:	Command leading code	
Addr:	Module address (2 characters hex from 0x00 to 0xFE) Sending the Address FF takes affect on all connected modules.	
R:	Command	
CAL:	F:	switching excitation ON
	G:	switching excitation OFF
\r:	Carriage return (0x0D)	

Response:

!ACK\r	Valid command. There is no response on this command if address is set to 0xFF!
!NOACK\r	Wrong parameter or communication between PIC fails.

Note: This command is ignored by the module, if measurement mode voltage is used! Reason: There may be a sensor connected with needed power supply -> result is anyway not correct.

Read/write custom range (CS available)

If in the command: “Set module configuration” the custom range is used the values for the gain and offset DAC are used depending on this values. The gain could be set to any value between 0.5 and 10 000.

Command: ##(Addr)SETX AAAA B EEEE(S)\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE) Sending the Address FF takes affect on all connected Bridge modules.
SETX:	Command
AAAA:	Gain Value Base (4 characters hex)
B:	Gain Value Exponent (1character hex)
EEEE:	Excitation (4 characters hex) 0.2 mV resolution in voltage mode, from 0 V to 12 V 1 µA resolution in current mode, from 0.1 mA to 20 mA
(S):	Sending “S” stores the values also in the internal E2Prom – without “S” it is just stored in the internal RAM.
\r:	Carriage return (0x0D)

Response:

!ACK\r	Valid command.
!NOACK\r	Wrong parameter or communication between PIC fails.

Sending \r instead of the custom values the module responses the stored values.

Example: ##(Addr)SETX\r -> Response: !AAAA B EEEE\r

Calculating gain out of AAAA and B:
Gain= AAAA * $2^{(B/65536)}$;
Calculating AAAA and B out of wanted gain:
B=trunc(log2(gain * 2)
AAAA=trunc((gain * 2^{16}) / 2^{B})

If valid customer range (AAAA > 0) or excitation (EEEE > 0) is written to the module the range is automatically changed to the custom range and/or excitation. If module is currently in custom range or excitation and AAAA and/or EEEE is set to “0000”, the module changes to Range Code “00” and/or Excitation Code “1”.

If customer range is used, automatically the amplifier offset adjust for this range is performed.
Note: The amplifier offset adjustment is not done, if the module is set with the range code to the customer range!

Example: 0.12 V Range; 3 V Excitation

Gain: output voltage / input voltage = 5 V / 0.12 V = 41666

$B_{decimal}$ = trunc(log2(41666 * 2)=6	B=6
$AAAA_{decimal}$= trunc((41666 * 2^{16}) / 2^{6})=42666;	AAAA=A6AA
Excitation = 3 V	
$EEEE_{decimal}$= 3/0.0002=15000	EEEE=3A98

Writing custom range:

Command: #04SETX A6AA 6 3A98\r

Response: !04ACK

Reading custom range:

Command: 04 SETX\r

Response: !A6AA 6 3A98\r

Command with CS:

Command: $#(Addr)SETX AAAA B EEEE (S)(CS)\r

Response: !(Addr)ACK(CS)\r or !(Addr)NOACK(CS)\r

or

Command: $#(Addr)SETX(CS)\r

Response: !(Addr)AAAA B EEEE(CS)\r

Read/write output offset value (CS available)

This command R/W the output offset value in % of range.

Command: ##(Addr)SETO±yyyy(S)\r

##:	Command leading code
Addr:	Module address (2 characters hex from 00 to FE)
SETO:	Command
±yyyy:	Output offset in ASCII HEX
(S):	Sending “S” stores the values also in the internal E2Prom – without “S” it is just sto red in the internal RAM.
\r:	Carriage return (0D hex)

Response: !ACK\r Valid command.

Sending \r instead of the calibration values the module responses the stored values from the EEPROM.

Example: ##(Addr)SETO\r
Response: !±yyyy\r

The resolution of the output offset = 0.005%

Example table:

+yyyy = 0x0000	No output offset
+yyyy = 0x0001	Output offset = 0.005% of range
+yyyy = 0x000A	Output offset = 0.05% of range
+yyyy = 0x0064	Output offset = 0.5% of range
-yyyy = 0x0001	Output offset = -0.005% of range
-yyyy = 0x000A	Output offset = -0.05% of range

Note: For all ranges higher then 1 Volt (or for all gains < 5) the allow output offset is 20%!!
For the 1 Volt ranges on lower (or gain 5 and higher) the output offset range is 200%!

Command with CS:

Command: ##(Addr)SETO±yyyy(S)(CS)\r

Response: !(Addr)ACK(CS)\r or !(Addr)NOACK(CS)\r

or

Command $#(Addr)SETO(CS)\r

Response: !(Addr)±yyyy(CS)\r

Read/write sensor offset value (CS available)

Caution: This command is only available in Firmware revision 1.44 (module revision V105) or higher.

This command read/writes the sensor offset value in % of the actual range.

Command: ##(Addr)SETS±CCCC(S)\r

##:	Command leading code
Addr:	Module address (2 characters hex from 00 to FE)
SETS:	Command
±CCCC:	Sensor offset in ASCII hex
(S):	Sending ‚S' stores the calues also in the internal E^2PROM. Without ‚S' it is just stored in the internal RAM
\r:	Carriage return (0D hex)

Response: !ACK\r Valid command.

Sending ‚\r' instead of the calibration values, the module responses the stored values from the E^2PROM.
Example: ##(Addr)SETS\r -> response: !±CCCC\r

The resolution of the output offset = 0.005%

Example table:

+CCCC = 0x0000	No output offset
+CCCC = 0x0001	Output offset = 0.005% of range
+CCCC = 0x000A	Output offset = 0.05% of range
+CCCC = 0x0064	Output offset = 0.5% of range
-CCCC = 0x0001	Output offset = -0.005% of range
-CCCC = 0x000A	Output offset = -0.05% of range

Note: For all ranges higher then 1 V (or for all gains <5) the maximum sensor offset is 20 %.
For the 1 V ranges or lower (or gain 5 and higher) the offset range is 200 %.

Command with CS:

Command: $#(Addr)SETS±CCCC(S)(CS)\r

Response: !(Addr)ACK(CS)\r or !(Addr)NOACK(CS)\r

or:

Command: $#(Addr)SETS(CS)\r

Response: !(Addr)±CCCC(CS)\r

Read/reset sensor offset adjustment settings (CS available)

If automatic sensor offset adjustment is done, the offset value and the basic amplifier settings when adjustment is performed are stored. Based on this values the Offset DAC value can be calculated if the range or the excitation voltage is changed after this offset calibration.

Command: ##(Addr)SETVxy\r

##:	Command leading code
Addr:	Module address (2 characters hex from 00 to FE)
SETV:	Command
x:	0: Module output offset not changed
	1: Module output offset set to “+0000”
y:	0: No reset on sensor output offset
	1: Sensor offset set to “0000”
\r:	Carriage return (0D hex)

Note on sensor offset reset:
Similar like Amplifier offset, all Sensor offsets needs to be cleared:
SensOffDacVal[Range] = AmpOffCalVal[Range]
SenseOffsetValPerc = SenseOffsetGainBase = SenseOffsetGainExp =0;

Response:	!ACK\r	Valid command.
	!NOACK\r	Wrong parameter

If “\r” is followed after the command (ex. “##(Addr)SETV\r”) the module responses the settings of the sensor offset adjustment.

Response: !AAAA B ±CCCC ±DDDD EEEE F\r

AAAA:	Gain Value Base
B:	Gain Value Exponent
Formula for calculating the gain: AAAA*2^B/65536	
±CCCC:	Sensor offset Value (percent of Range) in ASCII
±DDDD:	Module output offset Value (percent of Range) in ASCII
This is the same value you can read back from command 13!	
EEEE:	Excitation
	0.2mV resolution in voltage mode
	1µA resolution in current mode
F:	Excitation Mode (0: voltage, 1 current)
\r:	Carriage return (0x0D hex)

To reset the customer output offset

The resolution of the output offset = 0.005%

Example table:

+DDDD = 0x0000	No output offset
+DDDD = 0x0001	Output offset = 0.005% of range
+DDDD = 0x000A	Output offset = 0.05% of range
+DDDD = 0x0064	Output offset = 0.5% of range
-DDDD = 0x0001	Output offset = -0.005% of range
-DDDD = 0x000A	Output offset = -0.05% of range

Command with CS:

Command: $#(Addr)SETV(CS)\r

Response: !(Addr)AAAA B ±CCCC ±DDDD EEEE F(CS)\r

or:

Command: $#(Addr)STEVxy(CS)\r

Response: !(Addr)ACK(CS)\r or !(Addr)NOACK(CS)\r

Write input Cal-Mux mode (CS available)

Command: ##(Addr)RSTG(Mux)(InputRange)(Filter)(Excitation)(ExcMode)\r

##:	Command leading code
Addr:	Module address (2 characters hex from 00 to FE)
Sending the Address FF takes affect on all connected Bridge modules.	
RSTG:	Command
Mux:	0: Set to Exc+
	1: Set to HBE (=GND Level)
	2: Set to Sense+
	3: Set to Exc-
InputRange:	Range code -> refer to module configuration command
	If range code = FF -> range will be not changed
Filter:	Filter code -> refer to module configuration command
Excitation:	Excitation code -> refer to module configuration command
	If excitation code = F -> excitation level will be not changed
ExcMode:	0: Excitation in voltage mode
	1: Excitation in current mode
	2: Auto setting for InputRange and excitation level and mode (setting of InputRange and Excitation are ingnored)
	F: ExcMode not changed

Response:	!ACK\r	Valid command.
	!NOACK\r	Wrong parameter or communication between PIC fails.

Sending “\r” instead of (Mux) to (Excitation) sets the module to the previous measurement settings: ##(Addr) RSTG\r

The Cal Mux mode is automatically switched of, if any other command is send to the module.

Handling for setting gain and excitation in ExcMode = 2:
Definition: Bridge Mode means module in Mode 0 .. 5

If Bridge Mode and Excitation Mode is Voltage then
    If (Mode == Quarter bridge 120 Ohm 4 wire)
        Mode = Quarter bridge 120 Ohm 3 wire
    Else If (Mode == Quarter bridge 350 Ohm 4 wire)
        Mode = Quarter bridge 350 Ohm 3 wire
    Range is set to equal or higher value of Excitation
Else If Bridge Mode and Excitation Mode is Current then
    Range is 1 Volt
    Excitation is 1 Volt
Else if ((Mode == VoltagMode) && (ExcMode == Voltage))
    If (ExcVoltage == 0Volt)
        Range is 1 Volt
        Excitation is 1 Volt
    Else If (ExcVoltage < 0.25Volt)
No settings are changed
Else If Voltage ExcVoltage >= 0.25Volt)
        Range is set to equal or higher value of Excitation
Else If (Resistance or PT xxx mode or (Voltage Mode and ExcMode == Current)) and Range >= 0.25Volt
    Excitation Mode is Voltage
    Excitation Level is equal or lower then Range
If (Resistance or PT xxx mode or (Voltage Mode and ExcMode == Current)) and Range < 0.25Volt
    No settings are changed

Command with CS:

Command: $#(Addr)RSTG (Mux)(InputRange)(Filter)(Excitation)(ExcMode)(CS)\r
Response: !(Addr)ACK(CS)\r or !(Addr)NOACK(CS)\r

Command: $#(Addr)RSTG(CS)\r
Response: !(Addr)ACK(CS)\r or !(Addr)NOACK(CS)\r

Activate power-on default

Command: ##(Addr)SETPO\r

##:	Command leading code
Addr:	Module address (2 characters hex from 00 to FE)
Sending the Address FF takes affect on all connected Bridge modules.	
SETPO:	Command
\r:	Carriage return (0D hex)

Response: !ACK\r Valid command

There is no response on this command if address was set to FF!

Read text from module (CS available)

Command: ##(Addr)SETRx\r

##:	Command leading code
Addr:	Module address (2 characters hex from 00 to FE)
Sending the Address FF takes affect on all connected Bridge modules.	
SETR:	Command
x:	Storing address from 0 to F
\r:	Carriage return (0D hex)

Response: The text from the associated storing address.

Command with CS:

Command: $#(Addr)SETRx(CS)\r
Response: !(Addr)[Text](CS)\r

Disable Cecksum mode

Command: ##(Addr)R0\r

##:	Command leading code
Addr:	Module address (2 characters hex from 00 to FE)
Sending the Address FF takes affect on all connected Bridge modules.	
R0:	Command

Response: !ACK\r Valid command

Sending this command, the module accepts also commands without CS until again once a CS command is sent to the module.

Read serial number

Command: ##(Addr)SETB\r

##:	Command leading code
Addr:	Module address (2 characters hex from 00 to FE)
Sending the Address FF takes affect on all connected Bridge modules.	
SETB:	Command
\r:	Carriage return (0D hex)

Response: !xxxxxxxxxxxxxxxx\r 16 character for the serial number, not used characters are filled with blank (20h)

Notes

Programming commands DAQP-TRQ

Instruction set

Command	Syntax
Set module address	##(NewAddr)SETD\r
Read module configuration	??(Addr)\r
Set module configuration	##(Addr)(Range)(Filter)(Button Lock)\r
Lock/Unlock module buttons	##(Addr)Rx\r
Read serial (1)	##(Addr)SETB\r
(1) Commands applied in 08-2003	

Commands in detail

Set module address

This command has to be sent as long as the filter button is pressed on the module. After that, the new address is stored in an EEPROM and the module sends a response string to the system.

Command: ##(NewAddr)SETD\r

##:	Command leading code
NewAddr:	New module address (2 characters hex from 0x00 to 0xFE)
SETD:	Set address command
\r:	Carriage return (0x0D)

Response: !(NewAddr)(ModuleType)(Range)(Filter)(Button Lock)\r

!:	Response leading code
NewAddr:	Confirmed new module address (2 characters hex)
ModuleType:	Type of module (2 characters hex, 07 according general prog. info)
Range:	Measuring range position (2 characters hex, according table)
Filter:	Filter position (2 characters hex, according table)
Button lock:	Lock/Unlock module (1 character, according general prog. info)

Example: Command: ##01SETD\r

Response: !010704020\r
(Address 0x01, DAQP-TRQ, range E, 100 Hz, buttons unlocked)

Range and filter selection table

Range	Code	Filter	Code
A	0x00	1 kHz	0x00
B	0x01	300 Hz	0x01
C	0x02	100 Hz	0x02
D	0x03	30 Hz	0x03
E	0x04	10 Hz	0x04
F	0x05	-	-

DAQP-TRQ Module

Read module configuration

Command: ??(Addr)\r

??:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
\r:	Carriage return (0x0D)

Response: !(Addr)(ModuleType)(Range)(Filter)(Button Lock)\r

Response details see also ‘Set module address’

Example: Command: ??01\r

Response: !010704020\r
(Address 0x01, DAQP-TRQ, range E, 100 Hz, buttons unlocked)

Set module configuration

Command: ##(Addr)(Range)(Filter)(Button Lock)\r

Command details see also ‘Set module address’

Response: !ACK\r (Notice: an incorrect command get no response from module!)

!:	Response leading code
ACK:	Acknowledge
\r:	Carriage return (0x0D)

Example: Command: ##0104020\r
(Address 0x01, range E, 100 Hz, buttons unlocked)

Response: !ACK\r

Lock/Unlock module buttons

Command: ##(Addr)Rx\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE) Address 0xFF sets all modules with one command
Rx:	Module buttons lock/unlock command 0: buttons unlocked 1: buttons locked
\r:	Carriage return (0x0D)

Response: no response

Example: Command: ##01R0\r
(Module address 0x01, buttons unlocked)

Command: ##01R1\r
(Module address 0x01, buttons locked)

Read serial number

Command: ##(Addr)SETB\r

Command details see also ‘Set module address’

Response: !(Serial)xx(Revision)xxx\r

(Serial):	Module Serial Number (6 digits)
x:	Space
(Revision):	Revision Number (i.e. V200)
\r:	Carriage return (0x0D)

Programming commands DAQP-ACC-A

Instruction set

Command	Syntax
Set module address	##(NewAddr)SETD\r
Read module configuration	??(Addr)\r
Set module configuration	##(Addr)(Range)(Filter)(Current)(TimeConstant)(Button Lock)\r
Lock/Unlock module buttons	##(Addr)Rx\r
Activate Power-on default	##(Addr)P(Range)(Filter)(Current)(TimeConstant)(Button Lock)\r
Read serial (1)	##(Addr)SETB\r
(1) Commands applied in 08-2003	

Commands in detail

Set module address

This command has to be sent as long as the filter button has been pressed on the module. After that, the new address is stored in an EEPROM and the module sends a response string to the system.

Command: ##(NewAddr)SETD\r

##:	Command leading code
NewAddr:	New module address (2 characters hex from 0x00 to 0xFE)
SETD:	Set address command
\r:	Carriage return (0x0D)

Response: !(NewAddr)(ModuleType)(Range)(Filter)(Current)(TimeConstant)(Error)(Special)(Button Lock)\r

!:	Response leading code
NewAddr:	Confirmed new module address (2 characters hex)
ModuleType:	Type of module (2 characters hex, 0x1B according general prog. info)
Range:	Measuring range position (2 characters hex, according table)
Filter:	Filter position (2 characters hex, according table)
Current:	Selection between 4 mA and 8 mA (1 character, according table)
TimeConstant:	Input time constant of amplifier (1 character hex) 0: 5 seconds 1: 25 seconds
Error:	Error on module input (1 character, according table)
Special:	Reserved
Button lock:	Lock/Unlock module (1 character, according general prog. info)

Example: Command: ##01SETD\r

Response: !010902010000\r
(Address 0x01, DAQP-ACC, x10, 100 kHz, 4 mA, 5 sec., no error, unlocked)

Sensor supply

Code	Sensor supply
0	4 mA sensor supply
1	8 mA sensor supply

Error codes

Code	Error code
0	No input error
1	No sensor connected
2	Input short circuit

Range and filter selection table

Range	Code	Filter	Code
1	0x00	300 kHz	0x00
3	0x01	100 kHz	0x01
10	0x02	10 kHz	0x02
30	0x03	1 kHz	0x03
100	0x04	-	-

Read module configuration

Command: ??(Addr)\r

??:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
\r:	Carriage return (0x0D)

Response: !(Addr)(ModuleType)(Range)(Filter)(Current)(TimeConstant)(Error)(Special)(Button Lock)\r

Response details see also ‘Set module address’

Set module configuration

Command: ##(Addr)(Range)(Filter)(Current)(TimeConstant)(Button Lock)\r

Command details see also ‘Set module address’

Response: !ACK\r (Notice: an incorrect command gets no response from module!)

!:	Response leading code
ACK:	Acknowledge
\r:	Carriage return (0x0D)

Example: Command: ##010002110\r
(Address 0x01, 1, 10 kHz, 8 mA, 25 sec., unlocked)

Response: !ACK\r

Activate power-on default

Command: ##(Addr)P(Range)(Filter)(Current)(TimeConstant)(Button Lock)\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE) Address FF sets all modules with one command
Range:	Measuring range position (2 characters hex, according table)
Filter:	Filter position (2 characters hex, according table)
Current:	Selection between 4 mA and 8 mA (1 character, according table)
TimeConstant:	Input time constant of amplifier (1 character hex) 0: 5 seconds 1: 25 seconds
Button lock:	Lock/Unlock module (1 character, according general prog. info)
\r:	Carriage return (0x0D)

Sending an '\r' right after 'P' deactivates the power-on default and the last pushbutton selected range and Filter is adjusted at power on time.

Read serial number

Command: ##(Addr)SETB\r

Command details see also ‘Set module address’

Response: !(Serial)xx(Revision)xxx\r

(Serial):	Module Serial Number (6 digits)
x:	Space
(Revision):	Revision Number (i.e. V200)
\r:	Carriage return (0x0D)

Programming commands DAQP-CHARGE

Instruction set

Command	Syntax
Set module address	##(NewAddr)SETD\r
Read module configuration	??(Addr)\r
Set module configuration	##(Addr)(Range)(Filter)(InputType)(FineTuning)(Button Lock)\r
Lock/Unlock module buttons	##(Addr)R(Button Lock)\r

Commands in detail

Set module address

This command has to be sent as long as the filter button has been pressed on the module. After that, the new address is stored in an EEPROM and the module sends a response string to the system.

Command: ##(NewAddr)SETD\r

##:	Command leading code
NewAddr:	New module address (2 characters hex from 0x00 to 0xFE)
SETD:	Set address command
\r:	Carriage return (0x0D)

Response: !(NewAddr)(ModuleType)(Range)(Filter)(InputType)(FineTuning)(Overflow)(Button Lock)\r

!:	Response leading code
NewAddr:	Confirmed new module address (2 characters hex)
ModuleType:	Type of module (2 characters hex, 0x05 according general prog. info)
Range:	Measuring range position (2 characters hex, according table)
Filter:	Filter position (2 characters hex, according table)
InputType:	Selection between ‘charge’ and ‘ICP’ input (1 character, according table)
FineTuning:	Input amplifier fine tuning (3 characters hex) Detailed fine tuning information on next page
Overflow:	Overflow indicator (correspond with ‘Overflow’-LED)
Button lock:	Lock/Unlock module (1 character, according general prog. info)

Example: Command: ##01SETD\r

Response: !0105007F019900\r
(Address 0x01, DAQP-CHARGE, 0 dB, 20 kHz, ICP, 1:10, no overflow, local)

Input types

Code	Function
0	ICP input active
1	Charge input active

Overflow status

Code	Function
0	No input overflow
1	Input overflow

Range and filter selection table

Range	Code	Filter	Code
0 dB	0x00	20 kHz	0x7F
20 dB	0x01	10 kHz	0x74
40 dB	0x02	5 kHz	0x63
60 dB	0x03	1 kHz	0x0F

Notice: Detailed filter selection table on next page.

Fine tuning

As standard, the input divider is set to 1:10. In this case, the input ranges 0, 20, 40 and 60 dB are valid.

The module allows to varying the input division from 1:10 to 1:1 in 3686 steps. This feature can be used to do a sensor calibration directly in the module.

Division 1:10 199 hex (standard range, e.g. 0 dB)
Division 1:1 FFF hex (next range, e.g. 20 dB)

All values between 199 and FFF hex are valid for module programming. As soon as the division is above 1:10 (199 hex), the input range LED's are not active.

Detailed filter selection table

Filter No. [dec]	Code [hex]	Range [kHz]	Filter No. [dec]	Code [hex]	Range [kHz]	Filter No. [dec]	Code [hex]	Range [kHz]	Filter No. [dec]	Code [hex]	Range [kHz]
0	00	0,82	32	20	1,29	64	40	2,93	96	60	5,12
1	01	0,83	33	21	1,32	65	41	2,97	97	61	5,24
2	02	0,84	34	22	1,34	66	42	3,01	98	62	5,36
3	03	0,85	35	23	1,37	67	43	3,05	99	63	5,50
4	04	0,86	36	24	1,39	68	44	3,09	100	64	5,65
5	05	0,87	37	25	1,42	69	45	3,13	101	65	5,80
6	06	0,88	38	26	1,45	70	46	3,18	102	66	5,96
7	07	0,89	39	27	1,48	71	47	3,23	103	67	6,12
8	08	0,90	40	28	1,51	72	48	3,28	104	68	6,31
9	09	0,91	41	29	1,54	73	49	3,32	105	69	6,49
10	0A	0,93	42	2A	1,58	74	4A	3,38	106	6A	6,69
11	0B	0,94	43	2B	1,61	75	4B	3,43	107	6B	6,90
12	0C	0,95	44	2C	1,65	76	4C	3,49	108	6C	7,12
13	0D	0,96	45	2D	1,69	77	4D	3,54	109	6D	7,35
14	0E	0,98	46	2E	1,73	78	4E	3,60	110	6E	7,61
15	0F	0,99	47	2F	1,77	79	4F	3,66	111	6F	7,88
16	10	1,00	48	30	1,82	80	50	3,73	112	70	8,18
17	11	1,02	49	31	1,86	81	51	3,79	113	71	8,51
18	12	1,03	50	32	1,91	82	52	3,85	114	72	8,86
19	13	1,05	51	33	1,97	83	53	3,93	115	73	9,24
20	14	1,06	52	34	2,02	84	54	4,00	116	74	9,66
21	15	1,08	53	35	2,08	85	55	4,07	117	75	10,12
22	16	1,09	54	36	2,14	86	56	4,15	118	76	10,61
23	17	1,11	55	37	2,16	87	57	4,23	119	77	11,10
24	18	1,13	56	38	2,28	88	58	4,31	120	78	11,66
25	19	1,15	57	39	2,35	89	59	4,40	121	79	12,33
26	1A	1,17	58	3A	2,43	90	5A	4,49	122	7A	13,09
27	1B	1,19	59	3B	2,52	91	5B	4,58	123	7B	13,90
28	1C	1,21	60	3C	2,61	92	5C	4,68	124	7C	14,82
29	1D	1,23	61	3D	2,71	93	5D	4,79	125	7D	15,92
30	1E	1,25	62	3E	2,81	94	5E	4,89	126	7E	17,19
31	1F	1,27	63	3F	2,93	95	5F	5,00	127	7F	18,75

Read module configuration

Command: ??(Addr)\r

??:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
\r:	Carriage return (0x0D)

Response: !(Addr)(ModuleType)(Range)(Filter)(InputType)(FineTuning)(Overflow)(Button Lock)\r

Response details see also ‘Set module address’

Example: Command: ??01\r

Response: !0105007F019900\r
(Address 0x01, DAQP-CHARGE, 0 dB, 20 kHz, ICP, 1:10, no overflow, local)

Set module configuration

Command: ##(Addr)(Range)(Filter)(InputType)(FineTuning)(Button Lock)\r

Command details see also ‘Set module address’

Response: !ACK\r (Notice: an incorrect command gets no response from module!)

!:	Response leading code
ACK:	Acknowledge
\r:	Carriage return (0x0D)

Example: Command: ##01007F01990\r
(Address 0x01, 0 dB, 20 kHz, ICP, 1:10, local)

Response: !ACK\r

Lock/Unlock module buttons

Command: ##(Addr)Rx\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE) Address FF sets all modules with one command
Rx:	Module buttons lock/unlock command 0: buttons unlocked 1: buttons locked
\r:	Carriage return (0x0D)

Response: no response

Example: Command: ##01R0\r
(Module address 0x01, buttons unlocked)

Command: ##01R1\r
(Module address 0x01, buttons locked)

DAQP-CHARGE

Nr:	Command					Response		Function
1	##	AA	SETD		CR	!AA05RRFFitttOk	CR	Program address
2	??	AA			CR	!AA05RRFFitttOk	CR	Read configuration
3	##	AA		RRFFitttk	CR	!ACK	CR	SET configuration
4	##	AA+	R1		CR			Lock buttons
5	##	AA+	R0		CR			Unlock buttons

AA	Address
{adr+}	Address
	FF = all modules
MM	Module type (05)
ttt	Fine tuning
k	Button lock

RR	Range
0x00	0 dB
0x01	20 dB
0x02	40 dB
0x03	60 dB

i	Input type
0	ICP input active
1	Charge input active

O	Overflow status
0	No input overflow
1	Input overflow

FF Detailed filter selection table

Filter No. [dec]	Code [hex]	Range [kHz]	Filter No. [dec]	Code [hex]	Range [kHz]	Filter No. [dec]	Code [hex]	Range [kHz]	Filter No. [dec]	Code [hex]	Range [kHz]
0	00	0,82	32	20	1,29	64	40	2,93	96	60	5,12
1	01	0,83	33	21	1,32	65	41	2,97	97	61	5,24
2	02	0,84	34	22	1,34	66	42	3,01	98	62	5,36
3	03	0,85	35	23	1,37	67	43	3,05	99	63	5,50
4	04	0,86	36	24	1,39	68	44	3,09	100	64	5,65
5	05	0,87	37	25	1,42	69	45	3,13	101	65	5,80
6	06	0,88	38	26	1,45	70	46	3,18	102	66	5,96
7	07	0,89	39	27	1,48	71	47	3,23	103	67	6,12
8	08	0,90	40	28	1,51	72	48	3,28	104	68	6,31
9	09	0,91	41	29	1,54	73	49	3,32	105	69	6,49
10	0A	0,93	42	2A	1,58	74	4A	3,38	106	6A	6,69
11	0B	0,94	43	2B	1,61	75	4B	3,43	107	6B	6,90
12	0C	0,95	44	2C	1,65	76	4C	3,49	108	6C	7,12
13	0D	0,96	45	2D	1,69	77	4D	3,54	109	6D	7,35
14	0E	0,98	46	2E	1,73	78	4E	3,60	110	6E	7,61
15	0F	0,99	47	2F	1,77	79	4F	3,66	111	6F	7,88
16	10	1,00	48	30	1,82	80	50	3,73	112	70	8,18
17	11	1,02	49	31	1,86	81	51	3,79	113	71	8,51
18	12	1,03	50	32	1,91	82	52	3,85	114	72	8,86
19	13	1,05	51	33	1,97	83	53	3,93	115	73	9,24
20	14	1,06	52	34	2,02	84	54	4,00	116	74	9,66
21	15	1,08	53	35	2,08	85	55	4,07	117	75	10,12
22	16	1,09	54	36	2,14	86	56	4,15	118	76	10,61
23	17	1,11	55	37	2,16	87	57	4,23	119	77	11,10
24	18	1,13	56	38	2,28	88	58	4,31	120	78	11,66
25	19	1,15	57	39	2,35	89	59	4,40	121	79	12,33
26	1A	1,17	58	3A	2,43	90	5A	4,49	122	7A	13,09
27	1B	1,19	59	3B	2,52	91	5B	4,58	123	7B	13,90
28	1C	1,21	60	3C	2,61	92	5C	4,68	124	7C	14,82
29	1D	1,23	61	3D	2,71	93	5D	4,79	125	7D	15,92
30	1E	1,25	62	3E	2,81	94	5E	4,89	126	7E	17,19
31	1F	1,27	63	3F	2,93	95	5F	5,00	127	7F	18,75

Programming commands DAQP-CHARGE-A

Instruction set

Command	Syntax
Set module address	##(NewAddr)SETD\r
Read module configuration	??(Addr)\r
Set module configuration	##(Addr)(Range)(LowpassFilter)(Integration)(HighpassFilter)(InputType) (FineTuning)(Button Lock)\r
Lock/Unlock module buttons	##(Addr)R(Button Lock)\r
Activate power on default	##(Addr)P(Range)(LowpassFilter)(Integration)(HighpassFilter)(InputType)(FineTuning) (Button Lock)\r
Deactivate power on default	##(Addr)P\r
Read serial number	##(Addr)SETB\r

Commands in detail

Set module address

This command has to be sent as long as the filter button has been pressed on the module. After that, the new address is stored in an EEPROM and the module sends a response string to the system.

Command: ##(NewAddr)SETD\r

##:	Command leading code
NewAddr:	New module address (2 characters hex from 0x00 to 0xFE)
SETD:	Set address command
\r:	Carriage return (0x0D)

Response: !(NewAddr)(ModuleType)(Range)(LowpassFilter)(Integration)(HighpassFilter)(InputType) (FineTuning)(Overflow)(ICP-Voltage)(Button Lock)\r

!:	Response leading code
NewAddr:	Confirmed new module address (2 characters hex)
ModuleType:	Type of module (2 characters hex, 17 according general prog. info)
Range:	Amplifier range position (2 characters hex, according table)
LowpassFil.:	Output lowpass filter position (2 characters hex, according table)
Integration:	Integration mode (1 character, according table)
HighpassFil.:	Input highpass filter position (2 characters hex, according table)
InputType:	Selection between ‘charge’ and ‘ICP’ input (1 character, according table)
FineTuning:	Input amplifier fine tuning (3 characters hex 0x199 to 0xFFF)
Overflow:	Overflow indicator (correspond with ‘Overflow’-LED)
ICP-Voltage:	Current ICP-Voltage State (1 character, according table)
Button lock:	Lock/Unlock module (1 character, according general prog. info)

Example: Command: ##05SETD\r

Response: !051703020020FFF000

New address 05; DAQP-CHARGE-A module; 100 mV/pC (40db); lowpass 3 kHz; integration off; highpass 10 Hz; charge input active; 1:10; no overflow; no ICP sensor; buttons unlocked

DAQP-CHARGE-A Module

Input types

Code	Function
0	Charge input active
1	ICP input active

Overflow status

Code	Function
0	No input overflow
1	Input overflow

Low- highpass Filter

Code	Lowpass	Code	Highpass
0x00	50 kHz	0x00	0.1 Hz
0x01	10 kHz	0x01	1 Hz
0x02	3 kHz	0x02	10 Hz
0x03	1 kHz	-	-
0x04	0,1 kHz	-	-

Integration mode

Code	Function
0	Integration off
1	Single integration
2	Double integration

ICP-Voltage state

Code	Function
0	Sensor is inside range
1	ICP input short
2	ICP in idle

Amplifier range selection tables

Code	Charge	dB
0x00	0.1 mV/pC	-20
0x01	1 mV/pC	0
0x02	10 mV/pC	20
0x03	100 mV/pC	40
0x04	1000 mV/pC	60

Code	ICP	dB
0x00	5000 mV	0
0x01	500 mV	20
0x02	50 mV	40
0x03	5 mV	60

Fine tuning

As standard, the input divider is set to 1:10. In this case, the input ranges 0, 20, 40 and 60 dB are valid.

The module allows to varying the input division from 1:10 to 1:1 in 3686 steps. This feature can be used to do a sensor calibration directly in the module.

Division 1:1 199 hex
Division 1:5 8CC hex
Division 1:10 FFF hex

All values between 199 and FFF hex are valid for module programming. As soon as the division is above 1:10 (FFF hex), the input range LED's are not active.

Read module configuration

Command: ??(Addr)\r

??: Command leading code
Addr: Module address (2 characters hex from 0x00 to 0xFE)
\r: Carriage return (0x0D)

Response: !(Addr)(ModuleType)(Range)(LowpassFilter)(Integration)(HighpassFilter)(InputType)(FineTuning)(Overflow)(ICP-Voltage)(Button Lock)\r

Response details see also 'Set module address'

Example: Command: ??04\r

Response: !041703020021FFF000

New address 04; DAQP-CHARGE-A module; 5 mV (60 db); lowpass 3 kHz; integration off; highpass 10 Hz; ICP input active; 1:10; no overflow; sensor is inside range; buttons unlocked

Set module configuration

Command: ##(Addr)(Range)(LowpassFilter)(Integration)(HighpassFilter)(InputType)(FineTuning)(Button Lock)\r

Command details see also 'Set module address'

Response: !ACK\r (Notice: an incorrect command gets no response from module!)

!: Response leading code
ACK: Acknowledge
\r: Carriage return (0x0D)

Lock/Unlock module buttons

Command: ##(Addr)Rx\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE) Address FF sets all modules with one command
Rx:	Module buttons lock/unlock command 0: buttons unlocked 1: buttons locked
\r:	Carriage return (0x0D)

Response: no response

Example: Command: ##01R0\r
(Module address 0x01, buttons unlocked)

Command: ##01R1\r
(Module address 0x01, buttons locked)

Activate power on default

Command: ##(Addr)P(Range)(LowpassFilter)(Integration)(HighpassFilter)(InputType)(FineTuning)(Button Lock)\r

Command details see also ‘Set module address’

Response: !ACK\r Valid command

Deactivate power on default

Deactivates the power on default mode.
The last pushbutton selected range and filter is adjusted at power on time.

Command: ##(Addr)P\r

##:	Command leading code
Addr:	Module address
P:	Command
\r:	Carriage return (0x0D)

Response: !ACK\r Valid command

Example: Command: ##01P\r

Response: !ACK\r

Read Serial Number

Command: ##(Addr)SETB\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
SETB:	Command
\r:	Carriage return (0x0D)

Response: !(SERAL)xx(REVISION)xxx\r

(SERIAL):	Module serial number (6 digits)
x:	Space
(Revision):	Revision number (i.e. V200)
\r:	Carriage return (0x0D)

DAQP-CHARGE-A

Nr:	Command					Response		Function
1	##	AA	SETD		CR	!AAMMRRFFI{Hp}{icp}{gain}Ok	CR	Set module address
2	??	AA			CR	!AAMMRRFFI{Hp}{icp}{gain}Ok	CR	Read configuration
3	##	AA		RRFFI{Hp}{icp}{gain}k	CR	!ACK	CR	SET configuration
4	##	AA+	R1		CR			Lock buttons
5	##	AA+	R0		CR			Unlock buttons
6	##	AA	SETB		CR	!{16*ASCII}	CR	Read serial number
7	##	AA	P	RRFFI{Hp}{icp}{gain}k	CR	!ACK	CR	Write power on default to module

FF	Lowpass filter
0x00	Full bandwith (50 kHz)
0x01	10 kHz
0x02	3 kHz
0x03	1 kHz
0x04	0.1 kHz

RR	Charge	dB
0x00	0.1 mV/pC	-20
0x01	1 mV/pC	0
0x02	10 mV/pC	20
0x03	100 mV/pC	40
0x04	1000 mV/pC	60

Hp	Highpass filter
0x00	0.1 Hz
0x01	1 Hz
0x02	10 Hz

RR	ICP	dB
0x00	5000 mV	0
0x01	500 mV	20
0x02	50 mV	40
0x03	5 mV	60

AA	Address
AA+	Address
	FF = all modules
MM	Module type (17)
RR	Range
FF	Filter code
k	Button lock
Z	Special module
{gain}	Gain value 3 byte hex
	199 = *1
	8CC = *5
	FFF = *10

I	Integration mode
0	Integration off
1	Single integration
2	Double integration

O	Overflow status
0	No input overflow
1	Input overflow

Programming commands DAQP-CHARGE-B

Instruction set

Command	Syntax
Set module address	##(NewAddr)SETD\r
Read module configuration	??(Addr)\r
Set module configuration	##(Addr)(Range)(LowpassFilter)(InputShort)(HighpassFilter)(FilterType)(Button Lock)\r
Set remote control on/off	##(Addr)R(Button Lock)\r
Module Reset	##(Addr)R6\r
Activate highpass filter	##(Addr)R9\r
Deactivate highass filter	##(Addr)RA\r
Activate power on default	##(Addr)P(Range)(LowpassFilter)(InputShort)(HighpassFilter)(FilterType)(Button Lock)\r
Deactivate power on default	##(Addr)P\r
Read serial number	##(Addr)SETB\r

Commands in detail

Set module address

This command has to be sent while the filter button gets pressed on the module. After that, the new address is stored in an EEPROM and the module sends a response string to the system.

Command: ##(NewAddr)SETD\r

##:	Command leading code
NewAddr:	New module address (2 characters hex from 0x00 to 0xFE)
SETD:	Set address command
\r:	Carriage return (0x0D)

Response: !(NewAddr)(ModuleType)(Range)(LowpassFilter)(InputShort)(HighpassFilter)(FilterType)(Button Lock)(Special)(Isolation)\r

!:	Response leading code	
NewAddr:	Confirmed new module address (2 characters hex)	
ModuleType:	Type of module (2 characters hex, 0x1E according general prog. info)	
Range:	Amplifier range position (2 characters hex, according table)	
LowpassFil.:	Filter position (2 characters hex, according table)	
InputShort.:	Input short circuit (390 kOhm resistor)	
HighpassFil.:	Activate/deactivate highpass filter (1 character, according table)	
FilterType:	Filter characteristic selection (1 character, according table)	
Button lock:	Lock/Unlock module (1 character, according general prog. info)	
Special:	0	Standard module
	1	Special module
InputIsolation:	0	Galvanic isolation
	1	No galvanic isolation

DAQP-CHARGE-B Module

Lowpass filter type

Code	Function
0	Bessel filter activated
1	Butterworth filter activated

Amplifier range and filter selection table

Range	Code	Lowpass	Code
1000 k pC	0x00	100 kHz	0x00
200 k pC	0x01	30 kHz	0x01
40 k pC	0x02	10 kHz	0x02
10 k pC	0x03	3 kHz	0x03
2000 pC	0x04	1 kHz	0x04
500 pC	0x05	300 Hz	0x05
100 pC	0x06	100 Hz	0x06
-	-	30 Hz	0x07
-	-	10 Hz	0x08

Input short circuit

Attention: Activating the input short circuit causes a permanent reset. For discharging the command “reset module” is recommended.

Code	Function
0	Input short circuit deactivated
1	Input short circuit activated

Highpass filter

Code	Function
0	Highpass filter deactivated (DC)
1	Highpass filter activated (AC)

Read module configuration

Command: ??(Addr)\r

??: Command leading code
Addr: Module address (2 characters hex from 0x00 to 0xFE)
\r: Carriage return (0x0D)

Response: !(Addr)(ModuleType)(Range)(LowpassFilter)(InputShortCircuit)(HighpassFilter)(FilterType)(Button Lock)(Special)(Isolation)\r

Response details see also ‘Set module address’

Example: Command: ??(00)\r

Response: !001E010600100

Module address 00; CHARGE-B module; 200 k pC; 100 Hz; no short circuit; highpass filter activated; bessel filter activated; buttons locked; standard module; no galvanic isolation

Set module configuration

Command: ##(Addr)(Range)(LowpassFilter)(InputShortCircuit)(HighpassFilter)(FilterType)(Button Lock)\r

Command details see also ‘Set module address’

Response: !ACK\r (Notice: an incorrect command gets no response from module!)

!: Response leading code
ACK: Acknowledge
\r: Carriage return (0x0D)

Lock/Unlock module buttons

Command: ##(Addr)Rx\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE) Address FF sets all modules with one command
Rx:	Module buttons lock/unlock command 0: buttons unlocked 1: buttons locked
\r:	Carriage return (0x0D)

Response: No response

Example: Command: ##01R0\r
(Module address 0x01, buttons unlocked)

Command: ##01R1\r
(Module address 0x01, buttons locked)

Reset module

Command: ##(Addr)R6\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
	Address FF sets all modules with one command
R6:	Reset the module for 0.5 sec.
\r:	Carriage return (0x0D)

Response: !ACK\r

Activate / deactivate highpassfiter

Command: ##(Addr)Rx\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE) Address FF sets all modules with one command
Rx:	Hihgpassfilter activate / deactivate command 9: Activate highpassfilter A: Deactivate highpassfilter
\r:	Carriage return (0x0D)

Response: No response

Example: Command: ##01R9\r
(Module address 0x01, highpassfilter active)

Command: ##01RA\r
(Module address 0x01, highpassfilter inactive)

Read Serial Number

Command: ##(Addr)SETB\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE) Sending the address FF takes effect on all connected bridge modules.
SETB:	Command
\r:	Carriage return (0x0D)

Response: !(SERAL)xx(REVISION)xxx\r

(SERIAL):	Module serial number (6 digits)
x:	Space
(Revision):	Revision number (i.e. V200)
\r:	Carriage return (0x0D)

Activate power on default

Command: ##(Addr)P(Range)(LowpassFilter)(InputShortCircuit)(HighpassFilter)(FilterType)(Button Lock)\r

Command details see also ‘Set module address’

Response: !ACK\r Valid command

Deactivate power on default

Deactivates the power on default mode.
The last pushbutton selected range and filter is adjusted at power on time.

Command: ##(Addr)P\r

##:	Command leading code
Addr:	Module address
P:	Command
\r:	Carriage return (0x0D)

Response: !ACK\r Valid command

Example: Command: ##01P\r

Response: !ACK\r

Notes

DAQP-CHARGE-B

Nr:	Command					Response		Function
1	##	AA	SETD		CR	!AAMMRRFFECTKZG	CR	Program address
2	??	AA			CR	!AAMMRRFFECTKZG	CR	Read configuration
3	##	AA		RRFFECTR	CR	!ACK	CR	SET configuration
4	##	AA+	R6		CR			Module reset
5	##	AA+	R9		CR			Highpass on
6	##	AA+	RA		CR			Highpass off
7	##	AA+	R1		CR			Lock buttons
8	##	AA+	R0		CR			Unlock buttons
9	##	AA	SETB		CR	!{16*ASCII}	CR	Read serial number
10	##	AA	P	RRFFECTR	CR	!ACK	CR	Write power on default to module

AA	Address
AA+	Address FF = all modules
MM	Module type (1E)
RR	Range
FF	Filter code
E	Input short circuit on/off (390kOHM resistor)
C	Highpass
T	Filter type (Bessl/Buth.)
G	Galvanic isolation deactivate
k	Button lock
Z	Special module

RR	Range
0x00	1000000 pC
0x01	200000 pC
0x02	40000 pC
0x03	10000 pC
0x04	2000 pC
0x05	500 pC
0x06	100 pC

FF	Filter
0x00	100 kHz
0x01	30 kHz
0x02	10 kHz
0x03	3 kHz
0x04	1 kHz
0x05	300 Hz
0x06	100 Hz
0x07	30 Hz
0x08	10 Hz

Programming Commands DAQP-FREQ-A

Instruction Set

Command	Syntax
Set module address	##(NewAddr)SETD\r
Read module configuration	??(Addr)\r
Set module configuration	##(Addr)(Input Range)(Filter)(Button Lock)(AC/DC)(RESPONSE)r
Lock/Unlock module buttons	##(Addr)R(Button Lock)\r
Apply trigger level	##(Addr)V(Trigger)\r
Automatic trigger	##(Addr)T\r
Read serial number	##(Addr)SETB\r
Set module power on default	##(Addr)P)(Input Range)(Filter) (Button Lock)(AC/DC)(RESPONSE)r
Power on default off	##(Addr)P\r

Commands in detail

Set module address

The command has to be send to the module until the push button on the module has been pressed. Then the module sends the answer.

Command: ##(Addr)SETD\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
\r:	Carriage return (0x0D)

Response: !(Addr)21(Input Range)(Filter)(Button Lock)(AC/DC)(Response)(Special)(TRIGGER)\r

!:	Response leading code	
Addr:	Acknowledged new module address	
1A:	Module Type (DAQP-FREQ-A)	
Input Range:	Current measuring range (2 characters hex, according to table)	
Filter:	Current low pass filter (2 characters hex, according to table)	
Button lock:	0	Buttons unlocked
	1	Buttons locked
(AC/DC)	0	AC Coupling
	1	DC Coupling
Response:	0	Slow
	1	Fast
Special:	0	Standard Module
	1	Special Module
Trigger:	Trigger level in (5*Characters Hex)mV	
	0..120V	

Example: Command: ##01SETD\r

Response: !011A02020100F000\r

(DAQP-FREQ-A;20kHz;20kHz;Buttons activated; DC; Standard Module ;6144 0mV Triggerlevel)

DAQP-FREQ-A Module

Range and filter selection table

Range	Code	Filter	Code
200 kHz	0x00	200 kHz	0x00
100 kHz	0x01	100 kHz	0x01
20 kHz	0x02	20 kHz	0x02
5 kHz	0x03	5 kHz	0x03
1 kHz	0x04	1 kHz	0x04
100 Hz	0x05	100 Hz	0x05

Read Module Configuration

Command: ??(Addr)\r

??:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
\r:	Carriage return (0x0D)

Response: !(Addr)21(Input Range)(Filter) (Button Lock)(AC/DC)(Response)(Special)(TRIGGER)\r

!:	Response leading code	
Addr:	Acknowledged new module address	
21:	Module Type (DAQP-FREQ-A)	
Input Range:	Current measuring range (2 characters hex, according to table)	
Filter:	Current low pass filter (2 characters hex, according to table)	
Button lock:	0	Buttons unlocked
	1	Buttons locked
(AC/DC)	0	AC Coupling
	1	DC Coupling
Response:	0	Slow
	1	Fast
Special:	0	Standard Module
	1	Special Module
Trigger:	Trigger level in (5*Characters Hex)mV	
	0..120V	
/r	Carriage return (0x0D)	

Set Module Configuration

Command: ##(Addr)(Input Range)(Filter)(Button Lock)(AC/DC)(RESPONSE)r

Addr:	Acknowledged new module address	
21:	Module Type (DAQP-FREQ-A)	
Input Range:	Current measuring range (2 characters hex, according to table)	
Filter:	Current low pass filter (2 characters hex, according to table)	
Button lock:	0	Buttons unlocked
	1	Buttons locked
(AC/DC)	0	AC Coupling
	1	DC Coupling
Response:	0	Slow
	1	Fast
Special:	0	Standard Module
	1	Special Module
Trigger:	Trigger level in (5*Characters Hex)mV	
	0..120V	
/r	Carriage return (0x0D)	

Response:	!(Addr)\r	Valid command
	?(Addr)\r	Invalid command, syntax error

Lock/Unlock module buttons

Command: ##(Addr)R(Button Lock)\r

##:	Command leading code	
Addr:	Module address (2 characters hex from 0x00 to 0xFE)	

Sending the Address FF takes affect on all connected FREQ-A modules.

R:	Command	
Button lock:	0	buttons unlocked
	1	buttons locked
\r:	Carriage return (0x0D)	

Response: There is no response on this command.

Apply a Trigger Level

Command: ##(Addr)V(Trigger)\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)

Sending the Address FF takes affect on all connected FREQ-A modules.

V:	Command
Trigger:	Trigger level in (5*Characters Hex)mV 0..120V
\r:	Carriage return (0x0D)

Response: There is no response on this command.

Automatic Trigger

Command: ##(Addr)T\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)

Sending the Address FF takes affect on all connected FREQ-A modules.

T:	Command
\r:	Carriage return (0x0D)

Response: There is no response on this command.

Read Serial Number

Command: ##(Addr)SETB\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)

Sending the Address FF takes effect on all connected Bridge modules.
Address FF takes effect on all DAQP-BRIDGE-A modules in the system.

SETB	Command
\r:	Carriage return (0x0D)

Response: !(SERIAL)xx(REVISION)xxx\r

(SERIAL):	Module Serial Number (6 digits)
x:	Space
(Revision):	Revision Number V200
\r:	Carriage return (0x0D)

Set Module Power On Default

Command:	##(Addr)P(Input Range)(Filter) (Button Lock)(AC/DC)(RESPONSE)\r		
	Addr:	Acknowledged new module address	
	1A:	Module Type (DAQP-FREQ-A)	
	Input Range:	Current measuring range (2 characters hex, according to table)	
	Filter:	Current low pass filter (2 characters hex, according to table)	
	Button lock:	0	Buttons unlocked
		1	Buttons locked
	(AC/DC)	0	AC Coupling
		1	DC Coupling
	Response:	0	Slow
		1	Fast
	Special:	0	Standard Module
		1	Special Module
	Trigger:	Trigger level in (5*Characters Hex)mV 0..120V	
	\r:	Carriage return (0x0D)	
Response:	!(Addr)\r	Valid command	
	?(Addr)\r	Invalid command, syntax error	
Example:	Command:	##01P02020100F000\r	
	Response:	!ACK\r	

(Set the module to 20kHz;20kHz;Buttons activated; DC; Standard Module; 6144 0mV; at Power on)

Power On Default Off

Deactivates the Power on default mode.

The last Pushbutton selected range and Filter is adjusted at power on time.

Command:	##(Addr)P\r	
	##:	Command leading code
	Addr:	Acknowledged new module address
	P:	Command
	\r:	Carriage return (0x0D)
Response:	!(Addr)\r	Valid command
	?(Addr)\r	Invalid command, syntax error
Example:	Command:	##01P\r
	Response:	!ACK\r

Programming Commands DAQP-MULTI/-THERM

Instruction Set

Command	Syntax
Set module address	##(NewAddr)SETD\r
Read module configuration	??(Addr)\r
Set module configuration	##(Addr)(InputRange)(Filter)(Excitation)(Mode)(ModulType)(FilterType)(OutputType)\r
Set Baud rate for RS-485	##(Addr)BAUDRATEu\r
Read serial number	##(Addr)SETB\r
Write text to module	##(Addr)SETWn[TEXT]\r
Read text from module	##(Addr)SETRn\r
Read module type and software revision	??(Addr)VER\r
Set power-On default	##(Addr)SETPO\r
Read module status	##(Addr)FSTATE\r
Clear module status	##(Addr)FCLEAR\r
Automatic Sensor Offset Adjustment	##(Addr)R7\r
Read sensor offset adjustment settings	##(Addr)SETV\r
Set sensor offset manually	##(Addr)SETS+-AAAAAA\r
Set cal signal ON/OFF (80% output)	##(Addr)Rc\r
Set cal signal ON/OFF (0 V output)	##(Addr)Rc\r
Set excitation ON/OFF	##(Addr)Rc\r
Input short circuit on/off	##(Addr)Rc\r
Internal amplifier zero	##(Addr)R8\r
Read measurement results over RS485	##(Addr)A\r
Set custom excitation current and min. / max. output ranges	##(Addr)SETXABBBBBBACCCCCCEEEE\r
Read adjusted excitation current and min. / max output range values	??(Addr)SETX\r
Read custom excitation current (µA)	??(Addr)SETXF\r
Set current, custom resistor and custom R0 values for Ptxxxx type	##(Addr)SETTxxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx yyyyii yyyyii zzzz zzzz\r
Read current, custom resistor and custom R0 values of Ptxxxx type	??(Addr)SETT\r
Set linear correction ON/OFF	##(Addr)LCAENAx\r
Get custom linearization table header	??(Addr)GETCUSTx\r

DAQP-MULTI/-THERM Module

Range selection table

Range[1]	Code	Range[2]	Code	Range[3]	Code	Range[4]	Code	Range[5]	Code
±5 V	0x00	5000 mV/mA	0x00	1 MΩ	0x00	Max. range	0x00	Max. range	0x00
±2 V	0x01	2000 mV/mA	0x01	300 kΩ	0x01	±100 °C	0x01	±100 °C	0x01
±1 V	0x02	1000 mV	0x02	100 kΩ	0x02	-100 .. 200 °C	0x02	-100 .. 200 °C	0x02
±0.5 V	0x03	5000 mV	0x03	30 kΩ	0x03	0 .. 100 °C	0x03	0 .. 100 °C	0x03
±0.2 V	0x04	200 mV	0x04	10 kΩ	0x04	0 .. 200 °C	0x04	0 .. 200 °C	0x04
±0.1 V	0x05	100 mV	0x05	3 kΩ	0x05	0 .. 600 °C	0x05	0 .. 600 °C	0x05
±0.05 V	0x06	50 mV	0x06	1 kΩ	0x06	-	0x06	-	0x06
±0.02 V	0x07	20 mV	0x07	300 Ω	0x07	-	0x07	-	0x07
±0.01 V	0x08	10 mV	0x08	100 Ω	0x08	-	0x08	-	0x08
±0.005 V	0x09	5 mV	0x09	30 Ω	0x09	-	0x09	-	0x09
-	0x0A	2 mV	0x0A	10 Ω	0x0A	-	0x0A	-	0x0A
-	0x0B	1 mV	0x0B	3 Ω	0x0B	-	0x0B	-	0x0B
-	0x0C	0.5 mV	0x0C	1 Ω	0x0C	-	0x0C	-	0x0C
-	0x0D	-	0x0D	-	0x0D	-	0x0D	-	0x0D
-	0x0E	-	0x0E	-	0x0E	-	0x0E	-	0x0E
CUSTOM	0x0F	CUSTOM	0x0F	CUSTOM	0x0F	CUSTOM	0x0F	CUSTOM	0x0F

[1] in V @ Voltage mode

[2] in mV/mA @ Bridge with constant current

[3] in Ω @ Resistor mode

[4] in °C @ Ptxxxx mode

[5] in °C @ Thermocouple mode

Filter selection table

Filter (order)[1]	Code	Filter (cutoff)[2]	Code
2nd	0x00	3 kHz	0x00
4th	0x01	1 kHz	0x01
6th	0x02	300 Hz	0x02
8th	0x03	100 Hz	0x03
-	0x04	30 Hz	0x04
-	0x05	10 Hz	0x05
-	0x06	3 Hz	0x06

[1] MSB (e.g.: 0x14 -> 4th order)

[2] LSB (e.g.: 0x14 -> 30 Hz)

Excitation

Code	Excitation[1]	Code	Excitation[2]
0x00	0 mA	0x00	Fixed
0x01	0.5 mA	0x0F	CUSTOM
0x02	1 mA	Code	Excitation[3]
0x03	2 mA	0x00	Fixed
0x04	4 mA	0x0F	CUSTOM
0x0F	CUSTOM	Code	Excitation[4]

[1] Bridge with constant current

[2] Resistor mode

[3] Ptxxxx mode

[4] Thermocouple mode

0x00	Sensor break detecion OFF
0x01	Sensor break detecion ON

Module type and baud rate

Module type	Code	Baud rate	Code
Voltage	0x00	-	0x00
Bridge I	0x01	1200	0x01
Ohm	0x02	2400	0x02
Ptxxxx	0x03	4800	0x03
Thermocouple	0x04	-	0x04
-	0x05	9600	0x05
-	0x06	14400	0x06
-	0x07	19200	0x07
-	0x08	28800	0x08
-	0x09	-	0x09
-	0x0A	38400	0x0A
-	0x0B	57600	0x0B
-	0x0C	115200	0x0C

Ptxxxx & thermocouple selection table

Code	Mode[1]	Code	Mode[2]
0x00	Pt100	0x00	Type K
0x01	Pt200	0x01	Type J
0x02	Pt500	0x02	Type T
0x03	Pt1000	0x03	Type R
0x04	Pt2000	0x04	Type S
0x05	Pt13900	0x05	Type N
0x06	CUSTOM1	0x06	Type E
0x07	CUSTOM2	0x07	Type B
0x08	-	0x08	Type L
0x09	-	0x09	Type C
0x0A	-	0x0A	Type U
0x0B	-	0x0B	CUSTOM

[1] Ptxxxx temperature

[2] Thermocouple type

Commands in detail

Set module address

Receiving this command has no effect, unless the module's push-button is pressed. In this case, the module is permanently (EEPROM) set to the address included in the command.

Command: ##(Addr)SETD\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
\r:	Carriage return (0x0D)

Response: !(Addr)XX(InputRange)(Filter)(Excitation)(Mode)(ModulType)(FilterType)(OutputType)\r

!:	Response leading code	
Addr:	Acknowledged new module address	
XX:	Module type (DAQP-THERM = 0x37, DAQP-MULTI = 0x38)	
InputRange:	Measuring ranges (1 characters hex, according to table)	
Filter:	Filter order and cut-off (2 characters hex, according to table)	
Excitation:	Excitation types (1 character hex, according to table)	
Mode:	Measuring types (1 character hex, according to table)	
ModulType:	Modul type (1 character hex, according to table)	
FilterType:	0	Butterworth
	1	Bessel
OutputType:	0	±5 V
	1	0 .. 5 V

Example: Command: ##01SETD\r
Response: !013821403300\r

(Modul address: 1; DAQP-MULTI; -100 .. 200 °C; 4th order; 30 Hz; Fixed; Pt1000; Ptxxxx; Butterworth; ±5 V)

Set module configuration

Command: ##(Addr)(InputRange)(Filter)(Excitation)(Mode)(ModulType)(FilterType)(OutputType)\r

##:	Command leading code	
Addr:	Acknowledged new module address	
InputRange:	Measuring ranges (1 characters hex, according to table)	
Filter:	Filter order and cut-off (2 characters hex, according to table)	
Excitation:	Excitation types (1 character hex, according to table)	
Mode:	Measuring types (1 character hex, according to table)	
ModulType:	Modul type (1 character hex, according to table)	
FilterType:	0	Butterworth
	1	Bessel
OutputType:	0	±5 V
	1	0 .. 5 V
\r:	Carriage return (0x0D)	

Response:

!(Addr)ACK\r	Valid command
!(Addr)NOACK\r	Wrong paramter or communication between PIC fails or custom range/ excitation is selected but not defined.

Read module configuration

This command requests the current module configuration stored in RAM.
Note: If the actual module configuration has not been stored with the “SETPO” command, the EEPPROM can hold another module configuration than the one read with this command.

Command: ??(Addr)\r

??:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
\r:	Carriage return (0x0D)

Response: !(Addr)XX(InputRange)(Filter)(Excitation)(Mode)(ModulType)(FilterType)(OutputType)\r

!:	Response leading code	
Addr:	Acknowledged new module address	
XX:	Module type (DAQP-THERM = 0x37, DAQP-MULTI = 0x38)	
InputRange:	Measuring ranges (1 characters hex, according to table)	
Filter:	Filter order and cut-off (2 characters hex, according to table)	
Excitation:	Excitation types (1 character hex, according to table)	
Mode:	Measuring types (1 character hex, according to table)	
ModulType:	Modul type (1 character hex, according to table)	
FilterType:	0	Butterworth
	1	Bessel
OutputType:	0	±5 V
	1	0 .. 5 V
\r:	Carriage return (0x0D)	

Set baud rate for RS485

This command sets the desired baud rate for the module. The response to the command is still sent with the old baud rate.

Command: ##(Addr)BAUDRATEu\r

##:	Command leading code
Addr:	Acknowledged new module address
u:	Baud rate (1 character hex, according to table)

Response:	!(Addr)ACK\r	Valid command
	!(Addr)NOACK\r	Wrong paramter or communication failed.

Read Serial Number

Command: ##(Addr)SETB\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
SETB	Command
\r:	Carriage return (0x0D)

Response: !nnnnnnnnnnnnnnnn\r

!:	Command leading code
n:	Module Serial Number (16 digits hex)
\r:	Carriage return (0x0D)

Write text to module

This command writes to one of the 15 sub areas of EEPROM sector “TEXT”. Each sub area can be filled with text, comments, of up to 80 characters (80 Byte ASCII hex).

Command: ##(Addr)SETWn[TEXT]\r

##:	Command leading code
Addr:	Acknowledged new module address
SETW:	Command
n:	Storing address (area) from 0x00 to 0x0F
	0x00 .. 0x0E: Strings for internal use
	0x0F: Calibration date (reserved)
[TEXT]:	Character string up to 80 Bytes ASCII hex.
	Only the Bytes transmitted are stored. No fill up to 80 chars.
\r:	Carriage return (0x0D)

Response:

!(Addr)ACK\r:	Valid command
!(Addr)NOACK\r:	Wrong paramter or communication failed.

Read text from module

This command reads the content of one of the 15 sub areas from EEPROM sector “TEXT”. One area can contain up to 80 characters of text.

Command: ##(Addr)SETRn\r

##:	Command leading code
Addr:	Acknowledged new module address
SETR:	Command
n:	Reading address (area) from 0x00 to 0x0F
	0x00 .. 0x0E: Strings for internal use
	0x0F: Calibration date (reserved)
	Only the Bytes transmitted are stored. No fill up to 80 chars.
\r:	Carriage return (0x0D)

Response:

![TEXT]\r:	The text from the associated page address. Up to 80 Bytes ASCII hex. Only the count of stored Bytes is transmitted
!(Addr)NOACK\r:	Wrong or undefined command or communication failed.

Read module type and software revision

This command reads the module type and the software revision.

Command: ??(Addr)VER\r

??:	Command leading code
Addr:	Acknowledged new module address
VER:	Command
\r:	Carriage return (0x0D)

Response: !mVer. x.xx z.zz\r:

!:	Response leading code
m:	Module name (max. 16 characters, e.g. DAQP-MULTI)
Ver.:	Version string (4 characters hex)
Blank:	1 character hex
x.xx:	Software revision of NIOSII program; stored in source code (4 characters)
Blank:	1 character hex
z.zz:	FPGA hardware revision; stored in VHDL code (4 characters)
\r:	Carriage return (0x0D)

!(Addr)NOACK\r:	Wrong or undefined command or communication failed.

Set power on default

This command stores all configuration data held in RAM to EEPROM.

Command: ##(Addr)SETPO\r

##:	Command leading code
Addr:	Acknowledged new module address

Sending the Address FF takes effect on all connected DAQP-MULTI and on all DAQP-BRIDGE-B modules in the system.

SETPO:	Command
\r:	Carriage return (0x0D)

There is no response on this command if address was set to 0xFF!

Response:

!ACK\r:	Valid command
!NOACK\r:	Wrong or undefined command or communication failed.

Read module status

Check, read, the module's error message. If any part of the module generates fail behaviour, this is immediately reported to a reserved area in the EEPROM. With this command the cause for the error can be identified.

Command: ##(Addr)FSTATE\r

##:	Command leading code
Addr:	Acknowledged new module address
FSTATE:	Command
\r:	Carriage return (0x0D)

Response:

!--No ERRORS reported--\r:
!--ERROR: I2C TIMEOUT\r:

!:	Response leading code
Error Message:	20 .. 40 ASCII hex characters. Lenght depending on error message.
\r:	Carriage return (0x0D)
!(Addr)NOACK\r:	Wrong or undefined command or communication failed.

Clear module status

Reset the module and thus an existing Module Error Statement.
Note: After the FAIL has been cleared, the module reboots. If the FAIL still exists after reboot, the module is likely damaged and should be serviced.

Command: ##(Addr)FCLEAR\r

##:	Command leading code
Addr:	Acknowledged new module address
FCLEAR:	Command
\r:	Carriage return (0x0D)

Response: !(Addr)ACK\r: Valid command.

Automatic sensor offset adjustment

This command compensates the offset of a connected sensor. There is no response to the command if the address is set to “0xFF”. The Module is approximately 2 seconds off-line after sending this command. That means it can’t receive any command. The sensor offset adjustment is only performed, if the module is in bridge mode.

Command: ##(Addr)R7\r

##:	Command leading code
Addr:	Acknowledged new module address

Sending the Address FF takes effect on all connected modules in the system.

R7:	Command
\r:	Carriage return (0x0D)

There is no response on this command if address was set to 0xFF!

Response:

!ACK\r:	Valid command
!NOACK\r:	Wrong or undefined command or communication failed.

Read sensor offset adjustment settings

This command is only executable in Bridge I Mode. It can be used after a „Sensor offset adjustment“, „Set sensor offset“ and „Amplifier zero“ command.

In conjunction with the „Sensor offset adjustment“ the SETV command reads back the input signal value and the adjusted current excitation at which the „Sensor offset adjustment“ was executed.

The sign of the read back signal value is changed because of a DEWESoft (measurement software) issue.

E.g.: Module input -15 mV leads to +15 mV@SETV [!+003A98xxxx\r]

Using the SETV command after „Set sensor offset“, it reads back the value which was set by the „Set sensor offset“ and of course the adjusted excitation current which the „Set sensor offset“ command was executed.

Executing the SETV command after „Internal amplifier zero“ leads to a value of „0“ [!+000000\r].

- Resolution of the input signal value is µV
- Resolution of the excitation value is µA

Command: ##(Addr)SETV\r

##:	Command leading code
Addr:	Acknowledged new module address
SETV:	Command
\r:	Carriage return (0x0D)

Response: !SAAAAAACCCC\r

!:	Response leading code
S:	Value sign (1 Byte hex: ‚+‘ or ‚-‘)
AAAAAA:	Signal value in µV resolution (6 Byte ASCII hex)
CCCC:	Excitation value in µA resolution (4 Byte ASCII hex)
\r:	Carriage return (0x0D)

!NOACK\r:	Wrong or undefined command or communication failed.

Set sensor offset manually

This command is only executable in Bridge I Mode. It allows setting of a Sensor Offset for the incoming signal. The unit for the sensor offset is µV/mA.
Note: Available since firmware version 1.80

Command: ##(Addr)SETS+-AAAAAA\r

##:	Command leading code
Addr:	Acknowledged new module address
SETS:	Command
A:	Sensor offset (6 Byte ASCII hex; representing the sensor offset in µV/mA)
\r:	Carriage return (0x0D)

Response:

!ACK\r:	Valid command
!NOACK\r:	Wrong or undefined command or communication failed.

Example: Setting +10 mV offset @ 1 mA Excitation: ##aaSETS+002710\r
Setting +10 mV offset @ 2 mA Excitation: ##aaSETS+004E20\r
Setting -10 mV offset @ 1 mA Excitation: ##aaSETS-00271\r
Setting -5 mV offset @ 1 mA Excitation: ##aaSETS-001388\r

Set CAL-signal ON/OFF (80% output)

The module is either adjusted to 4 V_{DC} (80% of output) on output stage, or the CAL-signal (80% of output) is released to its previous value. This adjust is independent of any ADC adjust.
Note: Sending any configuration command to the module, clears the CAL-signal.

Command: ##(Addr)Rc\r

##:	Command leading code
Addr:	Acknowledged new module address
R:	Command
c:	Calibration signal ON/OFF (1 Byte ASCII hex) D: switching calibration signal ON E: switching calibration signal OFF
\r:	Carriage return (0x0D)

There is no response on this command if address was set to 0xFF!

Response:

!ACK\r:	Valid command
!NOACK\r:	Wrong or undefined command or communication failed.

Set CAL-signal ON/OFF (0 V output)

The module is either adjusted to 0 V_{DC} on output stage or CAL-signal is released to its previous value. This adjust is independent of any ADC adjust.
Note: Sending any configuration command to the module, clears the CAL-signal.

Command: ##(Addr)Rc\r

##:	Command leading code
Addr:	Acknowledged new module address
R:	Command
c:	Calibration signal ON/OFF (1 Byte ASCII hex) A: switching calibration signal ON B: switching calibration signal OFF
\r:	Carriage return (0x0D)

There is no response on this command if address was set to 0xFF!

Response:

!ACK\r:	Valid command
!NOACK\r:	Wrong or undefined command or communication failed.

Set excitation ON/OFF

This command enables or disables the adjusted current excitation.

Note: DAQP-MULTI thermocouple mode -> Command not allowed!
DAQP-THERM -> Command not allowed!

Command: ##(Addr)Rc\r

##:	Command leading code
Addr:	Acknowledged new module address
R:	Command
c:	Excitation ON/OFF (1 Byte ASCII hex) G: switching excitation ON F: switching excitation OFF
\r:	Carriage return (0x0D)

There is no response on this command if address was set to 0xFF!

Response:

!ACK\r:	Valid command
!NOACK\r:	Wrong or undefined command, communication failed or module type doesn't support this command.

Input short circuit ON/OFF

This command either shorts or releases a shortcut on the modules input stage.

Note: DAQP-MULTI thermocouple mode -> Command not allowed!
DAQP-THERM -> Command not allowed!

Command: ##(Addr)Rc\r

##:	Command leading code
Addr:	Acknowledged new module address
R:	Command
c:	Short ON/OFF (1 Byte ASCII hex) 2: Execute short on input 3: Release short on input
\r:	Carriage return (0x0D)

There is no response on this command if address was set to 0xFF!

Response:

!ACK\r:	Valid command
!NOACK\r:	Wrong or undefined command, communication failed or module type doesn't support this command.

Internal amplifier zero

This command compensates a offset at input ADC side.

Note: The command takes about 2.7 seconds to finish.
Not executable when in THERM mode or with DAQP-THERM module.

Command: ##(Addr)R8\r

##:	Command leading code
Addr:	Acknowledged new module address
R8:	Command
\r:	Carriage return (0x0D)

There is no response on this command if address was set to 0xFF!

Response:

!ACK\r:	Valid command
!NOACK\r:	Wrong or undefined command, communication failed or module type doesn‘t support this command.

Read measurement results over RS485/RS232

This command reads back the measurement results over the serial interface.
Note: Due to baud rate limitations the measurement results on host side are updated slower than with AD hardware.

Command: ##(Addr)A\r

##:	Command leading code
Addr:	Acknowledged new module address
A:	Command
\r:	Carriage return (0x0D)

There is no response on this command if address was set to 0xFF!

Response: !SD.DDDDDDDDE SD\r

!:	Response leading code
S:	Sign Byte (1 Byte hex: ‚+' or ‚-')
D:	ASCII decimal value from 0-9
E:	Exponential
\r:	Carriage return (0x0D)

!NOACK\r: Wrong or undefined command or communication failed.

Set custom excitation current and min. / max. output ranges

This command transfers the min. / max. output range values and the custom excitation current value to the modules RAM. In order to store the values to the EEPROM, the "SETPO" command has to be used afterwards. If the module configuration is adjusted to custom current, the received custom excitation current is adjusted immediately. Otherwise the already adjusted, in module setup defined current value, stays adjusted. The resolution for the custom current is µA.

The resolution of the min. and max. values differs between the module modes.

- Therm and Ptxxxx mode: resolution in mK
- Ohm mode: resolution in 100mΩ
- Voltage mode: resolution in µV
- Bridge I mode: resolution in µV/1mA

Command: ##(Addr)SETXABBBBBBACCCCCCEEEE\r

##:	Command leading code
Addr:	Acknowledged new module address
SETX:	Command
A:	Value sign (1 Byte hex: ‚+' or ‚-')
B:	Min. output range value (6 characters hex, resolution in µV, µV/1mA, mK or 100mΩ)
C:	Max. output range value (6 characters hex, resolution in µV, µV/1mA, mK or 100mΩ)
E:	Excitation current (4 characters hex, resolution in µA)
\r:	Carriage return (0x0D)

Response:

!(Addr)ACK\r:	Valid command
!(Addr)NOACK\r:	Wrong or undefined command or communication failed. Current is out of range (> 5 mA). Range too high.

Example:

Desired custom range:	-1500 mV/mA .. +1500 mV/mA and 2 mA excitation
Command to module:	##aaSEXTX-16E360+16E36007D0\r

Read adjusted excitation current and min. / max. output range values

This command reads the adjusted excitation current and the min. / max. output range values.

The resolution of the min. and max. values differs between the module modes.

- Therm and Ptxxxx Mode: resolution in mK
- Ohm mode: resolution in 100mΩ
- Voltage mode: resolution in µV
- Bridge I mode: resolution in µV/1mA

Command: ??(Addr)SETX\r

??:	Command leading code
Addr:	Acknowledged new module address
SETX:	Command
\r:	Carriage return (0x0D)

Response: !sAAAAAAsBBBBBBEEEE\r

!:	Response leading code
s:	Sign Byte (1 Byte hex: ‚+' or ‚-')
A:	Min. output range value (6 characters hex, resolution in µV, µV/1mA, mK or 100mΩ)
B:	Max. output range value (6 characters hex, resolution in µV, µV/1mA, mK or 100mΩ)
E:	Excitation current (4 characters hex, resolution in µA)
\r:	Carriage return (0x0D)

!(Addr)NOACK\r: Wrong or undefined command or communication failed.

Read custom excitation current (µA)

This command reads the custom excitation current stored in the module. It can differ from the actual adjusted excitation current.

Command: ##(Addr)SETXF\r

##:	Command leading code
Addr:	Acknowledged new module address
SETXF:	Command
\r:	Carriage return (0x0D)

Response: !nnnn\r

!:	Response leading code
nnnn:	Current value (4 characters hex, resolution in µA)
\r:	Carriage return (0x0D)

!(Addr)NOACK\r: Wrong or undefined command or communication failed.

Set current, custom resistor and custom R0 values for Ptxxxx type

This command stores the current values for all Ptxxxx types. Additionally also all custom adjusts are stored. If the custom adjusts are not used they should be set to “0”.

Command: ##(Addr)SETTxxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx yyyyii yyyyii zzzz zzzz\r

Pt100 Pt200 Pt500 Pt1000 Pt2000 Pt10039 PtCust1 PtCust2 R_Cust1 R_Cust2 R0_Cust1 R0_Cust2

##:	Command leading code
Addr:	Acknowledged new module address
SETT:	Command
xxxx:	Current value. Starting with current for Pt100 to current for PtCust2 (8 x 4 Byte ASCII hex)
yyyyii:	Custom resistor value. „yyyy“ is the value and „ii“ the exponent -> yyyy*e^{ii} e.g.: 100002 = 1000*e^{02} -> 100000 (2 x 6 Byte ASCII hex)
zzzz:	Custom R0 value (2 x 4 Byte ASCII hex for R0 values of Cust1 and Cust2 Ptxxxx linearization tables).
\r:	Carriage return (0x0D)

Response:

!(Addr)ACK\r:	Valid command
!(Addr)NOACK\r:	Wrong or undefined command, communication failed or module type doesn‘t support this command.

Example: Storing only default current values:

##aaSETT	xxxx (6x)	00000000	000000000000	00000000\r
	fixed currents	cust. currents	cust resistor values	custom R0 values

Read current, custom resistor and custom R0 values for Ptxxxx type

This command reads the fixed and custom current values, custom resistor values (linearization table res.) and custom R0 values for all Ptxxxx types.

Command: ??(Addr)SETT\r

??:	Command leading code
Addr:	Acknowledged new module address
SETT:	Command
\r:	Carriage return (0x0D)

Response: !(Addr)SETTxxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx yyyyii yyyyii zzzz zzzz\r

Pt100 Pt200 Pt500 Pt1000 Pt2000 Pt10039 PtCust1 PtCust2 R_Cust1 R_Cust2 R0_Cust1 R0_Cust2

!:	Response leading code
Addr:	Acknowledged new module address
xxxx:	Current value. Starting with current for Pt100 to current for PtCust2 (8 x 4 Byte ASCII hex)
yyyyii:	Custom resistor value. „yyyy“ is the value and „ii“ the exponent -> yyyy*e^{ii} e.g.: 100002 = 1000*e^{02} -> 100000 (2 x 6 Byte ASCII hex)
zzzz:	Custom R0 value (2 x 4 Byte ASCII hex for R0 values of Cust1 and Cust2 Ptxxxx linearization tables).
\r:	Carriage return (0x0D)

!(Addr)NOACK\r:	Wrong or undefined command, communication failed or module type doesn‘t support this command.

Set linear correction ON/OFF

This command enables or disables the linear correction for the digital data.

Note: This change is only temporary and is overwritten when next configuration command is sent.

Command: ##(Addr)LCAENAx\r

##:	Command leading code
Addr:	Acknowledged new module address
LCAENA:	Command
x:	Enable Bit (1 Byte ASCII hex) 0: linear correction is disabled (bypassed) 1: linear correction is enabled
\r:	Carriage return (0x0D)

Response:

!(Addr)ACK\r:	Valid command
!(Addr)NOACK\r:	Wrong or undefined command, communication failed or module type doesn‘t support this command.

Get custom linearization table header

This command reads back specific information of every custom linearization table.

- Min. temperature value
- Max. temperature value
- Linearization table name

Command: ??(Addr)GETCUSTx\r

??:	Command leading code
Addr:	Acknowledged new module address
GETCUST:	Command
x:	Table selector (1 Byte ASCII hex) x = 1: Pt Custom1 x = 2: Pt Custom2 x = 3: Thermocouple custom1
\r:	Carriage return (0x0D)

Response: !aaaabbbbccccccccccccccccACK\r

!:	Response leading code
aaaa:	Min. temperatur value (4 Byte ASCII hex)
bbbb:	Max. temperature value (4 Byte ASCII hex)
c:	Linearization table name (up to 16 Byte ASCII hex)
\r:	Carriage return (0x0D)

!(Addr)NOACK\r: No custom table for this table selector or wrong command.

Notes

Programming commands DAQN-OHM, -POT and -RTD

Instruction set

Command	Syntax
Set module address	##(NewAddr)SETD\r
Read serial number	##(Addr)SETB\r

Commands in detail

Set module address

This command has to be sent as long as the filter button is pressed on the module. After that, the new address is stored in an EEPROM and the module sends a response string to the system.

Command: ##(NewAddr)SETD\r

##:	Command leading code
NewAddr:	New module address (2 characters hex from 0x00 to 0xFE)
SETD:	Set address command
\r:	Carriage return (0x0D)

Response: !(NewAddr)(ModuleType)\r

!:	Response leading code
NewAddr:	Confirmed new module address (2 characters hex)
ModuleType:	Type of module (2 characters hex, xx according to used module)

Example: Command: ##01SETD\r

Response: !01xx\r

xx=0x03 for DAQN-RTD
xx=0x01 for DAQN-POT
xx=0x19 for DAQN-OHM

(Address 0x01, DAQN-xxx)

Read serial number

Command: ##(Addr)SETB\r

Command details see also ‘Set module address’

Response: !(Serial)xx(Revision)xxx\r

(Serial):	Module Serial Number (6 digits)
x:	Space
(Revision):	Revision Number (i.e. V200)
\r:	Carriage return (0x0D)

Notes

Module commands

Command summary for PAD-V8-P module

Command	Syntax
GENERAL COMMANDS	
Set module configuration	**%(OldAddr)(NewAddr)(InputRange)(BaudRate)(Format)\r**
Set module address	**##(Addr)SETP\r**
Read module channel configuration	??(Addr)\r
Read module type and firmware version	??(Addr)VER\r
Read firmware version number	$(Addr)F\r
Read module name	$(Addr)M\r
Read serial number	??(Addr)SNR\r
FUNCTIONAL COMMANDS	
Set input range for channel x	**%(Addr)T(ChannelNo)(InputRange)\r**
Read analog input from channel x	**#(Addr)(ChannelNo)\r**
Read all 8 channel data values	**$(Addr)A\r**
Synchronized sampling	#**\r
Read all internally stored values	$(Addr)S\r
Read output range of channel x	$(Addr)W(ChannelNo)\r
Duration of averaging	%(Addr)M(AverageLength)\r
Set LED state	$(Addr)L(State)\r
Read LED and digital input state	$(Addr)I\r
Read module configuration	$(Addr)2\r
Read channel multiplex state	$(Addr)6\r
Set channel multiplex state	$(Addr)5(Channels)\r
Zero calibration for channel x	%(Addr)Z(ChannelNo)\r
Zero calibration for all channels	$(Addr)1\r (only modules older rev. 1.14)
Span calibration for channel x	%(Addr)S(ChannelNo)\r
Span calibration for all channels	$(Addr)0\r (only modules older rev. 1.14)
Open memory area for custom linearisation	##(Addr)OK(Number)\r
Set custom linearisation point	##(Addr)T(PointNo)(Sign)(Voltage)(Sign)(Value)\r
Close memory area for custom linearisation	##(Addr)C(InputRangeCode)\r

Set module configuration

Command: %(OldAddr)(NewAddr)(InputRange)(BaudRate)(DataFormat)\r

%	Command leading code
(OldAddr)	The original / old module address. The factory default address of a module is 00. The address range is 0x00 to 0xFE.
(NewAddr)	The new module address (0x00 to 0xFE - be careful to get no conflict with existing module addresses).
(InputRange)	Define the analog input range, refered to the table *input ranges*.
(BaudRate)	Define communication baud rate, recommended '06' hex (= 9600 bps).
(DataFormat)	Define checksum and output data format, refered to the table *data format*

Response:

!(Addr)<CR>	The command to this address is valid.
?(Addr)<CR>	The command is invalid, parameter values are invalid or change settings not possible.

Example: Command: %0130080600<CR>

%:	command leading code
01:	old module address
30:	new module address (0x30 = 48 dez)
08:	voltage input ±10 V
06:	baud rate (9600)
00:	data format (engineering units, checksum disabled)
<CR>:	carriage return

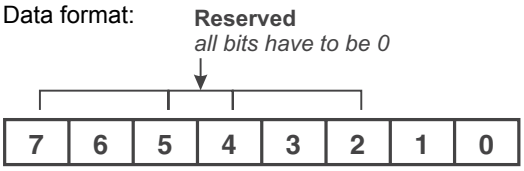
Response !30<CR>
Address successfully changed from 01 to 30.

Input ranges:

Code (hex)	Input range	Resolution	Precision mode	Digits
02	±100 mV	100 uV	10 uV	8 or 9
03	±500 mV	100 uV	10 uV	8 or 9
04	±1 V	100 uV	10 uV	8 or 9
05	±2.5 V	100 uV	10 uV	8 or 9
08	±10 V	100 uV	10 uV	8 or 9
09	±5 V	100 uV	10 uV	8 or 9
0A	±1 V	100 uV	10 uV	8 or 9
0B	±500 mV	100 uV	10 uV	8 or 9
0C	±150 mV	100 uV	10 uV	8 or 9
30	-150 mV to +1.5 V	100 uV	10 uV	8 or 9
32	±50 V	100 uV	10 uV	8 or 9
48	DEWETRON defined linearization (area 4)			
49	DEWETRON defined linearization (area 5)			
50	Custom defined linearization (area 6)			
51	Custom defined linearization (area 7)			

Code	Baud rate
03	1200 bps
04	2400 bps
05	4800 bps
06	9600 bps
07	1920 bps
08	38400 bps
09	57600 bps
0A	115200 bps

Data format:



Checksum
0: disable
1: enable

Precision
0: 8 digits
1: 9 digits

Analog input data format
00: engineering units
01: 2 byte signed, binary, scaled to full scale range, MSB first
10: 3 byte signed, binary, scaled to full scale range, MSB first
11: 4 byte signed, binary, scaled to full scale range, MSB first

Set module address

The command has to be sent to the module until the push button on the module gets pressed. Then the module sends the answer.

Command: ##(Addr)SETP\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
\r:	Carriage return (0x0D)

Response: !(Addr)(ModuleType)\r

!:	Response leading code
Addr:	Acknowledged new module address
ModuleType:	11 for PAD-V8-P module

Example: Command: ##01SETP\r

Response: !0111\r
(Address 0x01, PAD-V8-P)

Read module configuration (first channel)

Command: $(Addr)2\r

Response: !(Addr)(InputRange)(BaudRate)(DataFormat)\r

!:	Response leading code
Addr:	Module address
InputRange:	2 bytes hex, values 0x00 to 0xFF hex, MSB first
BaudRate:	Communication baud rate
DataFormat:	Define checksum and output data format
\r:	Carriage return (0x0D)

Read module channel configuration

Command: ??(Addr)\r

Response: !(Addr)(ModuleType)(Averaging)(S)(InputRanges)*8\r

!:	Response leading code
Addr:	Module address
ModuleType:	‘11’ hex for PAD-V8-P
Averaging:	‘01’, ‘04’ or ‘08’ for number of averaged values
S:	Reserve
InputRange:	8*2 bytes hex, values 0x00 to 0xFF hex, MSB first
\r:	Carriage return (0x0D)

Example: Command: ??01\r

Response: !011101S0809053202030432\r
(Address 0x01, PAD-V8-P, no averaging:
channel 0: ±10 V, channel 1: ±5 V, channel 2: ±2.5 V, channel 3: ±50 V,
channel 4: ±0.1 V, channel 5: ±0.5 V, channel 6: ±1 V, channel 7: ±50 V)

Read module type and firmware verison

Command: ??(Addr)VER\r

Response: !PAD-V8-P Ver. x.xx\r

!:	Response leading code Module type and software version x.xx
\r	Carriage return (0x0D)

Read firmware verison number

Command: $(Addr)F\r

Response: !(Addr)(Version)\r

!:	Response leading code
Addr:	Module address
Version:	Firmware version (e.g., V1.14)
\r	Carriage return (0x0D)

Read module name

Command: $(Addr)M\r

Response: !(Addr)(Name)\r

!:	Response leading code
Addr:	Module address
Name:	Module name (e.g., PAD-V8-P)
\r	Carriage return (0x0D)

Read serial number

Command: ??(Addr)SNR\r

Response: !(Addr)(SNR)\r

!:	Response leading code
SNR:	6 ot 8 bytes serial number (ASCII)
\r:	Carriage return (0x0D)

Set input range for channel x

The input range can be set different for each channel.

Command: %(Addr)T(ChannelNo)(InputRange)\r

%:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
ChannelNo:	Channel number from 0 to 7
InputRange:	2 bytes, MSB first, values according table above
\r:	Carriage return (0x0D)

Response:

!(Addr)\r	Command ok
?(Addr)\r	Command failure (e.g. sensor identification used)

Example:

Command: %01T00C\r

Response: !01\r
(Module set to ±150 mV input at channel 0)

Read analog input from channel x

Command: #(Addr)(ChannelNo)\r

#:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
ChannelNo:	Channel number from 0 to 7
\r:	Carriage return (0x0D)

Response: >(Data)\r

	>	Resonse leading code
	Data:	Input value (in mV)
	\r:	Carriage return (0x0D)
Example:	Command:	#010\r
	Response:	+00025.7\r (input value 25.7 mV at channel 0 with 8 digit precision)

Read all 8 channels data values

Command:	$(Addr)A\r	
Response:	Valid:	>(Data)(Data)(Data)(Data)(Data)(Data)(Data)(Data)\r
	Invalid:	?AA\r
	>:	Response leading code for valid command
	?:	Response leading code for invalid command
	Addr:	Module address (2 characters hex from 0x00 to 0xFE)
	Data:	8 or 9 character ASCII value (depending on precision)
Example:	Command:	$01A\r
	Response:	>+01100.1+00257.3-47004.7+00237.0+08029.2+00097.4-00002.3+05119.5\r

channel 0:	+1100.1 mV	channel 4:	+8029.2 mV
channel 1:	+0257.3 mV	channel 5:	+0097.4 mV
channel 2:	-47004.7 mV	channel 6:	-0002.3 mV
channel 3:	+0237.0 mV	channel 7:	+5119.5 mV

Synchronized sampling

This command will store the current analog values for all channels. As this command uses no module address, it has an effect on all PAD modules within the system. The values are stored within each module in a temporary register.

Command: #**\r

Response: no response; to read out the values, use the following command.

Read all internally stored values
This command will read out the internally stored values.

Command: $(Addr)S\r

Response: !(S)(Data)(Data)(Data)(Data)(Data)(Data)(Data)(Data)\r

!:	Command leading code
S:	Readout state (1 = first read out; 0 = already read out)
Data:	Stored analog values for channel 0 to 7

Read all internally stored values

This command will read out the internally stored values.

Command: $(Addr)S\r

Response: !(S)(Data)(Data)(Data)(Data)(Data)(Data)(Data)(Data)\r

!:	Command leading code
S:	Readout state (1 = first read out; 0 = already read out)
Data:	Stored analog values for channel 0 to 7 (in mV)

Read output range of channel x

Command: $(Addr)W(ChannelNo)\r

Response: !(Addr)(MinSign)(MinValue)(MaxSign)(MaxValue)\r

!:	Response leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
MinSign:	Sign for minimum (‘-’ or ‘+’)
MinValue:	Minimum value of input range, 8 bytes ASCII
MinSign:	Sign for minimum (‘-’ or ‘+’)
MaxValue:	Maximum value of input range, 8 bytes ASCII

Example: Command: $01W0\r

Response: !01-173.7500+275.2450\r
(Module address 01, min. range -173.7500 N, max. range +275.2450 N)

Set duration of averaging

Command: %(Addr)M(Duration)\r

%:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
Duration:	‘01’, ‘04’ or ‘08’ allowed 01: no averaging 04: average over 4 values 08: average over 8 values

Response:

!(Addr)\r	Valid command
?(Addr)\r	Invalid command, syntax error

Example: Command: $01M04\r

Response: !01\r
(Module at address 01 set to 4 values averaging)

Set LED state (EPAD modules don´t support a programmable LED)

Command: $(Addr)L(State)\r

$:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
State:	‘E’, ‘A’ or ‘B’ allowed E: LED on A: LED off B: LED flashing with 1 Hz

Response:

!(Addr)\r	Valid command
?(Addr)\r	Invalid command, syntax error

Read LED and digital input state

This command shows the state of the LED and the digital inputs.

Command: $(Addr)I\r

Response:

Valid:	!(Addr)(LEDState)(DIState)\r
Invalid:	?(Addr)\r
!:	Response leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
LEDState:	’E’ = LED on, ’A’ = LED off, ’B’ = LED flashing with 1 Hz
DIState:	Digital input state, 4 bytes ’A’ = high or no signal, ’E’ = low

Set channel multiplex state

Used to reduce data transfered with $(Addr)A and $(Addr)S command. Data of disabled channels will not be transfered.

Command: $(Addr)5(Channels)\r

$:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
Channels:	2 character value from 0x00 to 0xFF hex, 8 bits refer to channels 7 to 0 1 = enable, 0 = disable

Response:

!(Addr)\r	Valid command
?(Addr)\r	Invalid command
No response	Syntax error

Example:

Command: $015AA\r

Response: !01\r
(channel 1, 3, 5, 7 enabled, channel 0, 2, 4, 6 disabled)

Read channel multiplex state

Command: $(Addr)6\r

Response:

Valid:	!(Addr)(Channels)\r
Invalid:	?(Addr)\r
Syntax error:	No response
!:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
Channels:	2 character value from 0x00 to 0xFF hex, 8 bits refer to channels 7 to 0 1 = enable, 0 = disable

Example:

Command: $016\r

Response: !01AA\r
(channel 1, 3, 5, 7 enabled, channel 0, 2, 4, 6 disabled)

Zero calibration for channel x (1)

Connect 0.000 mV signal to input channel x and wait at least 5 sec. before calibration.

Command: %(Addr)Z(ChannelNo)\r

%:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
ChannelNo:	Input channel number from 0 to 7, where signal is connected

Response:

!(Addr)\r	Valid command
?(Addr)\r	Invalid command

Zero calibration for all channels (1)

Connect 0.000 mV signal to ALL input channels and wait at least 5 sec. before calibration.

Command: %(Addr)ZA\r

%:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)

Response:

!(Addr)\r	Valid command
?(Addr)\r	Invalid command

Note: This command is valid for mudules after rev. 1.14 only (older versions used $(Addr)2\r)

(1) *Sending this commands without connecting the appropriate signals causes wrong measurement values!*

Span calibration for channel x (1)

Connect +10 V signal to input channel and wait at least 5 sec. before calibration.

Command: %(Addr)S(ChannelNo)\r

	%:	Command leading code
	Addr:	Module address (2 characters hex from 0x00 to 0xFE)
	ChannelNo:	Input channel number from 0 to 7, where signal is connected
Response:	!(Addr)\r	Valid command
	?(Addr)\r	Invalid command

Span calibration for all channels (1)

Connect +10 V signal to ALL input channels and wait at least 5 sec. before calibration.

Command: %(Addr)SA\r

	%:	Command leading code
	Addr:	Module address (2 characters hex from 0x00 to 0xFE)
Response:	!(Addr)\r	Valid command
	?(Addr)\r	Invalid command

Note: This command is valid for mudules after rev. 1.14 only (older versions used $(Addr)0\r)

Open memory area for custom linearisation

As the PAD-V8-P module allows different sensor linearisations directly within the module, the following three commands are necessary to write the data.

Command: ##(Addr)OK(Number)\r

	##:	Command leading code
	Addr:	Module address (2 characters hex from 0x00 to 0xFE)
	Number:	Customer defined linearisation can be stored into area ‘6’ (range code 50) and ‘7’ (range code 51), see also table on page 5-2
Response:	!(Addr)\r	Valid command
	?(Addr)\r	Invalid command
Example:	Command:	##01OK6\r
	Response:	!01\r (open linearisation area 6 for writing data)

Set custom linearisation point

Command: ##(Addr)T(PointNo)(Sign)(Voltage)(Sign)(Value)\r

	##:	Command leading code
	Addr:	Module address (2 characters hex from 0x00 to 0xFE)
	PointNo:	2 bytes ASCII from 00 to 47; number of the value; 48 values can be stored
	Sign:	‘-’ or ‘+’ sign
	Voltage:	6 bytes ASCII, MSB first, values from 0 to 999999 in 100 µV
	Value:	5 bytes ASCII, MSB first, physical values from 0 to 29999, one or two dezimals (depending on precision mode)
Response:	!(Addr)\r	Valid command
	?(Addr)\r	Invalid command
Example:	Command:	##01T01+012300+01230\r
	Response:	!01\r (Module address 01, point 1, +1.23 V = +123.0 mm)

(1) *Sending this commands without connecting the appropriate signals causes wrong measurement values!*

Close memory area for custom linearisation

Close memory area to avoid unintentional writing to module.

Command:	##(Addr)C(InputRange)\r	
	##:	Command leading code
	Addr:	Module address (2 characters hex from 0x00 to 0xFE)
	InputRange:	2 bytes hex input range code, area 6 = 50 hex, area 7 = 51 hex
Response:	!(Addr)\r	Valid command
	?(Addr)\r	Invalid command

PAD-V8-P

Nr:	Command					Response	
1	##	AA	SETP		CR	!AAMM	Set module address
2	??	AA	VER		CR	!PAD-V8-P VER. X.xx	Read module type and firmware version
3	??	AA			CR	!{AAMMoSFF}*8	Read module channel configuration
4	??	AA	SNR		CR	!AAxxxxxxxx	Read serial number
5	%	AA	T	nFF	CR	!AA	Set input range for channel n
6	%	AA	M	o	CR	!AA	Duration of averaging
7	%	AA	Z	n (1)	CR	!AA	Zero calibration for channel n
8	%	AA	S	n (1)	CR	!AA	Span calibration for channel n @ 10.00000 V
9	$	AA	S		CR	!S{xxxxxxxx}*8	Read all internally stored values
10	$	AA	L	b	CR	!AA	Set LED state
11	#	**			CR		Synchronized sampling
12	##	AA	OK	k	CR	!AA	Open memory area for custom linearisation
13	##	AA	T	NNsxxxxxxsxxxxx	CR	!AA	Set custom linearisation point
14	##	AA	C	KK	CR	!AA	Close memory area for custom linearisation
15	%	AA	{adr.new}	FFBBDD	CR	!AA	Set module configuration
16	#	AA		n	CR	>sxxxxxxxx	Read analog input from channel n
17	#	AA			CR	>{sxxxxxxxx}*8	Read all channels
18	$	AA	0		CR	!AA old command	Span calibration for all channels @ 10.00000 V
19	$	AA	1		CR	!AA old command	Zero calibration for all channels
20	$	AA	2		CR	!AAFFBBDD	Read module configuration (first channel)
21	$	AA	5	hh	CR		Set channel multiplex state
22	$	AA	6		CR	hh	Read channel multiplex state
23	$	AA	A		CR	>{sxxxxxxxx}*8	Read all 8 channel data values
24	$	AA	F		CR	!AAVx.xx	Read module firmware
25	$	AA	M		CR	!AAPAD-V8-P	Read module name

AA	Address
MM	Module type (11)
b	E/A/B
xxx	ASCII
n	Channel number
n (1)	0 to 7 channel A = all channels after rev. 1.14
FF	Input range
o	01, 04, 08
s	"+" "-" sign
k	6/7 custom lin.
KK	lin. type (FE)
w	"M" "I"
hh	HEX wert
BB	Baud rate
DD	Data format
S	Flag 0/1

BB	Baud rate
3	1200 bps
4	2400 bps
5	4800 bps
6	9600 bps
7	1920 bps
8	38400 bps
9	57600 bps
0A	115200 bps

Code (hex)	Input range	Resolution	Precision mode	Digits
02	±100 mV	100 uV	10 uV	8 or 9
03	±500 mV	100 uV	10 uV	8 or 9
04	±1 V	100 uV	10 uV	8 or 9
05	±2.5 V	100 uV	10 uV	8 or 9
08	±10 V	100 uV	10 uV	8 or 9
09	±5 V	100 uV	10 uV	8 or 9
0A	±1 V	100 uV	10 uV	8 or 9
0B	±500 mV	100 uV	10 uV	8 or 9
0C	±150 mV	100 uV	10 uV	8 or 9
30	-150 mV to +1.5 V	100 uV	10 uV	8 or 9
32	±50 V	100 uV	10 uV	8 or 9
48	DEWETRON defined linearization (area 4)			
49	DEWETRON defined linearization (area 5)			
50	Custom defined linearization (area 6)			
51	Custom defined linearization (area 7)			

Module commands

Command summary for PAD-TH8-P module

Command	Syntax
GENERAL COMMANDS	
Set module configuration	%(OldAddr)(NewAddr)(InputRange)(BaudRate)(Format)\r
Set module address	##(Addr)SETP\r
Read module configuration	??(Addr)\r
Read module type and firmware version	??(Addr)VER\r
Read firmware version number	$(Addr)F\r
Read module name	$(Addr)M\r
Read serial number	??(Addr)SNR\r
FUNCTIONAL COMMANDS	
Set input range for channel x	%(Addr)T(ChannelNo)(InputRange)\r
Set PAD-CB8 -P input range for channel x	%(Addr)I(ChannelNo)(InputRange)\r
Read analog input from channel x	#(Addr)(ChannelNo)\r
Read all 8 channel data values	$(Addr)A\r
Synchronized sampling	#**\r
Read all internally stored values	$(Addr)S\r
Read output range of channel x	$(Addr)W(ChannelNo)\r
Set duration of averaging	%(Addr)M(AverageLength)\r
Set LED state	$(Addr)L(State)\r
Read LED and TC-type state	$(Addr)I\r
Read channel multiplex state	$(Addr)6\r
Set channel multiplex state	$(Addr)5(Channels)\r
Read CJC value	$(Addr)3\r
Set CJC offset value	$(Addr)9(Sign)(Value)\r
Zero calibration for channel x	%(Addr)Z(ChannelNo)\r
Zero calibration for all channels	$(Addr)1\r (only modules older rev. 5.03)
Span calibration for channel x	%(Addr)S(ChannelNo)\r
Span calibration for all channels	$(Addr)0\r (only modules older rev. 5.03)
Open memory area for custom linearisation	##(Addr)OK(Number)\r
Set custom linearisation point	##(Addr)T(PointNo)(Sign)(Voltage)(Sign)(Value)\r
Close memory area for custom linearisation	##(Addr)C(InputRangeCode)(CJC)\r
Reset module	Connect Pin 21(Reset) to pin 22(GND) on the DSUB 25 connector Press the ID button during powering on the module.

Set module configuration

Command: %(OldAddr)(NewAddr)(InputRange)(BaudRate)(DataFormat)\r

%	Command leading code
(OldAddr)	The original / old module address. The factory default address of a module is 00. The address range is 0x00 to 0xFE.
(NewAddr)	The new module address (0x00 to 0xFE - be careful to get no conflict with existing module addresses).
(InputRange)	Define the analog input range, refered to the table *input range*.
(BaudRate)	Communication baud rate; recommended ‘06’ hex (= 9600 bps).
(DataFormat)	Define checksum and output data format, refered to the table *data format*

Response:

!(Addr)<CR>	The command to this address is valid.
?(Addr)<CR>	The command is invalid, parameter values are invalid or change settings not possible.

Example:

Command %01300F0600<CR>

%:	command leading code
01:	old module address
30:	new module address
0F:	thermocouple type K input (-270 °C to +1372 °C)
06:	baud rate (9600)
00:	data format (engineering units, checksum disabled)
<CR>:	carriage return

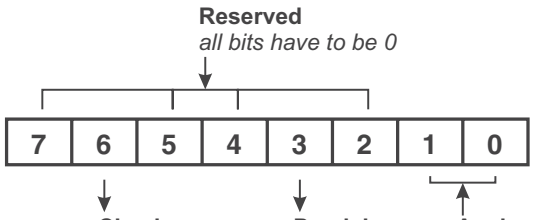
Response !30<CR>

Description: Address successfully changed from 0x01 to 0x30.

Input ranges:

Input range code (hex)	Connected sensor				Input connector state SI1	SI2	SI3	SI4
00	Voltages, ±15 mV range				A	A	A	A
01	Voltages, ±50 mV range				A	A	A	A
02	Voltages, ±100 mV range				A	A	A	A
0C	Voltages, ±150 mV range				A	A	A	A
30	Voltages, -150 mV to +1.5 V range				A	A	A	A
0E	Thermocouple type 'J'	-210	°C to	1200 °C	E	A	A	A
0F	Thermocouple type 'K'	-270	°C to	1372 °C	A	E	A	A
10	Thermocouple type 'T'	-270	°C to	400 °C	E	E	A	A
11	Thermocouple type 'E'	-200	°C to	1000 °C	Special Inputranges : The linearization table and the calibration values are directly stored in the inteligent Connector Block. (Revision 5.00 and higher)			
12	Thermocouple type 'R'	-50	°C to	1760 °C				
13	Thermocouple type 'S'	-50	°C to	1760 °C				
14	Thermocouple type 'B'	500	°C to	1820 °C				
15	Thermocouple type 'N'	-200	°C to	1300 °C				
16	Thermocouple type 'C'	0	°C to	2316 °C				
17	Thermocouple type 'L'	-200	°C to	900°C				
48	DEWETRON defined linearization (area 4)				A	A	E	A
49	DEWETRON defined linearization (area 5)				E	A	E	A
50	Custom defined linearization (area 6)				A	E	E	A
51	Custom defined linearization (area 7)				E	E	E	A
70	Ohm	0	Ohm to	999	CB8-RTD		CB8-RTD S3	
71	PT100 a=385	-200	°C to	800	CB8-RTD		CB8-RTD S3	
72	PT200 a=385	-200	°C to	630	CB8-RTD		CB8-RTD S3	
73	PT500 a=385	-200	°C to	250	CB8-RTD		CB8-RTD S3	
74	PT100 a=3916	-200	°C to	630	CB8-RTD		CB8-RTD S3	
75	NI120	-80	°C to	260	CB8-RTD			
76	Cu427	-80	°C to	260				
77	PT 1000 a=385						CB8-RTD S3	
78	PT1000							
79	PT2000 a=385						CB8-RTD S3	

Data format:



Checksum
0: disable
1: enable

Precision
0: 8 digits
1: 9 digits

Analog input data format
00: engineering units
01: 2 byte signed, binary, scaled to full scale range, MSB first
10: 3 byte signed, binary, scaled to full scale range, MSB first
11: 4 byte signed, binary, scaled to full scale range, MSB first

Code	Baud rate
03	1200 bps
04	2400 bps
05	4800 bps
06	9600 bps
07	1920 bps
08	38400 bps
09	57600 bps
0A	115200 bps

Set module address

The command has to be sent to the module until the push button on the module gets pressed. Then the module sends the answer.

Command: ##(Addr)SETP\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
\r:	Carriage return (0x0D)

Response: !(Addr)(ModuleType)\r

!:	Response leading code
Addr:	Acknowledged new module address
ModuleType:	10 for PAD-TH8-P module

Example:
Command: ##01SETP\r
Response: !0110\r
(Address 0x01, PAD-TH8-P)

Read module configuration (first channel)

Command: $(Addr)2\r

Response: !(Addr)(InputRange)(BaudRate)(DataFormat)\r

!:	Response leading code
Addr:	Module address
InputRange:	2 bytes hex, values 00 to FF hex, MSB first
BaudRate:	Communication baud rate
DataFormat:	Define checksum and output data format
\r:	Carriage return (0x0D)

Read module channel configuration

Command: ??(Addr)\r

Response: !(Addr)(ModuleType)(Averaging)(RangeSel)(InputRanges)*8\r

!:	Response leading code
Addr:	Module address
ModuleType:	‘10’ hex for PAD-TH8-P
Averaging:	‘01’, ‘04’ or ‘08’ for number of averaged values
RangeSel:	‘E’ if external CB8-x-P connector block or EPAD modules are connected
	‘S’ if input range is defined by software
	‘I’ if intelligent connector block is connected
InputRange:	8*2 bytes hex, values 0x00 to 0xFF hex, MSB first (refer ot *input range* table)
\r:	Carriage return (0x0D)

Example:
Command: ??01\r
Response: !011001E0F0F0F0F0F0F0F0F\r
(Address 0x01, PAD-TH8-P, no averaging, ext. sensor, 8 thermocouples ‘K’)

Read module type and firmware verison

Command: ??(Addr)VER\r

Response: !PAD-TH8-P Ver. x.xx\r

!:	Response leading code
Module type and software version	
\r	Carriage return (0x0D)

Read firmware version number

Command: $(Addr)F\r

Response: !(Addr)(Version)\r

!:	Response leading code
Addr:	Module address
Version:	Firmware version
\r	Carriage return (0x0D)

Read module name

Command: $(Addr)M\r

Response: !(Addr)(Name)\r

!:	Response leading code
Addr:	Module address
Name:	Module name
\r	Carriage return (0x0D)

Read serial number

Command: ??(Addr)SNR\r

Response: !(Addr)(SNR)\r

!:	Response leading code
SNR:	6 to 8 bytes serial number (ASCII)
\r:	Carriage return (0x0D)

Set input range for channel x

The input range can be set different for each channel. This function is only available if CB8-x-P is not connected! Different input ranges within one PAD module are not supported in DeweSoft 6.

Command: %(Addr)T(ChannelNo)(InputRange)\r

%:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
ChannelNo:	Channel number from 0 to 7 "A" instead of a number will take effect on all channels
T:	Command
InputRange:	2 bytes, MBS first, values according table above
\r:	Carriage return (0x0D)

Response:	!(Addr)\r	Command ok
	?(Addr)\r	Command failure (e.g. CB8-x-P connected)
Example:	Command:	%01T20C\r
Response:	!01\r	
		(Module set to ±150 mV input (Range 0C) at channel 2)

Set PAD-CB8-P input range for channel x

Changes the input range of the PAD-CB8-xyz module.

This command will work for all PAD-CB8 break out boxes with revision 5.0 or higher. All PAD-CB8-RTD belongs to that group. The input range can be selected seperate for each channel. This is not supported by DeweSoft 6. Every PAD-Cb8 where the channels are individually configured will work in DeweSoft but recognized as "AUTO" range. For thermocouple breakout boxes it is not recommended to change the input range.

Command: %(Addr)I(ChannelNo)(InputRange)\r

%:	Command leading Code
Addr:	Module address (2 characters from 0x00 to 0xFE)
I:	Command
ChannelNo:	Channel number from 0 to 7 "A" instead of a number will take effect on all channels
InputRange:	2 bytes, MSB first, values according to table above
\r:	Carraige return (0x0D)

Response:

!(Addr)\r	Valid Command
?(Addr)\r	Invalid command (e.g no CB8-x-P connected)

Example: Command: %01IA71\r

Response: !01\r
(set all PAD-CB8-RDT channels to Pt100 input (range71))

Read analog input from channel x

Command: #(Addr)(ChannelNo)\r

#:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
ChannelNo:	Channel number from 0 to 7
\r:	Carriage return (0x0D)

Response: >(Data)\r

>	Resonse leading code
Data:	Input value
\r:	Carriage return (0x0D)

Example: Command: #010\r

Response: >+00025.7\r (input value 25.7 at channel 0 with 8 digit precision)

Read all 8 channels data values

Command: $(Addr)A\r

Response:

Valid:	>(Data)(Data)(Data)(Data)(Data)(Data)(Data)(Data)\r
Invalid:	?AA\r

>:	Response leading code for valid command
?:	Response leading code for invalid command
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
Data:	8 or 9 character ASCII value (depending on precision)

Example: Command: $01A\r

Response: >+01100.1+00257.3-00004.7+00023.7+00029.2+00097.4-00002.3+00119.5\r

channel 0:	+1100.1 °C	channel 4:	+29.2 °C
channel 1:	+257.3 °C	channel 5:	+97.4 °C
channel 2:	-4.7 °C	channel 6:	-2.3 °C
channel 3:	+23.7 °C	channel 7:	+119.5 °C

Synchronized sampling

This command will store the current analog values for all channels. As this command uses no module address, it has an effect on all PAD modules within the system. The values are stored within each module in a temporary register.

Command: #**\r

Response: no response; to read out the values, use the following command.

Read all internally stored values

This command will read out the internally stored values.

Command: $(Addr)S\r

Response: !(S)(Data)(Data)(Data)(Data)(Data)(Data)(Data)(Data)\r

!:	Command leading code
S:	Readout state (1 = first read out; 0 = re-read of old values)
Data:	Stored analog values for channel 0 to 7

Read output range of channel x

Command: $(Addr)W(ChannelNo)\r

Response: !(Addr)(MinSign)(MinValue)(MaxSign)(MaxValue)\r

!:	Response leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
MinSign:	Sign for minimum (‘-’ or ‘+’)
MinValue:	Minimum value of input range, 8 bytes ASCII
MaxSign:	Sign for maximum (‘-’ or ‘+’)
MaxValue:	Maximum value of input range, 8 bytes ASCII

Example: Command: $01W0\r

Response: !01-000270.0+001372.0\r
(Module address 01, min. range -270.0 °C, max. range +1372.0 °C)

Read serial number of connectorblock

Command: ??(Addr)ISNR\r

Response: !(Addr)(Serial)\r

!:	Response leading code
(Addr):	Module address (2 characters hex from 0x00 to 0xFE)
(Serial):	Serial of connectorblock
\r:	Carriage return (0x0D)

Set duration of averaging

Command: %(Addr)M(AverageLength)\r

%:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
Average:	‘01’, ‘04’ or ‘08’ allowed 01: no averaging 04: average over 4 values 08: average over 8 values

Response:

!(Addr)\r	Valid command
?(Addr)\r	Invalid command, syntax error

Example: Command: $01M04\r

Response: !01\r
(Module at address 0x01 set to 4 values averaging)

Set LED state (Not supported at EPAD-TH8-P)

Command: $(Addr)L(State)\r

$:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
State:	'E', 'A' or 'B' allowed E: LED on A: LED off B: LED flashing with 1 Hz

Response:

!(Addr)\r	Valid command
?(Addr)\r	Invalid command, syntax error

Read LED and thermocouple type state

This command shows the state of the LED and the connected sensor according the following table.
If the input connector pin 17, 18, 19 (SI1, SI2, SI3) are not connected to GND, the module is set to software selected range with or without linearisation. Typestate of sensors with Rev. < 5.0 are not recognized by this command.

Command: $(Addr)I\r

Response:

Valid:	!(Addr)(LEDState)(TCState)\r
Invalid:	?(Addr)\r

!:	Response leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
LEDState:	'E' = LED on, 'A' = LED off, 'B' = LED flashing with 1 Hz
TCState:	Input line state for sensor type recognition, 4 bytes 'A' = no sensor - pin open, 'E' = sensor connected - pin grounded

Input range code (hex)	Connected sensor	Input connector state			
		SI1	SI2	SI3	SI4
00	Voltages, ±15 mV range	A	A	A	A
01	Voltages, ±50 mV range	A	A	A	A
02	Voltages, ±100 mV range	A	A	A	A
0C	Voltages, ±150 mV range	A	A	A	A
30	Voltages, -150 mV to +1.5 V range	A	A	A	A
31	Voltages, -1.5 V to +1.5 V range	A	A	A	A
0E	Thermocouple type 'J' -210 °C .. 1200 °C	E	A	A	A
0F	Thermocouple type 'K' -270 °C .. 1372 °C	A	E	A	A
10	Thermocouple type 'T' -270 °C .. 400 °C	E	E	A	A
48	DEWETRON defined linearization (area 4)	A	A	E	A
49	DEWETRON defined linearization (area 5)	E	A	E	A
50	Custom defined linearization (area 6)	A	E	E	A
51	Custom defined linearization (area 7)	E	E	E	A

Set channel multiplex state

Used to reduce data transfered with $(Addr)A(read all channels data values) and
$(Addr)S (read all internally stored values)
command. Data of disabled channels will not be transfered.

Command:	$(Addr)5(Channels)\r	
	$:	Command leading code
	Addr:	Module address (2 characters hex from 0x00 to 0xFE)
	Channels:	2 character value from 0x00 to 0xFF hex, 8 bits refer to channels 7 to 0 1 = enable, 0 = disable
Response:	!(Addr)\r	Valid command
	?(Addr)\r	Invalid command
	No response	Syntax error
Example:	Command:	$015AA\r
	Response:	!01\r (channel 1, 3, 5, 7 enabled, channel 0, 2, 4, 6 disabled)

Read channel multiplex state

Command:	$(Addr)6\r	
Response:	Valid:	!(Addr)(Channels)\r
	Invalid:	?(Addr)\r
	Syntax error:	No response
	!:	Command leading code
	Addr:	Module address (2 characters hex from 0x00 to 0xFE)
	Channels:	2 character value from 00 to FF hex, 8 bits refer to channels 7 to 0 1 = enable, 0 = disable
Example:	Command:	$016\r
	Response:	!01AA\r (channel 1, 3, 5, 7 enabled, channel 0, 2, 4, 6 disabled)

Read CJC value

Command:	$(Addr)3\r	
Response:	Valid:	!(Sign)(Value)\r
	Invalid:	?(Addr)\r
	!:	Response leading code
	Sign:	'-' or '+' sign
	Value:	CJC temperature value (ASCII)
Example:	Command:	$013\r
	Response:	!+0023.8\r (CJC temperature +23.8 °C)

Read type and firmware version of connectorblock

Command:	$(Addr)IVER\r	
Response:	!PAD-CB-x-P Ver.5.025	
	!:	Response leading code Connectorblock type and software version
	\r:	Carriage return (0x0D)

Set CJC offset value

Used to calibrate the CJC temperature according to a reference sensor.

Command: $(Addr)9(Sign)(Value)\r

$:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
Sign:	'-' or '+' sign
Value:	4 character hex value, 1 count = 0.01 °C

Response:

!(Addr)\r	Valid command
?(Addr)\r	Invalid command

Example: Command: $019+000A\r

Response: !01\r
(CJC offset set to 000A hex = 10 dec = 10 * 0.01 °C = 0.1 °C)

Zero calibration for channel x (1)

Connect 0.000 mV signal to input channel and wait at least 5 sec. before calibration.

Command: %(Addr)Z(ChannelNo)\r

%:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
ChannelNo:	Input channel number from 0 to 7, where signal is connected

Response:

!(Addr)\r	Valid command
?(Addr)\r	Invalid command

Zero calibration for all channels (1)

Connect 0.000 mV signal to ALL input channels and wait at least 5 sec. before calibration.

Command: %(Addr)ZA\r

%:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)

Response:

!(Addr)\r	Valid command
?(Addr)\r	Invalid command

Note: This command is valid for mudules after rev. 5.04 only (older versions used $(Addr)1\r)

Span calibration for channel x (1)

Connect +1 V signal to input channel and wait at least 5 sec. before calibration.

Command: %(Addr)S(ChannelNo)\r

%:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
ChannelNo:	Input channel number from 0 to 7, where signal is connected

Response:

!(Addr)\r	Valid command
?(Addr)\r	Invalid command

(1) ***Sending this commands without connecting the appropriate signals causes wrong measurement values!***

(1) **Sending this commands without connecting the appropriate signals causes wrong measurement values!**

Span calibration for all channels (1)

Connect +1 V signal to ALL input channels and wait at least 5 sec. before calibration.

Command: %(Addr)\r

%:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)

Response:

!(Addr)\r	Valid command
?(Addr)\r	Invalid command

Note: This command is valid for mudules after rev. 5.04 only (older versions used $(Addr)0\r)

(open linearisation area 6 for writing data)

Open memory area for custom linearisation

As the PAD-TH8-P module allows different sensor linearisations directly within the module, the following three commands are necessary to write the data.

Command: ##(Addr)OK(Number)\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
Number:	Customer defined linearisation can be stored into area ‘6’ and ‘7’

Response:

!(Addr)\r	Valid command
?(Addr)\r	Invalid command

Example:

Command:	##01OK6\r
Response:	!01\r

(1) *Sending this commands without connecting the appropriate signals causes wrong measurement values!*

Set custom linearisation point

Command: ##(Addr)T(PointNo)(Sign)(Voltage)(Sign)(Value)\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
PointNo:	2 bytes ASCII from 00 to 47; number of the value; 48 values can be stored
Sign:	‘-’ or ‘+’ sign
Voltage:	6 bytes ASCII, MSB first, values from 0 to 999999 in µV
Value:	5 bytes ASCII, MSB first, values from 0 to 29999 in 0.1 °C

Response:

!(Addr)\r	Valid command
?(Addr)\r	Invalid command

Example:

Command:	##01T01+012300+01230\r
Response:	!01\r

(Module address 0x01, point 1, +12.3 mV = +123.0 °C)

Close memory area for custom linearisation

Close memory area to avoid unintentional writing to module.

Command: ##(Addr)C(InputRange)(CJC)\r

##:	Command leading code
Addr:	Module address (2 characters hex from 0x00 to 0xFE)
InputRange:	2 bytes hex input range code, values above 50 hex recommended (for details see ‘set module configuration’ command)
(CJC):	CJC temperature use / ignore (Y / N)

Response:

!(Addr)\r	Valid command
?(Addr)\r	Invalid command

PAD Module reset

It could happen that because of numerous combinations of baud rates and addresses the Pad module gets “lost”. This means it is on a baud rate and/or address you don’t know. Then the only possibility to communicate with the module is to perform a hardware reset. This will restore all default values of the module except the calibration information.

Reset Procedure:

Connect Pin 21(Reset) to pin 22(GND) on the DSUB 25 connector.
Press the ID button during powering on the module.

Default Values:

Baud rate:	9600Bps
Address:	00h (equals a cleared module in DeweSoft)
Data Format:	engineering unit, no checksum.

Notes

PAD-TH8-P

Nr:	Command					Response	
1	##	AA	SETP		CR	!AAMM	Set module address
2	??	AA	VER		CR	!PAD-TH8-P VER. X.xx	Read module type and firmware version
3	??	AA			CR	!{AAMMoSFF}*8	Read module channel configuration
4	??	AA	IVER		CR	!PAD-CB-x-P VER. 5.025	Read type and firmware version of the connectorblock
5	??	AA	SNR		CR	!AAxxxxxxxx	Read serial number
6	??	AA	ISNR		CR	!AAxxxxxxxx	Read serial number of the connectorblock
7	%	AA	T	n [1] FF	CR	!AA	Set input range for channel n
8	%	AA	I	n [1] FF	CR		Set input range for channel n of the connectorblock
9	$	AA	W	n	CR	!AAsiiiiiiiismmmmmmmm	Read output range of channel n
10	%	AA	M	o	CR	!AA	Duration of averaging
11	%	AA	Z	n [1]	CR	!AA	Zero calibration for channel n
12	%	AA	S	n [1]	CR	!AA	Span calibration for channel n @ 1.00000 V
13	$	AA	S		CR	!S{xxxxxxxx}*8	Read all internally stored values
14	$	AA	L	b	CR	!AA	Set LED state
15	#	**			CR	!AA	Synchronized sampling
16	##	AA	OK	k	CR	!AA	Open memory area for custom linearisation
18	##	AA	T	NNsxxxxxxsxxxxx	CR	!AA	Set custom linearisation point
19	##	AA	C	KK{CJC}	CR	!AA	Close memory area for custom linearisation
20	%	AA	{adr.new}	FFBBDD	CR	!AA	Set module configuration
21	#	AA		n	CR	>sxxxxxxxx	Read analog input from channel n
22	#	AA			CR	>{sxxxxxxxx}*8	Read all channels
23	$	AA	0		CR	!AA old command	Span calibration for all channels @ 1.00000 V
24	$	AA	1		CR	!AA old command	Zero calibration for all channels
25	$	AA	2		CR	!AAFFBBDD	Read module configuration (first channel)
26	$	AA	3		CR	!sxxxx.x	Read CJC value
27	$	AA	5	hh	CR	!AA	Set channel multiplex state
28	$	AA	6		CR	!AAhh	Read channel multiplex state
29	$	AA	9	sHHHH	CR	!AA	Set CJC offset value
30	$	AA	A		CR	>{sxxxxxxxx}*8	Read all 8 channel data values
31	$	AA	F		CR	!AAVx.xx	Read module firmware
32	$	AA	M		CR	!AAPAD-TH8-P	Read module name

AA	Address
MM	Module type (10)
NN	Lin. point number
b	E, A, B
xxx	ASCII
n	Channel number
n [1]	0 to 7 channel A = all channels after rev. 5.04
FF	Input range
ii..i	Range min.
mm..m	Range max.
o	01, 04, 08
s	"+" "-" sign
k	0 to 7 custom lin.
KK	lin. type (FE)
{CJC}	"Y" "N"
w	"M" "I"
hh	HEX value
HHHH	Hex CJC Temp
BB	Baud rate
DD	Data format
S	Flag 0/1

BB	Baud rate
03	1200 bps
04	2400 bps
05	4800 bps
06	9600 bps
07	1920 bps
08	38400 bps
09	57600 bps
0A	115200 bps

Input range code (hex)	Connected sensor	Input connector state			
		SI1	SI2	SI3	SI4
00	Voltages, ±15 mV range	A	A	A	A
01	Voltages, ±50 mV range	A	A	A	A
02	Voltages, ±100 mV range	A	A	A	A
0C	Voltages, ±150 mV range	A	A	A	A
30	Voltages, -150 mV to +1.5 V range	A	A	A	A
31	Voltages, -1.5 V to +1.5 V range	A	A	A	A
0E	Thermocouple type 'J' -210 °C .. 1200 °C	E	A	A	A
0F	Thermocouple type 'K' -270 °C .. 1372 °C	A	E	A	A
10	Thermocouple type 'T' -270 °C .. 400 °C	E	E	A	A
48	DEWETRON defined linearization (area 4)	A	A	E	A
49	DEWETRON defined linearization (area 5)	E	A	E	A
50	Custom defined linearization (area 6)	A	E	E	A
51	Custom defined linearization (area 7)	E	E	E	A

Module commands

Command summary for PAD-RTD3 module

Command	Syntax
GENERAL COMMANDS	
Set configuration	**%(OldAddr)(NewAddr)(InputRange)(BaudRate)(DataFormat)**
Read configuration	$(Addr)2
Read module name	$(Addr)M
Read firmware version	$(Addr)F
FUNCTIONAL COMMANDS	
Read analog data from channel 0	**#(Addr)**
Read analog data from channel N	**#(Addr)(ChannelNo)**
Read analog data from all channels	**#(Addr)A**
Span calibration	$(Addr)0(ChannelNo)
Offset calibration	$(Addr)1(ChannelNo)
Enable / disable channel for multiplexing	$(Addr)5(ChannelVal)
Read channel multiplexing status	$(Addr)6

Set configuration:

Command: %(OldAddr)(NewAddr)(InputRange)(BaudRate)(DataFormat)\r

%: Command leading code

(OldAddr): The original / old module address. The factory default address of a module is 00. The address range is 0x00 to 0xFF hex.

(NewAddr): The new module address (be careful to get no conflict with existing module addresses).

(InputRange): Define the analog input range, refered to the table *input range* on next page.

(BaudRate): Define communication baud rate, recommended '06' hex (= 9600 bps).

(DataForm.): Define checksum and output data format, refered to the table *data format* on next page.

Response: !(Addr)\r: The command to this address is valid.

?(Addr)\r: The command is invalid, parameter values are invalid or attempt to change settings without short circuit on the default pins 21 and 22.

Example: Command: %0130210600\r

%: command leading code
01: old module address
30: new module address
21: input range Pt100 0 °C to 100 °C a = 0.00385
06: baud rate (9600)
00: data format (engineering units, checksum disabled)
<\r: carriage return

Response: !30\r

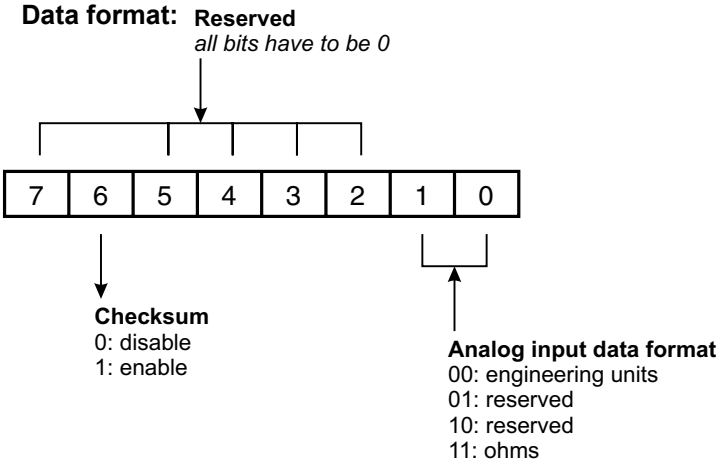
Description: Address successfully changed from 0x01 to 0x30.

Module commands - configuration tables

Input ranges:

Code (hex)	Input range		
20	Pt100	-100 °C to 100 °C	a = 0.00385
21	Pt100	0 °C to 100 °C	a = 0.00385
22	Pt100	0 °C to 200 °C	a = 0.00385
23	Pt100	0 °C to 600 °C	a = 0.00385
24	Pt100	-100 °C to 100 °C	a = 0.003916
25	Pt100	0 °C to 100 °C	a = 0.003916
26	Pt100	0 °C to 200 °C	a = 0.003916
27	Pt100	0 °C to 600 °C	a = 0.003916
28	Ni100	0 °C to 100 °C	
29	Ni120	0 °C to 100 °C	

Data format:



Read module configuration

Command: $(Addr)2\r

Response: !(Addr)(InputRange)(BaudRate)(DataFormat)\r

!: Response leading code
Addr: Module address
(InputRange): Define the input range, refered to the table *Input range* above.
(BaudRate): Define communication baud rate, recommended '06' hex (= 9600 bps).
(DataForm): Define checksum and output data format
\r: Carriage return (0x0D)

Example: Command: $012\r

Response: !01210600\r

01: module address
21: input range Pt100 0 °C to 100 °C a = 0.00385
06: baud rate (9600)
00: data format (engineering units, checksum disabled)
<\r: carriage return

Read module name

Command: $(Addr)M\r

Response: !(Addr)(Name)\r

!:	Response leading code
Addr:	Module address
(Name):	Module name
\r:	Carriage return (0x0D)

Example: Command: $01M\r

Response: !01PAD-RTD3\r

01:	module address
PAD-RTD3:	module name is PAD-RTD3
\r:	carriage return

Read firmware version

Command: $(Addr)F\r

Response: !(Addr)(Firmware)\r

!:	Response leading code
Addr:	Module address
(Firmware):	Module firmware version (5 characters)
\r:	Carriage return (0x0D)

Example: Command: $01F\r

Response: !01E1.2\r

01:	module address 01
E1.2:	current firmware version is E1.2
\r:	carriage return

Read analog data from channel 0

Command: #(Addr)\r

Response: >(InputData)\r

>:	Response leading code
(InputData):	Analog input data from channel 0. The Data format is a + or - sign with five decimal digits and fixed decimal point. The unit depends on the used data format.
\r:	Carriage return (0x0D)

Example: Command: #01\r

Response: >+1.6888\r

>:	response leading code
+1.6888:	analog input module response (channel 0) is +1.6888 units
\r:	carriage return

Read analog data from channel n

Command: #(Addr)(ChannelNo)\r

Response: >(InputData)\r

>:	Response leading code
(InputData):	Analog input data from channel n. The Data format is a + or - sign with five decimal digits and fixed decimal point. The unit depends on the used data format.
\r:	Carriage return (0x0D)

Example: Command: #031\r

Response: >+1.6888\r

>:	response leading code
+1.6888:	module (addr. 03) response channel 1 is +1.6888 units
\r:	carriage return

Read analog data from all channels

Command: #(Addr)A\r

Response: >(InputData)(InputData)(InputData)\r

>: Response leading code
(InputData): Analog input data of Channel 0, 1 and 2. The Data format is a + or - sign with five decimal digits and fixed decimal point. The unit depends on the used data format.
\r: Carriage return (0x0D)

Example: Command: #04A\r

Response: >+102.67+030.45+007.89\r
>: response leading code
+102.67: input value channel 0 of analog input module (addr. 04)
+030.45: input value channel 1 of analog input module (addr. 04)
+007.89: input value channel 2 of analog input module (addr. 04)
\r: carriage return

Span calibration

See also module calibration on next page

Command: $(Addr)0(ChannelNo)\r

$: Command leading code
(Addr): Module address
0: Span calibration command
(ChannelNo): Channel for calibration (1 character) 0 to 2
\r: Carriage return (0x0D)

Response: !(Addr)\r

Example: Command: $0501\r: Perform the span calibration for module 05, channel 1

Response: !05\r: Valid command

Offset calibration

See also module calibration on next page

Command: $(Addr)1(ChannelNo)\r

$: Command leading code
(Addr): Module address
1: Offset calibration command
(ChannelNo): Channel for calibration (1 character) 0 to 2
\r: Carriage return (0x0D)

Response: !(Addr)\r

Example: Command: $0511\r: Perform the offset calibration for module 05, channel 1

Response: !05\r: Valid command

Enable / Disable channels for multiplexing

Command: $(Addr)50(ChValue)\r

$: Command leading code
(Addr): Module address
5: Enable / disable channel
0: Fixed
(ChValue): Bit 2 to 0 of character control channel 2 to 0
bit value 0: Disable channel
bit value 1: Enable channel
\r: Carriage return (0x0D)

Response: !(Addr)\r

Example: Command: $01506\r

$:	command leading code
01:	module address
5:	enable / disable command
0:	fixed
6:	binary 110 - Channel 1 and 2 enabled, channel 0 disabled
\r:	Carriage return

Response: !01\r

Read channel multiplexing status

Command: $(Addr)6\r

Response: !(Addr)(ChValue)\r

Example: Command: $016\r

Response: !014\r

!:	response leading code
01:	module address
5:	enable / disable command
0:	fixed
4:	binary 100 - Channel 1 and 2 enabled, channel 0 disabled
\r:	Carriage return

Module calibration

All PAD-RTD3 modules are within specifications when leaving the factory. The time interval for recalibration depends on environmental conditions. Typically, the calibration should be checked once a year.

For recalibration, perform following steps:

1. Select the correct input range on the module.
2. Enable module channel 0 (disable channel 1 and 2 with command '$(Addr)501').
3. Apply the correct *offset calibration resistance* (see table) for the selected range to module channel 0.
4. Send *'Offset calibration $(Addr)1(ChannelNo)'* to the module.
5. Apply the correct *span resistance* (see table) for the selected range to the module.
6. Send *'Span calibration $(Addr)0(ChannelNo)'* to the module.
7. Repeat steps 2 to 5 three to five times for channels 1 and 2.
8. Set all channels active with command '$(Addr)507'.

Sending this commands without connecting the appropriate signals causes wrong measurement values! Calibration commands are only possible at default settings (pin-21/22 short circuit, address 00, baud rate 9600)

Code (hex)	Input range			Offset calibration resistance	Span calibration resistance
20	Pt100	-100 °C .. 100 °C	a = 0.00385	50 Ohm	200 Ohm
21	Pt100	0 °C .. 100 °C	a = 0.00385	50 Ohm	200 Ohm
22	Pt100	0 °C .. 200 °C	a = 0.00385	50 Ohm	200 Ohm
23	Pt100	0 °C .. 600 °C	a = 0.00385	50 Ohm	350 Ohm
24	Pt100	-100 °C .. 100 °C	a = 0.003916	50 Ohm	200 Ohm
25	Pt100	0 °C .. 100 °C	a = 0.003916	50 Ohm	200 Ohm
26	Pt100	0 °C .. 200 °C	a = 0.003916	50 Ohm	200 Ohm
27	Pt100	0 °C .. 600 °C	a = 0.003916	50 Ohm	350 Ohm
28	Ni100	0 °C .. 100 °C		50 Ohm	350 Ohm
29	Ni120	0 °C .. 100 °C		50 Ohm	200 Ohm

PAD-RTD3

Nr:	Command					Response		
1	%	AA	AA(new)	FFBBDD	CR	!AA	CR	Set module configuration
2	$	AA	2		CR	!AAFFBBDD	CR	Read configuration
3	$	AA	M		CR	!AAM	CR	Read module name
4	$	AA	F		CR	!AAV	CR	Read firmware version
5	#	AA			CR	>L	CR	Read analog data from channel 0
6	#	AA	N		CR	>L	CR	Read analog data from channel N
7	#	AA	A		CR	>LLL	CR	Read analog data from all channels
8	$	AA	0	N	CR	!AA	CR	Span calibration
9	$	AA	1	N	CR	!AA	CR	Offset calibration
10	$	AA	5	0v	CR	!AA	CR	Enable / disable channel for multiplexing
11	$	AA	6		CR	!AA0v	CR	Read channel multiplexing status

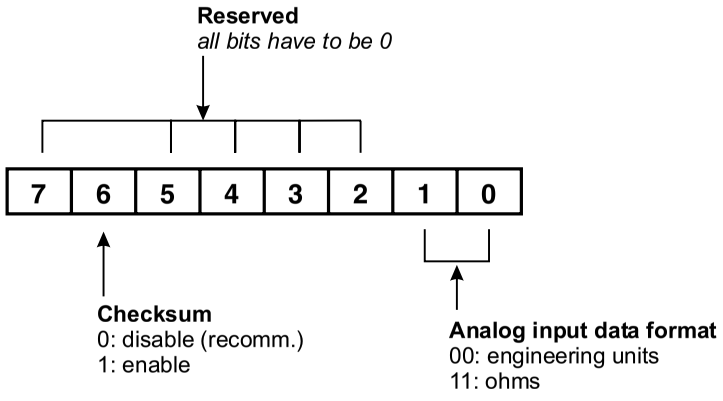
AA	Address
M	Module name
N	Channel number
V	Firmware version
L	Analog input data

BB	Baud rate
03	1200 bps
04	2400 bps
05	4800 bps
06	9600 bps
07	1920 bps
08	38400 bps
09	115200 bps
0A	57600 bps

FF	Input range		
20	Pt100	-100 °C to 100 °C	a = 0.00385
21	Pt100	0 °C to 100 °C	a = 0.00385
22	Pt100	0 °C to 200 °C	a = 0.00385
23	Pt100	0 °C to 600 °C	a = 0.00385
24	Pt100	-100 °C to 100 °C	a = 0.003916
25	Pt100	0 °C to 100 °C	a = 0.003916
26	Pt100	0 °C to 200 °C	a = 0.003916
27	Pt100	0 °C to 600 °C	a = 0.003916
28	Ni100	0 °C to 100 °C	
29	Ni120	0 °C to 100 °C	

v	Channel value
1	Channel 0
2	Channel 1
4	Channel 2
v = sum of activated	

DD	Configuration code table



Module commands

Command summary for PAD-AO1 module

Command	Syntax
GENERAL COMMANDS	
Set configuration	**%(OldAddr)(NewAddr)(OutputRange)(BaudRate)(DataFormat)**
Read configuration	$(Addr)2
Read module name	$(Addr)M
Read firmware version	$(Addr)F
FUNCTIONAL COMMANDS	
Send data to analog output	**#(Addr)(data)**
Save current output as startup value	$(Addr)4
Read back reset status	$(Addr)5
Read back last value	$(Addr)6

Set configuration:

Command: %(OldAddr)(NewAddr)(OutputRange)(BaudRate)(DataFormat)\r

%: Command leading code
(OldAddr): The original / old module address. The factory default address of a module is 00. The address range is 0x00 to 0xFE.
(NewAddr): The new module address (be careful to get no conflict with existing module addresses).
(Outp.Range): Define the analog output range, refered to the table *output range* on next page.
(BaudRate): Define communication baud rate, recommended '06' hex (= 9600 bps).
(DataFormat): Define checksum and output data format, refered to the table *data format* on next page.

Response: !(Addr)\r The command to this address is valid.
?(Addr)\r The command is invalid, parameter values are invalid or attempt to change settings without short circuit on the default pins 21 and 22.

Example: Command: %0102300600\r

%: command leading code
01: old module address
02: new module address
30: output range (0 to 20 mA)
06: baud rate (9600 bps)
00: data format (engineering units, checksum disabled, immediate change)
\r: carriage return

Response: !02\r

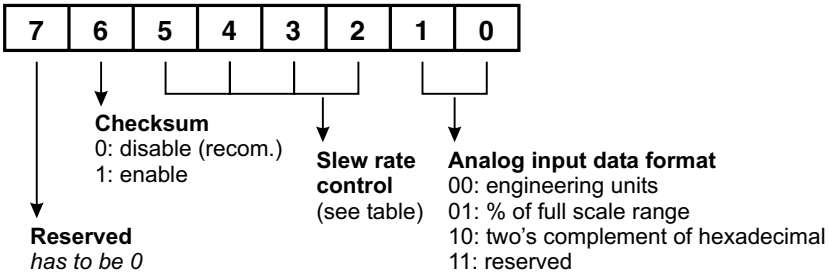
Address successfully changed from 0x01 to 0x02.

Module commands - configuration tables

Output ranges:

Code (hex)	Input range
30	0 .. 20 mA (max. load 500 Ohm!)
31	4 .. 20 mA (max. load 500 Ohm!)
32	0 .. 10 V

Data format:



Code				Slew rate control		Code				Slew rate control	
5	4	3	2	Voltage	Current	5	4	3	2	Voltage	Current
0	0	0	0	immediate change	immediate change	1	0	0	0	8 V/sec	16 mA/sec
0	0	0	1	0.0625 V/sec	0.125 mA/sec	1	0	0	1	16 V/sec	32 mA/sec
0	0	1	0	0.125 V/sec	0.250 mA/sec	1	0	1	0	32 V/sec	64 mA/sec
0	0	1	1	0.250 V/sec	0.500 mA/sec	1	0	1	1	64 V/sec	128 mA/sec
0	1	0	0	0.500 V/sec	1 mA/sec	1	1	0	0	128 V/sec	256 mA/sec
0	1	0	1	1 V/sec	2 mA/sec	1	1	0	1	256 V/sec	512 mA/sec
0	1	1	0	2 V/sec	4 mA/sec	1	1	1	0	512 V/sec	1024 mA/sec
0	1	1	1	4 V/sec	8 mA/sec						

Module calibration

All PAD-AO1 modules are within specifications when leaving the factory. The time interval for recalibration depends on environmental conditions. Typically, the calibration should be checked once a year.

Calibration performance: to be implemented

Read module configuration

Command: $(Addr)2\r

Response: !(Addr)(OutputRange)(BaudRate)(DataFormat)\r

!: Response leading code
Addr: Module address
(OutpRange): Define the analog output range, refered to the table *output range* on next page.
(BaudRate): Define communication baud rate, recommended '06' hex (= 9600 bps).
(DataForm): Define checksum and output data format
\r: Carriage return (0x0D)

Example: Command: $012\r

Response: !01300600\r

01:	module address
30:	output range (0 to 20 mA)
06:	baud rate (9600 bps)
00:	data format (engineering units, checksum disabled, immediate change)
\r:	carriage return

Read module name

Command: $(Addr)M\r

Response: !(Addr)(Name)\r

!:	Response leading code
Addr:	Module address
(Name):	Module name
\r:	Carriage return (0x0D)

Example: Command: $01M\r

Response: !01PAD-A01\r

01:	module address
PAD-A01:	module name is PAD-A01
\r:	carriage return

Read firmware version

Command: $(Addr)F\r

Response: !(Addr)(Firmware)\r

!:	Response leading code
Addr:	Module address
(Firmware):	Module firmware version (5 characters)
\r:	Carriage return (0x0D)

Example: Command: $01F\r

Response: !01B1.0\r

01:	module address 01
B1.0:	current firmware version is B1.0
\r:	carriage return

Send data to analog output

Command: #(Addr)(data)\r

#:	Command leading code
(Addr):	Module address
(data):	Output value (6 characters, refer to data format table)
\r:	Carriage return (0x0D)

Response:

>\r:	Valid command
?(Addr)\r:	Out of range (output will go to the most close value)
!\r:	Command ignore
no response:	Syntax error, communication error or address error

Example: Command: #0110.654\r

#:	command leading code
01:	module address
10.654:	current output = 10.654 mA
\r:	carriage return

Response: >\r: valid command

Save current output as startup value

Command: $(Addr)4\r

Response:
- !(Addr)\r: Valid command
- ?(Addr)\r: Out of range (output will go to the most close value)
- !\r: Command ignore
- no response: Syntax error, communication error or address error

Example:
- Command: #0110.654\r (set output as 10.654 mA for module 01)
- Response: >\r: valid command
- Command: $014\r (set the start-up analog output for module 01 as 10.654 mA)
- Response: !01\r valid command

Read back reset status

Command: $(Addr)5\r

Read back the reset status to detect the module watchdog failure.

Response:
- !(Addr)0\r: Module has not been reset since the last reset status read
- !(Addr)1\r: Module has been reset since the last reset status read
- ?(Addr)\r: Invalid command
- no response: Syntax error, communication error or address error

Example:
- Command: $015\r
- Response: !010\r valid command
  - !: response leading code
  - 01: module address
  - 0: module has not been reset since the last reset status read (value 1 at first time power on)
  - \r: carriage return

Read back last value

Command: $(Addr)6\r

Read back the latest analog output value (no measurement)

Response:
- !(Addr)(Data)\r: Module has not been reset since the last reset status read
- ?(Addr)\r: Invalid command
- no response: Syntax error, communication error or address error

Example:
- Command: #0110.654\r (set output as 10.654 mA for module 01)
- Response: >\r: valid command
- Command: $016\r
- Response: !0110.654\r
  - !: response leading code
  - 01: module address
  - 10.654: last output value was 10.654 mA for module 01
  - \r: carriage return

Notes

PAD-AO1

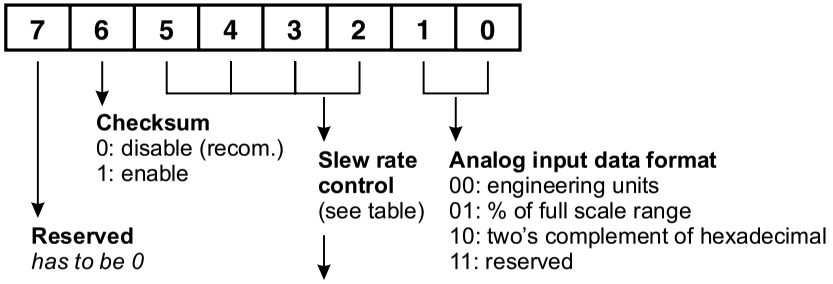
Nr:	Command					Response		
1	%	AA	AA(new)	FFBBDD	CR	!AA	CR	Set configuration
2	$	AA	2		CR	!AAFFBBDD	CR	Read configuration
3	$	AA	M		CR	!AAM	CR	Read module name
4	$	AA	F		CR	!AAV	CR	Read firmware version
5	#	AA	zz.zzz		CR	>	CR	Send data to analog output
6	$	AA	4		CR	!AA	CR	Save current output as startup value
7	$	AA	5		CR	!AAS	CR	Read back reset status
8	$	AA	6		CR	!AAzz.zzz	CR	Read back last value

AA	Address
z	Digit
M	Module name
V	Firmware version
S	Reset status

FF	Output range
30	0 to 20 mA
31	4 to 20 mA
32	0 to 10 V

BB	Baud rate
03	1200 bps
04	2400 bps
05	4800 bps
06	9600 bps
07	1920 bps
08	38400 bps
09	57600 bps
0A	115200 bps

DD Configuration code table



Code				Slew rate control	
5	4	3	2	Voltage	Current
0	0	0	0	immediate change	immediate change
0	0	0	1	0.0625 V/sec	0.125 mA/sec
0	0	1	0	0.125 V/sec	0.250 mA/sec
0	0	1	1	0.250 V/sec	0.500 mA/sec
0	1	0	0	0.500 V/sec	1 mA/sec
0	1	0	1	1 V/sec	2 mA/sec
0	1	1	0	2 V/sec	4 mA/sec
0	1	1	1	4 V/sec	8 mA/sec

Code				Slew rate control	
5	4	3	2	Voltage	Current
1	0	0	0	8 V/sec	16 mA/sec
1	0	0	1	16 V/sec	32 mA/sec
1	0	1	0	32 V/sec	64 mA/sec
1	0	1	1	64 V/sec	128 mA/sec
1	1	0	0	128 V/sec	256 mA/sec
1	1	0	1	256 V/sec	512 mA/sec
1	1	1	0	512 V/sec	1024 mA/sec

Module commands

Command summary for PAD-CNT2 module

Command	Syntax
GENERAL COMMANDS	
Set configuration	**%(OldAddr)(NewAddr)(InputRange)(BaudRate)(DataFormat)**
Read configuration	$(Addr)2
Read module name	$(Addr)M
Read firmware version	$(Addr)F
FUNCTIONAL COMMANDS	
Read counter or frequency x	**#(Addr)(Counter/FrequencyNo)**
Set input mode	$(Addr)B(Status)
Read input mode	$(Addr)B
Set high trigger level	$(Addr)1H(TriggerHighLevel)
Read high trigger level	$(Addr)1H
Set low trigger level	$(Addr)1L(TriggerLowLevel)
Read low trigger level	$(Addr)1L
Set maximum counter value	$(Addr)3(CounterNumber)(MaxValue)
Read maximum counter value	$(Addr)3
Set digital filter status	$(Addr)4(CounterNumber)(DigitalFilter)
Read digital filter status	$(Addr)
Set the counter status	$(Addr)5(Start/Stop)
Read the counter status	$(Addr)5
Reset counter	$(Addr)6(CounterNumber)
Read overflow status	$(Addr)7(CounterNumber)
Set gate control mode	$(Addr)A(Status)
Read gate control mode	$(Addr)A

Set configuration:

Command: %(OldAddr)(NewAddr)(InputType)(BaudRate)(DataFormat)\r

- %: Command leading code
- (OldAddr): The original / old module address. The factory default address of a module is 00. The address range is 0x00 to 0xFE.
- (NewAddr): The new module address (be careful to get no conflict with existing module addresses).
- (InputType): Define the input type (counter, frequency) refered to table *input type* on next page.
- (BaudRate): Define communication baud rate, recommended '06' hex (= 9600 bps).
- (DataForm.): Define checksum and frequency gate time, refered to the table *data format* on next page.
- \r: Carriage return (0x0D)

Response:
- !(Addr)\r The command to this address is valid.
- ?(Addr)\r The command is invalid, parameter values are invalid or attempt to change settings without short circuit on the default pins 21 and 22.

Example:

Command: %0102500600\r

- %: command leading code
- 01: old module address
- 02: new module address
- 50: input type (counter)
- 06: baud rate (9600)
- 00: data format (0.1 second frequency gate time, checksum disabled)
- \r: carriage return

Response: !02\r

Description: Address successfully changed from 0x01 to 0x02.

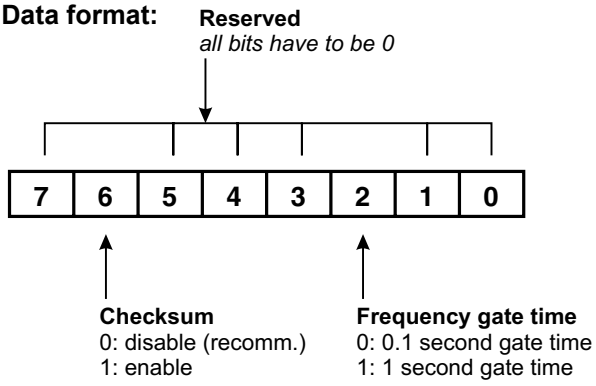
Module commands - configuration tables

Input type:

Code (hex)	Input type
50	Counter
51	Frequency

Valid for entire module - both channels in frequency or counter mode.

Data format:



Read module configuration

Command: $(Addr)2\r

Response: !(Addr)(InputType)(BaudRate)(DataFormat)\r

!:	Response leading code
(Addr):	Module address
(InputType):	Define the input type (counter, frequency)
(BaudRate):	Define communication baud rate, recommended ‘06’ hex (= 9600 bps)
(DataForm.):	Define checksum and frequency gate time
\r:	Carriage return (0x0D)

Example: Command: $012\r

Response: !01500600\r

!:	Response leading code
01:	module address
50:	input type (counter)
06:	baud rate (9600)
00:	data format (0.1 second frequency gate time, checksum disabled)
\r:	carriage return

Read module name

Command: $(Addr)M\r

Response: !(Addr)(Name)\r

!: Response leading code
(Addr): Module address
(Name): Module name
\r: Carriage return (0x0D)

Example: Command: $01M\r

Response: !01CNT-2\r
01: module address
CNT-2: module name is CNT-2
\r: carriage return

Read firmware version

Command: $(Addr)F\r

Response: !(Addr)(Firmware)\r

!: Response leading code
(Addr): Module address
(Firmware): Module firmware version (5 characters)
\r: Carriage return (0x0D)

Example: Command: $01F\r

Response: !01A1.2\r
01: module address 01
A1.2: current firmware version is A1.2
\r: carriage return

Read counter or frequency

Command: #(Addr)(Channel)\r

#: Command leading code
(Addr): Module address
(Channel): Channel of counter or frequency (0 or 1)
\r: Carriage return (0x0D)

Response: >(Addr)(Cnt/Frq)\r

>: Response leading code
(Cnt/Frq): 8 characters counter or frequency value (hex)
\r: Carriage return (0x0D)

Example: Command: $012\r

Response: !01500600

Command: #010\r

Response: >0000001D\r
>: Response leading code
0000001D: Counter 0 = 1D = 29 (dec)
\r: Carriage return (0x0D)

Example: Command: $072\r

Response: !07510600

Command: #071\r

Response: >0000001D\r
>: Response leading code
0000001D: Frequency 1 = 1D Hz = 29 Hz (dec)
\r: Carriage return (0x0D)

Set input mode

Command: $(Addr)B(Status)\r

$: Command leading code
(Addr): Module address
(Status): Input mode (isolated / non isolated) refered to table *set input mode*
\r: Carriage return (0x0D)

Response: !(Addr)\r Valid command

Example: Command: $01B2\r

$: Command leading code
01: Module address
B2: Set channel 0 non isolated and channel 1 isolated
\r: Carriage return (0x0D)

Response: !01\r Valid command

Input mode	Command	Channel 0	Channel 1
Input mode 0	$(Addr)B0	Non isolated (Pin 9 .. 11)	Non isolated (Pin 11 .. 13)
Input mode 1	$(Addr)B1	Isolated (Pin 1 .. 4)	Isolated (Pin 5 .. 8)
Input mode 2	$(Addr)B2	Non isolated (Pin 9 .. 11)	Isolated (Pin 5 .. 8)
Input mode 3	$(Addr)B3	Isolated (Pin 1 .. 4)	Non isolated (Pin 11 .. 13)

Read input mode

Command: $(Addr)B\r

Response: !(Addr)(Status)\r

!: Response leading code
(Addr): Module address
(Status): Input mode (isolated / non isolated) refered to table *set input mode*
\r: Carriage return (0x0D)

Example: Command: $01B\r

Response: !012\r (Channel 0 non isolated and channel 1 isolated)

Set high trigger level of non-isolated input

Command: $(Addr)1H(TrigLevel)\r

$: Command leading code
(Addr): Module address
(TrigLevel): 2 characters decimal value for high trigger level. The unit is 0.1 V and the range can be from 0.0 to 5.0 V. Default setting is 2.4 V.
\r: Carriage return (0x0D)

Response: !(Addr)\r Valid command

Example: Command: $011H2.3\r

$: Command leading code
01: Module address
1H2.3: Set the high trigger level to 2.3 V
\r: Carriage return (0x0D)

Response: !01\r Valid command

Read high trigger level of non-isolated input

Command: $(Addr)1H\r

Response: !(Addr)(TrigLevel)\r

!:	Response leading code
(Addr):	Module address
(TrigLevel):	2 characters decimal value for high trigger level
\r:	Carriage return (0x0D)

Example:

Command: $011H\r

Response: !0123\r

!:	Response leading code
01:	Module address
2.3:	The high trigger is 2.3 V
\r:	Carriage return (0x0D)

Set low trigger level of non-isolated input

Command: $(Addr)1L(TrigLevel)\r

$:	Command leading code
(Addr):	Module address
(TrigLevel):	2 characters decimal value for low trigger level. The unit is 0.1 V and the range can be from 0.0 to 5.0 V. Default setting is 0.8 V.
\r:	Carriage return (0x0D)

Response: !(Addr)\r Valid command

Example:

Command: $011L0.9\r

$:	Command leading code
01:	Module address
1L0.9:	Set the high trigger level to 0.9 V
\r:	Carriage return (0x0D)

Response: !01\r Valid command

Read low trigger level of non-isolated input

Command: $(Addr)1L\r

Response: !(Addr)(TrigLevel)\r

!:	Response leading code
(Addr):	Module address
(TrigLevel):	2 characters decimal value for high trigger level
\r:	Carriage return (0x0D)

Example:

Command: $011L\r

Response: !0109\r

!:	Response leading code
01:	Module address
0.9:	The high trigger is 0.9 V
\r:	Carriage return (0x0D)

Attention! The CNT module has no automatic overflow. You have to reset the counter at value FF FF FF FF!

Reset counter

Reset the counter to preset value and clear the overflow flag.

Command: $(Addr)6(Channel)\r

$:	Command leading code
(Addr):	Module address
(Channel):	Number of counter (0 or 1)
\r:	Carriage return (0x0D)

Response: !(Addr)\r Valid command

Example: Command: $0160\r Reset counter 0 of module 01

Response: !01\r Valid command

Set preset counter value

This is the counter value after “reset” or “power on” the module.
Preset value will be ignored in frequency mode.

Command: @(Addr)P(Channel)(PreValue)\r

@:	Command leading code
(Addr):	Module address
(Channel):	Number of counter (0 or 1)
(PreValue):	8-character value (hex)
\r:	Carriage return (0x0D)

Response: !(Addr)\r Valid command

Example: Command: @01P0FFFF0000\r
set preset value of counter 0 to FFFF0000

Response: !01\r Valid command

Read preset counter value

Command: @(Addr)G(Channel)\r

Response: !(Addr)(PreValue)\r

@:	Response leading code
(Addr):	Module address
(PreValue):	8-character value (hex)
\r:	Carriage return (0x0D)

Example: Command: @01G0\r

Response: !01FFFF0000\r
The preset value of counter 0 is FFFF0000

Set maximum counter value

Command: $(Addr)3(Channel)(MaxValue)\r

$:	Command leading code
(Addr):	Module address
(Channel):	Number of counter (0 or 1)
(MaxValue):	8-character value (hex)
\r:	Carriage return (0x0D)

Response: !(Addr)\r Valid command

Example: Command: $0130FFFF0000\r
set maximum value of counter 0 to FFFF0000

Response: !01\r Valid command

Read maximum counter value

Command: $(Addr)3(Channel)\r

Response: !(Addr)(MaxValue)\r

$:	Response leading code
(Addr):	Module address
(MaxValue):	8-character value (hex)
\r:	Carriage return (0x0D)

Example: Command: $01G0\r

Response: !01FFFF0000\r
The maximum value of counter 0 is FFFF0000

Set digital filter status

The digital filter is disable in frequency mode. Valid for both none-isolated & isolated inputs.

Command: $(Addr)4(DigFilter)\r

$:	Command leading code	
(Addr):	Module address	
(DigFilter):	0	Digital filter disabled
	1	Digital filter enabled
\r:	Carriage return (0x0D)	

Response: !(Addr)\r Valid command

Example: Command: $0141\r
Digital filter enabled

Response: !01\r Valid command

Read digital filter status

Command: $(Addr)4\r

Response: !(Addr)(DigFilter)\r

$:	Response leading code	
(Addr):	Module address	
(DigFilter):	0	Digital filter disabled
	1	Digital filter enabled
\r:	Carriage return (0x0D)	

Example: Command: $014\r

Response: !010\r
Digital filter is disabled

Set the counter status

Command: $(Addr)5(Counter)(Start/Stop)\r

$:	Command leading code	
(Addr):	Module address	
(Counter):	Select counter 0 or 1	
(Start/Stop):	0	Stop counter
	1	Start counter
\r:	Carriage return (0x0D)	

Response: !(Addr)\r Valid command

Example: Command: $0150\r
Stop counter

Response: !01\r Valid command

Read the counter status

Command: $(Addr)5\r

Response: !(Addr)(Start/Stop)\r

$:	Response leading code	
(Addr):	Module address	
(Start/Stop):	0	Stop counter
	1	Start counter
\r:	Carriage return (0x0D)	

Example: Command: $015\r

Response: !010\r
Counter stopped

Read overflow status

Command: $(Addr)7(Channel)\r

$:	Command leading code
(Addr):	Module address
(Channel):	Channel (0 or 1)
\r:	Carriage return (0x0D)

Response: !(Addr)(Status)\r

!:	Command leading code
(Addr):	Module address
(Status):	0 for no overflow; 1 for overflow
\r:	Carriage return (0x0D)

Example: Command: $0170\r

Response: !011\r

!:	Response leading code
01:	Module address
1:	Counter 0 is overflow
\r:	Carriage return (0x0D)

Set gate control mode

This command will be ignored in frequency mode.

Command: $(Addr)A(GateContr)\r

$:	Command leading code
(Addr):	Module address
(GateContr):	Gate control mode (1 character refered to *gate control* table)
\r:	Carriage return (0x0D)

Response: !(Addr)\r Valid command

Example: Command: $01A0

$:	Command leading code
01:	Module address
A0:	Gate is low active
\r:	Carriage return (0x0D)

Response: !01\r Valid command

Command	Gate status
$(Addr)A0	low active
$(Addr)A1	high active
$(Addr)A2	disabled (= always active)

Read gate control mode

Command: $(Addr)A\r

Response: !(Addr)(GateContr)\r

!:	Command leading code
(Addr):	Module address
(GateContr):	Gate control mode (1 character refered to *gate control* table)
\r:	Carriage return (0x0D)

Example: Command: $01A\r

Response: !010\r

!:	Response leading code
01:	Module address
0:	Gate is low active
\r:	Carriage return (0x0D)

PAD-CNT2

Nr:	Command					Response		
1	%	AA	AA(new)	TTBBDD	CR	!AA	CR	Set configuration
2	$	AA	2		CR	!AATTBBDD	CR	Read configuration
3	$	AA	M		CR	!AAM	CR	Read module name
4	$	AA	F		CR	!AAV	CR	Read firmware version
5	#	AA	N		CR	>hhhhhhhh	CR	Read counter or frequency
6	$	AA	BS		CR	!AA	CR	Set input mode
7	$	AA	B		CR	!AAS	CR	Read input mode
8	$	AA	1H	zz	CR	!AA	CR	Set high trigger level
9	$	AA	1H		CR	!AAzz	CR	Read high trigger level
10	$	AA	1L	zz	CR	!AA	CR	Set low trigger level
11	$	AA	1L		CR	!AAzz	CR	Read low trigger level
12	$	AA	3	Nhhhhhhhh	CR	!AA	CR	Set maximum counter value
13	$	AA	3		CR	!AAhhhhhhhh	CR	Read maximum counter value
14	$	AA	4	Nz	CR	!AA	CR	Set digital filter status
15	$	AA	4		CR	!AAz	CR	Read digital filter status
16	$	AA	5	z	CR	!AA	CR	Set the counter status
17	$	AA	5		CR	!AAz	CR	Read the counter status
18	$	AA	6	N	CR	!AA	CR	Reset counter
19	$	AA	7	N	CR	!AAO	CR	Read overflow status
20	$	AA	A	G	CR	!AA	CR	Set gate control mode
21	$	AA	A		CR	!AAG	CR	Read gate control mode
22	@	AA	P	Nhhhhhhhh	CR	!AA	CR	Set preset counter value
23	@	AA	G		CR	!AAhhhhhhhh	CR	Read preset counter value

AA	Address
M	Module name
V	Firmware version
z	Digit
h	HEX value

BB	Baud rate
03	1200 bps
04	2400 bps
05	4800 bps
06	9600 bps
07	1920 bps
08	38400 bps
09	57600 bps
0A	115200 bps

TT	Input type
50	Counter
51	Frequency

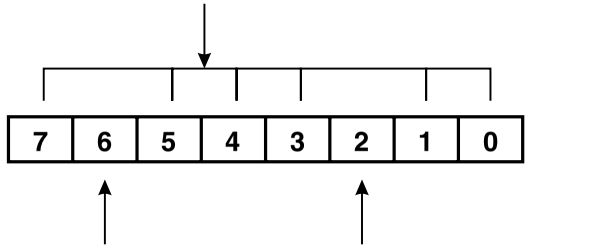
N	Channel
0	Channel 0 of module
1	Channel 1 of module

Input mode Channel 0	S	Input mode Channel 1
Non isolated	0	Non isolated
Isolated	1	Isolated
Non Isolated	2	Isolated
Isolated	3	Non isolated

O	Overflow
0	No counter overflow
1	Counter overflow

G	Gate control
0	Low active
1	High active
2	Disabled

DD	Configuration code table



Module commands

Command summary for PAD-DI8 module

Command	Syntax
GENERAL COMMANDS	
Set configuration	**%(OldAddr)(NewAddr)40(BaudRate)(DataFormat)**
Read configuration	$(Addr)2
Read module name	$(Addr)M
Read firmware version	$(Addr)F
FUNCTIONAL COMMANDS	
Read digital input data	**$(Addr)6**
Read latch value of digital input	$(Addr)L(LatchStatus)
Clear latch input	$(Addr)C
Synchronized sampling	#**
Read synchronized data	$(Addr)4

Set configuration:

Command: %(OldAddr)(NewAddr)(InputRange)(BaudRate)(DataFormat)\r

%: Command leading code
(OldAddr): The original / old module address. The factory default address of a module is 00. The address range is 0x00 to 0xFE.
(NewAddr): The new module address (be careful to get no conflict with existing module addresses).
(InputRange): Type 40 for DI module
(BaudRate): Define communication baud rate, recommended ‘06’ hex (= 9600 bps).
(DataForm.): Define dataformat (checksum) refered to the table *data format* on next page.

Response: !(Addr)\r The command to this address is valid.
?(Addr)\r The command is invalid, parameter values are invalid or attempt to change settings without short circuit on the default pins 21 and 22.

Example: Command: %0102400600\r

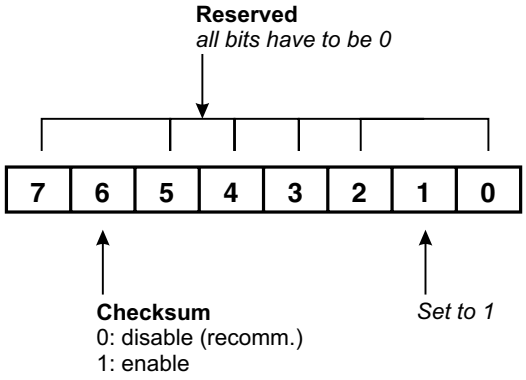
%: command leading code
01: old module address
02: new module address
40: DI module
06: baud rate (9600)
00: data format (checksum disabled)
\r: carriage return

Response: !02\r

Description: Address successfully changed from 0x01 to 0x02

Module commands - configuration tables

Data format:



Input			
First data		Second data	
DI (0 to 7)	00 to FF	00	00
	first byte	second byte	third byte

Read module configuration

Command: $(Addr)2\r

Response: !(Addr)(InputRange)(BaudRate)(DataFormat)\r

!: Response leading code
(Addr): Module address
(InputRange): Type 40 for DI module
(BaudRate): Define communication baud rate, recommended '06' hex (= 9600 bps)
(DataForm.): Define dataformat (checksum)
\r: Carriage return (0x0D)

Example: Command: $012\r

Response: !01400602\r
!: response leading code
01: module address
40: DI module
06: baud rate (9600)
02: data format (checksum disabled)
\r: carriage return

Read module name

Command: $(Addr)M\r

Response: !(Addr)(Name)\r

!: Response leading code
(Addr): Module address
(Name): Module name
\r: Carriage return (0x0D)

Example: Command: $01M\r

Response: !01PAD-DI8\r
01: module address
PAD-DI8: module name is PAD-DI8
\r: carriage return

Read firmware version

Command: $(Addr)F\r

Response: !(Addr)(Firmware)\r

!:	Response leading code
(Addr):	Module address
(Firmware):	Module firmware version (5 characters)
\r:	Carriage return (0x0D)

Example: Command: $01F\r

Response: !01B1.3\r

!:	response leading code
01:	module address 01
B1.3:	current firmware version is B1.3
\r:	carriage return

Read digital input data

Command: $(Addr)6\r

Response: !(InputData)\r

!:	Response leading code
(InputData):	Input data refered to the table *Input data format* 3 bytes - information only in the first byte
\r:	Carriage return (0x0D)

Example: Command: $016\r

Response: !0F0000\r

!:	response leading code
0F0000:	Input 0 to 3 are high (first byte = 0F = 00001111)
\r:	carriage return

Read latch value of digital input

Triggering at rising/falling edge and storing them.

Command: $(Addr)L(Latch)\r

$:	Command leading code
(Addr):	Module address
L:	Command for read latched digital input
(Latch):	Select latch at rising edge (1) or falling edge (0)
\r:	Carriage return (0x0D)

Response: !(LatchData)\r

!:	Response leading code
(LatchData):	Input channel is latched (1) or input channel is not latched (0)
\r:	Carriage return (0x0D)

Example:

Read:	Latch:	
!000000	!000000	
!010000	!010000	
!000000	!010000	Rising edge stays stored until “clear latch” command

Command: $01L1\r

Response: !010000\r

!:	response leading code
010000:	Input 0 is high
\r:	carriage return

Clear latch input

Command: $(Addr)C\r

$:	Command leading code
(Addr):	Module address
C:	Command for clear latched digital input
\r:	Carriage return (0x0D)

Response: !(Addr)\r

Synchronized sampling

This command will store the current input values for all channels. As this command uses no module address, it has an effect on all PAD modules within the system. The values are stored within each module in a temporary register.

Command: #**\r

Response: no response; to read out the values, use the following command.

Read synchronized data

Command: $(Addr)4\r

Response: !(S)(Data)\r

!:	Command leading code
S:	Readout state (1 = first read out; 0 = re-read of old values)
Data:	Stored input values for channel 0 to 7

Example: Command: $014\r

Response: !1000000\r

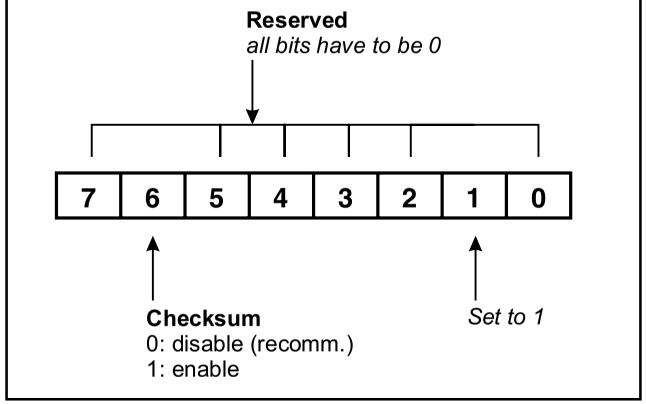
!:	response leading code
000000:	readed data
\r:	carriage return

Notes

PAD-DI8

Nr:	Command						Response		
1	%	AA	AA(new)	40BBDD		CR	!AA	CR	Set module configuration
2	$	AA	2			CR	!AA40BBDD	CR	Read configuration
3	$	AA	M			CR	!AAM	CR	Read module name
4	$	AA	F			CR	!AAV	CR	Read firmware version
5	#	AA	6			CR	!{Input}	CR	Read digital input data
6	$	AA	L	S		CR	!{Input}	CR	Read latch value of digital input
7	$	AA	C			CR	!AA	CR	Clear latch value
8	#	**				CR			Synchronized sampling
9	$	AA	4			CR	!zRhhhhhh	CR	Read synchronized data

DD	Configuration code table



Input			
First data		Second data	
DI (0 to 7)	00 to FF	00	00

AA	Address
M	Module name
V	Firmware version
z	Digit
h	HEX value

BB	Baud rate
03	1200 bps
04	2400 bps
05	4800 bps
06	9600 bps
07	1920 bps
08	38400 bps
09	57600 bps
0A	115200 bps

S	Latch status
0	Latch is low
1	Latch is high

R	Readout state
0	Re-read out of old value
1	First read out

Module commands

Command summary for PAD-DO7 module

Command	Syntax
GENERAL COMMANDS	
Set configuration	**%(OldAddr)(NewAddr)(InputRange)(BaudRate)(DataFormat)**
Read configuration	$(Addr)2
Read module name	$(Addr)M
Read firmware version	$(Addr)F
FUNCTIONAL COMMANDS	
Set value of digital output	**#(Addr)(MultichannelOn/Off)(Value)**
Set digital output	@(Addr)(Outputdata)

Set configuration:

Syntax: %(OldAddr)(NewAddr)(InputRange)(BaudRate)(DataFormat)

Description: % Command leading code
(OldAddr): The original / old module address. The factory default address of a module is 00. The address range is 0x00 to 0xFE.
(NewAddr): The new module address (be careful to get no conflict with existing module addresses).
(Range): Type 40 for DO module
(BaudRate): Define communication baud rate, recommended '06' hex (= 9600 bps).
(DataForm.): Define checksum and output data format, refered to the table *data format* on next page.

Response: !(Addr)\r The command to this address is valid.
?(Addr)\r The command is invalid, parameter values are invalid or attempt to change settings without short circuit on the default pins 21 and 22.

Example: Command: %0102400600\r

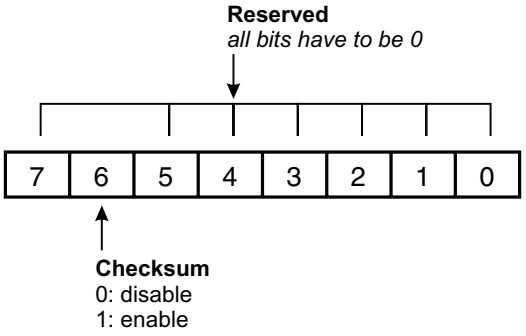
%: command leading code
01: old module address
02: new module address
40: DO module
06: baud rate (9600)
00: data format (checksum disabled)
\r: carriage return

Response: !02\r

Description: Address successfully changed from 0x01 to 0x02

Module commands - configuration tables

Data format:



Read module configuration

Command: $(Addr)2\r

Response: !(Addr)(InputRange)(BaudRate)(DataFormat)\r

!:	Response leading code
(Addr):	Module address
(InputRange):	Type 40 for DO module
(BaudRate):	Define communication baud rate, recommended ‘06’ hex (= 9600 bps)
(DataForm.):	Define dataformat (checksum)
\r:	Carriage return (0x0D)

Example: Command: $012\r

Response: !01400600\r

!:	response leading code
01:	module address
40:	DO module
06:	baud rate (9600)
00:	data format (checksum disabled)
\r:	carriage return

Read module name

Command: $(Addr)M\r

Response: !(Addr)(Name)\r

!:	Response leading code
(Addr):	Module address
(Name):	Module name
\r:	Carriage return (0x0D)

Example: Command: $01M\r

Response: !01PAD-DO7\r

01:	module address
PAD-DO7:	module name is 0815
\r:	carriage return

Read firmware version

Command: $(Addr)F\r

Response: !(Addr)(Firmware)\r

!:	Response leading code
(Addr):	Module address
(Firmware):	Module firmware version (4 characters)
\r:	Carriage return (0x0D)

Example: Command: $01F\r

Response: !01B1.2\r

!:	response leading code
01:	module address 01
B1.2:	current firmware version is B1.2
\r:	carriage return

Set value of digital output:

Command: #(Addr)(MultichannelOn/Off)(Value)\r

#	Command leading code	
(Addr)	Current module address (0x00 to 0xFE).	
(Multichannel)	00:	setting value for all channels (Multichannel)
	1x:	setting value for channel x (x = 0 to 6)(Singlechannel)
(Value)	Multichannel:	00 to 7F hex
	Singlechannel (1x):	00 turn channel x off
		01 turn channel x on
\r:	carriage return	

Response: >\r

Example: Command: #021001\r
Module 02; channel 0 on

Response: >\r

Command: #021601\r
Module 02; channel 6 on

Response: >\r

Set digital output:

Command: @(Addr)(Data)\r

@	Command leading code
(Addr)	Current module address (0x00 to 0xFE).
(Data)	Output data from 0x00 to 0x7F (7 bit)
\r:	carriage return

Response: >\r

Example: Command: #027F\r
Set module 02; output data 0x7F

Response: >\r

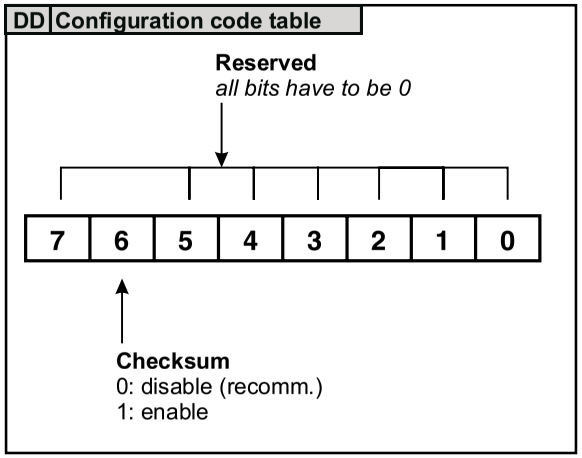
PAD-DO7

Nr:	Command					Response		
1	%	AA	AA(new)	40BBFF	CR	!AA	CR	Set module configuration
2	$	AA	2		CR	!AA40BBFF	CR	Read configuration
3	$	AA	M		CR	!AAM	CR	Read module name
4	$	AA	F		CR	!AAV	CR	Read firmware version
5	#	AA	{Output}		CR	>	CR	Set value of digital output
6	@	AA	hh		CR	>	CR	Set digital output

AA	Address
M	Module name
h	HEX value
V	Firmware version

BB	Baud rate
03	1200 bps
04	2400 bps
05	4800 bps
06	9600 bps
07	1920 bps
08	38400 bps
09	57600 bps
0A	115200 bps

DD Configuration code table



Output			
Multi channel			
00	00 to 7F		
Flag	Data		
Single channel			
10	00	=ch 0	off
10	01	=ch 0	on
16	00	=ch 6	off
16	01	=ch 6	on

Module commands

Getting SNo and settings

Commando to module:
Identifier from host: SNoIdent or GlobIdent
DLC = 1
Byte 0: 0x00

If GlobIdent is used: response on all modules:

Answer: from module
Identifier: SNoIdent
Byte 0: 0x00
Byte 1 to 3: CAN_Address
Byte 4: Module code
Byte 5: RangeCode Ch0
Byte 6: Module Information

0:	No Sensor block connected[1)]
1:	Fixed Block (all Ranges identical Range)
2:	Inteligent Sensor Block
3:	Ranges from Module

Byte 7: HW-Revision, SW Revision (first two numbers)

Example: if Rev. 1.53 -> Answer = 0x15

[1)] Means no external E²PROM, no CJC detected and no hardwired connector block detected.

Read sample rate and data format

Commando to module:
Identifier from host: SNoIdent or GlobIdent
DLC = 1
Byte 0: 0x00

Answer: from module
Identifier: SNoIdent
Byte 0: 0x01
Byte 1: Sample rate code:

0:	Off (no measurement data are send)
1:	Synchronized with Synch identifier
2:	0.01 Hz
3:	0.02 Hz
4:	0.05 Hz
5:	0.1 Hz
6:	0.2 Hz
7:	0.5 Hz
8:	1 Hz
9:	2 Hz
A:	5 Hz
B:	10 Hz
FF:	Max. sample rate (around 12 Hz at EPAD)

Byte 2 .. 4: CAN_Address:
Byte 5: Data Format

Bit 0:	0: Intel (16 Bit); 1: Motorola (16 Bit)
Bit 1:	0: Standard identifier; 1: Extended identifier
Bit 2:	0: Data identifier is SNoIdent (Not allowed if Standard Identifier is used, Bit 1 automatically set to “1”) 1: Data identifier is CAN_Address 1)
Bit 3:	0: Module sends no SyncIdent 1: Module sends SyncIdent (only valid, if Sample Rate code > 1)

Note: DLC of message with SyncIdent = 0 (no data)
No SyncIdentMessage is send if sample rate code = FF!

Write sample rate and data format

Commando to module:
Identifier from host: SNoIdent or GlobIdent
DLC = 6
Byte 0: 0x02

Byte 1: Sample rate code:

0:	Off (no measurement data are send)
1:	Synchronized with Synch identifier
2:	0.01 Hz
3:	0.02 Hz
4:	0.05 Hz
5:	0.1 Hz
6:	0.2 Hz
7:	0.5 Hz
8:	1 Hz
9:	2 Hz
A:	5 Hz
B:	10 Hz
FF:	Max. sample rate (around 12 Hz at EPAD)

Byte 2 .. 4: CAN_Address:

Byte 5: Data Format

Bit 0:	0: Intel (16 Bit); 1: Motorola (16 Bit)
Bit 1:	0: Standard identifier; 1: Extended identifier
Bit 2:	0: Data identifier is SNoIdent (Not allowed if Standard Identifier is used, Bit 1 automatically set to “1”) 1: Data identifier is CAN_Address 1)
Bit 3:	0: Module sends no SyncIdent 1: Module sends SyncIdent (only valid, if Sample Rate code > 1)

Note: DLC of message with SyncIdent = 0 (no data)
No SyncIdentMessage is send if sample rate code = FF!

Answer from module: same like command “Read sample rate and data Format”.

Get available range code

Commando to module:
Identifier from host: SNoIdent
DLC = 1
Byte 0: 0x03

Answer: From module
Identifier: SNoIdent

Byte 0: 0x03
Byte 1: Available Range Code 1
Byte 2: Available Range Code 2
Byte 3: Available Range Code 3
Byte 4: Available Range Code 4
Byte 5: Available Range Code 5
Byte 6: Available Range Code 6
Byte 7: Available Range Code 7

Range Code 0xFF means : No range available. For example if connector block with hard wired identification Type K: Byte 1: 0x0F; Byte 2 to Byte 7: 0xFF

Get selected range code

Commando to module:
Identifier from host: SNoIdent
DLC = 1
Byte 0: 0x04

If ChnBit = 0, then response from CH 0 .. 3,
If ChnBit = 1, then response from CH 4 .. 7,

Answer: From module
Identifier: SNoIdent

Byte 0: 0x04
Byte 1: Range Code Ch0 (or CH4 if ChnBit = 1)
Byte 2: Range Code Ch1 (or CH5 if ChnBit = 1)
Byte 3: Range Code Ch2 (or CH6 if ChnBit = 1)
Byte 4: Range Code Ch3 (or CH7 if ChnBit = 1)

Set range code

Commando to module:
Identifier from host: SNoIdent

Byte 0 : 0x05
Byte 1: Range Code Ch0 (or CH4 if ChnBit = 1)
Byte 2: Range Code Ch1 (or CH5 if ChnBit = 1)
Byte 3: Range Code Ch2 (or CH6 if ChnBit = 1)
Byte 4: Range Code Ch3 (or CH7 if ChnBit = 1)

DLC=5

Answer: Same like on command „Get selected Range Code“.

Set LED

Commando to module:
Identifier from host: SNoIdent or GlobeIdent

Byte 0: 0x06
Byte 1: 0: LED off
1: LED on
2: LED 7 times inverting (blinking)
3: LED blinking, when communication

Change CAN baud rate

Commando to module:
Identifier from host: SNoIdent or GlobeIdent

Byte 0: 0x07
Byte 1:

0:	50 kBaud
1:	100 kBaud
2:	125 kBaud
3:	200 kBaud
4:	250 kBaud
5:	333.33 kBaud
6:	500 kBaud
7:	1000 kBaud

Notes