Instruction manual

# IQ+FLOW®
# Digital mass flow / pressure Controllers for gases

Doc. no.: 9.17.045K Date: 26-07-2011

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

Bronkhorst®
HIGH-TECH

Even though care has been taken in the preparation and publication of the contents of this manual, we do not assume legal or other liability for any inaccuracy, mistake, mis-statement or any other error of whatsoever nature contained herein. The material in this manual is for information purposes only, and is subject to change without notice.

Bronkhorst High-Tech B.V.
July 2011

# Warranty

The products of Bronkhorst High-Tech B.V. are warranteed against defects in material and workmanship for a period of three years from the date of shipment, provided they are used in accordance with the ordering specifications and the instructions in this manual and that they are not subjected to abuse, physical damage or contamination. Products that do not operate properly during this period may be repaired or replaced at no charge. Repairs are normally warranteed for one year or the balance of the original warranty, whichever is the longer.
See also paragraph 9 of the Conditions of sales.

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

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

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

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

# Introduction

The Bronkhorst High-Tech IQ$^+$FLOW® mass flow / pressure instrument for gases is an accurate device for controlling gas flows up to 2000 sccm, virtually independent of pressure and temperature changes. The smallest MFC on the market can be applied for applications with pressure conditions up to 10 bars (145 psi) and temperatures between 0 to 50 °C (32 to 122 °F).

# Short-Form Operation Instruction

Before installing the IQ$^+$FLOW® instrument it is important to read the attached label and check:

- Flow rate
- Fluid to be metered
- Up- and downstream pressures
- Input/output signals

✓ Check the red-coloured sticker and make sure the test pressure is in agreement with normal safety factors for your application.

✓ Check if the piping system is clean. For absolute cleanliness always install filters to assure a clean, moisture- and oil- free gas stream.

✓ Install the IQ$^+$FLOW® instrument in the line and tighten the fittings according to the instructions of the supplier of the fittings.

✓ Check the system for leaks before applying fluid pressure.

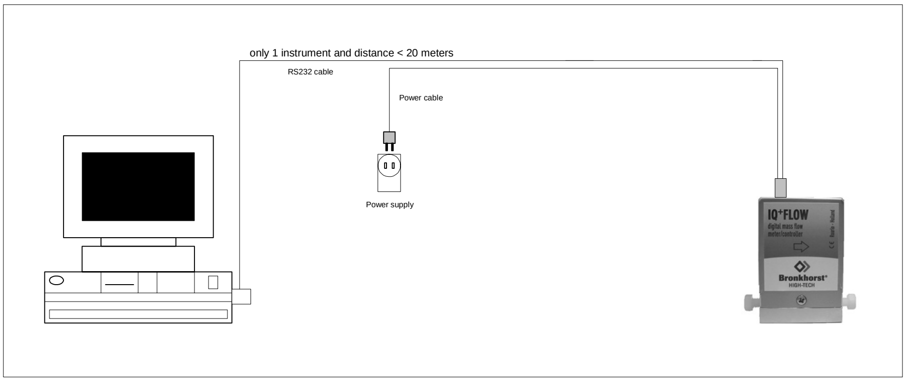
✓ Electrical connections must be made with a standard cable or according to the hook-up diagram in the back of this manual.

# Short from start-up

✓ Install instrument in your process or test bench.

✓ Provide instrument with correct pressure(s).

# 1. Installation for evaluation, using the special cable (7.03.385)




- ✓ Connect the Personal computer to the IQ$^+$FLOW® instrument with RS232 cable.
- ✓ Install the software on the personal computer (information about installing software on the personal computer can be found on the disc).
- ✓ Send a setpoint to the instrument and check the measured value.
- ✓ Let the instrument warm-up for 5 minutes for best accuracy.
- ✓ Your IQ$^+$FLOW® instrument is now ready for operation.

# 2. Process installation

For installation, please consult the hook-up diagrams.

# 1. Installation

## 1.1. Receipt of equipment

Check the outside packing box for damage incurred during shipment. Should the packing box be damaged, then the local carrier must be notified at once regarding his liability, if so required. At the same time a report should be submitted to:

BRONKHORST HIGH-TECH B.V.
RUURLO HOLLAND

If applicable, otherwise contact your distributor.
Remove the envelope containing the packing list; carefully remove the equipment from the packing box. Do not discard spare or replacement parts with the packing material and inspect the contents for damaged or missing parts.

## 1.2. Mounting

For IQ$^+$ FLOW® the preferred position is horizontal. Avoid installation in close proximity of mechanic vibration and/or heat sources. The housing of the instrument is according to class IP40, which means that the instrument is suitable for indoor (dry) applications, like laboratories or well protected (OEM) housings.

## 1.3. In-Line filters (optional)

Fluids to be measured should be absolutely free of dirt, oil, moisture and other particles. It is recommended to install an in-line filter upstream of the mass flow controller, and if backflow can occur, a downstream filter is recommended too. Be aware of the pressure drop caused by the filters.

## 1.4. Fluid connections

Bronkhorst High-Tech controllers are equipped with compression type fittings or
10-32 UNF. For leak tight installation of compression type fittings be sure that the tube is inserted to the shoulder in the fitting body and that no dirt or dust is present on tube, ferrules or fittings. Tighten the nut fingertight; while holding the instrument, then tighten the nut 1 turn. If applicable follow the guidelines of the supplier of the fittings. Special types of fittings are available on request.

*** Note**: Always check your system for leaks, before applying fluid pressure. Especially if toxic, explosive or other dangerous fluids are used.

## 1.5. Electrical connections

For product evaluation use the special cable (7.03.385) (enclosed in the TESTCASE) For process installation, hook-up diagrams are enclosed in the back of this manual.

## 1.6. Software

A CD with PC software is included for interfacing the IQ+FLOW® instrument using a personal computer.

Software included:

- **FlowDDE**
  for communication between IQ+FLOW® instrument and Microsoft Windows applications
- **FlowView**
  easy user interface for control of IQ+FLOW® instrument
- **FlowPlot**
  user interface for control, monitoring and advanced settings
  (preliminary demo-software)

### 1.6.1. Getting started

- ✓ Check Operating System language settings: see System requirements below
- ✓ Install FlowDDE32, FlowPlot and FlowView by running setup.exe from their directories
- ✓ Connect the RJ45 connector to the instrument and the PC connector to the PC
- ✓ Turn on instrument power supply by putting the adapter in the socket.
- ✓ Open FlowDDE. Select the right com-port in menu *Communication – Communication Settings* (F2). Open communication (F3).
- ✓ Open FlowView (easy) or FlowPlot (advanced).

### 1.6.2. System requirements

Operating System language requirements:
(Number) Decimal symbol: **.** (point)
(Number) Digit grouping symbol: **,** (comma)

These language settings can be checked and changed (if required) via “Regional options” in Windows’ Control Panel.

The digit-grouping symbol of currency must always differ from the numbers decimal symbol (see Microsoft Q198098). In this situation you have to check that the digit grouping symbol of currency does not contain a . (point). If so, choose any other character. Having equal symbols is not valid and will cause problems, not only for this program but for other programs as well.

Hardware requirements (Personal Computer):

Advised:
Pentium PC - 300 MHz / 64 MB internal RAM
20 MB free disk space for program files
800 x 600 x 256 color display
RS232 port with FIFO buffers for RS232/FLOW-BUS interface at 38K4 BPS
Operating system: Windows 95 UK with Internet Explorer V4.0 UK (or higher), Windows 98 UK, Windows NT 4.0 UK (SP6) or Windows 2000 UK.

Minimum requirements:
Pentium PC - 166 MHz / 32 MB internal RAM
20 MB free disk space for program files
800 x 600 x 256 color display
RS232 port with FIFO buffers for RS232/FLOW-BUS interface at 38K4 BPS
Operating system: Windows 95 UK with Internet Explorer V4.0 UK (or higher) or Windows NT 4.0 UK (SP6) or Windows 2000 UK

## 1.7. Caution

Each mass flow controller is pressure tested to at least 1.5 times the working pressure of the process conditions stipulated by the customer, with a minimum of 8 bar.

The tested pressure is stated on the IQ+FLOW® instrument with a red coloured sticker. Check test pressure before installing in the line.
If the sticker is not available or the test pressure is incorrect, the instrument should not be mounted in the process line and be returned to the factory.
Each instrument is helium leak tested.

## 1.8. Supply pressure

Do not apply pressure until electrical connections are made. When applying pressure to the system, take care to avoid pressure shocks in the system and increase pressure gradually.

## 1.9. System purging

If explosive gases are to be used, purge the process with inert dry gas like Nitrogen, Argon etc. for at least 30 minutes.
In systems with corrosive or reactive fluids, purging with an inert gas is absolutely necessary, because if the tubing has been exposed to air, introducing these fluids will tend to clog up or corrode the system due to a chemical reaction with oxygen or moist air.
Complete purging is also required to remove such fluids from the system before exposing the system to air.
It is preferred not to expose the system to air, when working with these corrosive fluids.

## 1.10. Seals

Bronkhorst High-Tech B.V. has gathered a material compatibility chart from a number of sources believed to be reliable. However, it is a general guide only. Operating conditions may substantially change the accuracy of this guide.
Therefore there is no liability for damages accruing from the use of this guide.
The customers application will demand its own specific design or test evaluation for optimum reliability. So check if the seals like O-rings, epoxy, plunger and packing gland of capillary are correct for the process.

## 1.11. Equipment storage

The equipment should be stored in its original packing in a cupboard warehouse or similar. One should take care the equipment is not exposed to excessive temperatures or humidity.

## 1.12. Electromagnetic compatibility

The instruments described in this manual carry the CE-mark.
Therefore they have to comply with the EMC requirements as are valid for these instruments. However compliance with the EMC requirements is not possible without the use of proper cables and connector/gland assemblies.
For good results Bronkhorst High-Tech B.V. can provide standard cables.

## 1.13. Maintenance

### 1.13.1. General

No routine maintenance is required on the meters or controllers. Units may be flushed with clean, dry inert gas. In case of severe contamination the instrument must be returned to factory.

### 1.13.2. Calibration procedure

All instruments are factory calibrated. For re-calibration or re-ranging contact supplier or factory.

# 2. Operation

## 2.1. General

The Bronkhorst High-Tech instruments are designed in such a way that they will meet user process requirements in the best possible way.

Basically all digital meters/controllers are powered with only +15 Vdc (on request also +24 Vdc). When providing your own power supply be sure that voltage and current ratings are according to the specifications of the instrument(s) and furthermore that the source is capable of delivering enough energy to the instrument(s).
Cable wire diameters should be sufficient to carry the supply current and voltage losses must be kept as low as possible. When in doubt: consult factory.
Operation can be done by means of the analog interface or digital with a Personal Computer.
Also the micro push-button switch and the LED's on top of the instrument can be used for manual operation of some options.
The green LED will indicate in what mode the instrument is active.
The red LED will indicate error/warning situations.

## 2.2. Power and warm-up

Before switching on power check if all connections have been made according to the hook-up diagram which belongs to the instrument.
Check fluid connections and make sure there is no leakage. If needed purge the system with a proper fluid.
For a gas instrument only purging with gases is allowed.
Turn on power and allow at least 5 minutes to warm up. During warm-up period, fluid pressure may either be on or off.

## 2.3. Start-up

Turn on fluid supply gently. Avoid pressure shocks, and bring the instrument gradually up to the level of the actual operating conditions. Also switch off fluid supply gently.

## 2.4. Operating conditions

Each instrument has been calibrated and adjusted for customer process conditions.
Controllers or valves may not operate correctly if process conditions vary too much, because of the restriction of the orifice in the valve.
For flowmeters performance and accuracy may be affected tremendously if physical fluid properties such as heat capacity and viscosity change due to changing process conditions.

## 2.5. Instrument performance

### 2.5.1. Sensors

Assuming that the transfer function of a system is an exponential shaped curve, the time constant is defined as follows:
time constant = time for the signal to reach 63.2 % of its final output value. Approx. five time constants is the time to reach the final value.
The IQ$^+$FLOW® sensor has a time constant of 300 milliseconds.

### 2.5.2. Controllers

The dynamic response of a controller is factory set. Standard settling time is defined as the time to reach the setpoint (and stay) within ±2% of the initial setpoint.
The control mode is factory set in such a way that after a step change, there will be little overshoot.

## 2.6. Digital operation

The IQ$^+$FLOW® instrument is set for digital operation (in RS232 mode) and ready for use in combination with a (personal) computer. By means of the micro push-button switch some important actions for the instrument can be selected/started. (see also chapter 13).

These functions are:

- reset (instrument firmware-program reset)
- auto-zeroing (remove zero-drift)
- restore factory settings (in case of accidentally changing of the settings)

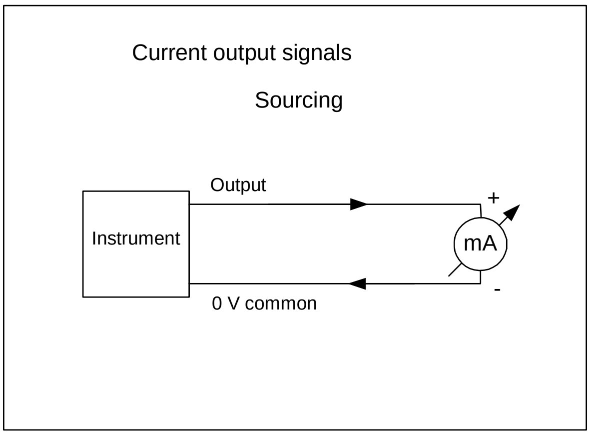
Furthermore the digital IQ$^+$FLOW® series are featuring counter and alarm functions, enhanced response time setting capability and the possibility to store up to 8 calibrations curves (for different fluids or process conditions).

## 2.7. Analog operation

Digital instruments can be operated with analog signals through the RJ45 modular jack (see hook-up diagram). The electronic p.c.board is set for one of the following output (and corresponding input) signals:

By default the IQ$^+$FLOW® is set to signal mode A.

| **Signal Mode** | **Output (sensor) Signal** | **Input (setpoint) signal** |
|---|---|---|
| A | 0…5 Vdc | 0…5 Vdc |
| B | 0…10 Vdc | 0…10 Vdc |
| F | 0…20 mA (sourcing) | 0…20 mA (sinking) |
| G | 4…20 mA (sourcing) | 4…20 mA (sinking) |




For meters only the output signal is available.

At analog operation following parameters are available:

- measured value
- setpoint (controllers only)
- valve voltage (controllers only)

## 2.8. Switching between analog and digital operation

Switching between digital operation (in RS232 mode) and analog operation is done by setting parameter IOStatus. See paragraph 0 for more details.

# 3. Digital functionalities

## 3.1. Using different fluid calibrations (optional)

Digital instruments can store to 8 different fluid calibrations. The parameters for these calibrations can be uploaded or modified using Bronkhorst applications software (e.g. Flowplot). Contact factory for detailed information.

## 3.2. Look-up tables

In certain cases it is not possible to describe a sensor signal curve by means of a polynomial function of the 3rd degree. In those cases a look-up table is a good alternative.
A look-up table is a table filled with calibration points. The embedded software inside the digital instrument calculates a continuous smooth function which fits exactly through these calibration points.
Using this method it is possible to describe any monotone rising sensor signal curve with high accuracy.

## 3.3. General form of 2-dimensional look-up tables

The general form of a 2-dimensional look-up table is as follows:

| index | X | Y |
|---|---|---|
| 0 | $x_0$ | $y_0$ |
| 1 | $x_1$ | $y_1$ |
| 2 | $x_2$ | $y_2$ |
| 3 | $x_3$ | $y_3$ |
| … | … | … |
| n | $x_n$ | $y_n$ |

In which 'Y' is the normalized measured value (0-100%), 'X' is the value of the sensor signal and ‘index’ represents the position in the look-up table.

## 3.4. Look-up table for sensor signal

A digital instrument with look-up table capability for sensor signal can store look-up tables with at maximum 15 calibration points.

## 3.5. Using mathematical functions at a digital instrument

Digital instruments are capable of storing 8 different fluid calibrations.
Parameters for these calibrations are stored inside the instrument and can be changed through the RS232 connection by means of a PC-program or a digital Readout and Control module.
Factory calibration parameters are secured and can not be changed unless you have special rights to do this.
Selection of another fluid is part of operation and therefore not secured.
Digital instruments will need at least 1 fluidset of calibration parameters for operation.

## 3.6. Controller

The controlling algorithm for the valve handled by the micro-controller consists of several parameters which can be set via the RS232.
Main parameter settings to be adjusted by a user are:

- Controller response when starting-up from 0% (when valve opens)
- Controller response during normal control (at setpoint step)
- Controller response when controller is stable (within band of 2 %)

All response factors can be set independently by means of a PC-program.

## 3.7. Setpoint/control modes

For switching between different functions in use of a digital meter or controller several modes are available. Output signals (sensor signals) on both analog and digital interface are available at the same time.

| Nr | Mode | Instrument action | Setpoint source | Master source | Slave factor source |
|---|---|---|---|---|---|
| 0 | RS232/digital interface | controlling | RS232/digital interface | | |
| 1 | External input | controlling | Analog input | | |
| 3 | Valve close | close valve | | | |
| 4 | Controller idle | Stand-by on digital interface | | | |
| 5 | Test mode | Testing enabled | | | |
| 7 | Setpoint 100% | Controlling on 100% | 100% | | |
| 8 | Valve Fully open | Fully open valve | | | |
| 9 | Calibration mode | Calibration enabled | | | |
| 12 | Setpoint 0% | Controlling on 0% | 0% | | |
| 18 | RS232 | | RS232 | | |

## 3.8. Auto-zero

The auto-zero procedure is able to remove zero offset signals on the sensor signal automatically. This automatic procedure can be started through the digital interface or by means of the micro push-button switch on top of the instrument.
First be sure there is no flow and the right gas in the instrument, then start auto-zero. During auto-zero procedure the green LED will blink fast. This will take approx. 10 seconds.
When indication is showing 0% signal and the green indication LED is burning continuously again, then auto-zero has been performed well.
This action will be performed already during production at Bronkhorst High-Tech, but may be repeated at wish on site.

## 3.9. Restore parameter settings

All parameter value settings in the instruments are stored in non-volatile memory so each time at power-up these settings are known. However, a user can change several settings afterwards in the field if needed. Sometimes it may be necessary to get back all original settings. Therefore a backup of all settings at production final-test will be stored in non-volatile memory also. Because of this it will be possible to restore these original factory settings at any moment. Of course this will only function as long as there is no memory failure.

Restoring original factory settings can be achieved by means of the micro push-button switch on top of the instrument. See chapter 13 for operation instructions of push-button with switch and LED's.

## 3.10. Changing parameters with Personal Computer Program

For security reasons all important instrument settings can only be changed after sending a security-parameter.
For **operation** of instruments there is always **free access** to the parameters (e.g. setpoint, control mode, setpoint slope, changing fluid).
**Changing of settings** however, is **secured** (e.g. calibration parameters, input + output adjustments, identification).

For changing secured parameter values by means of a PC-software program contact

Bronkhorst High-Tech B.V.

# 4. Parameters and Properties

Digital instruments consist of a microcontroller with several processes running simultaneously for:
Measuring sensor value
Reading analog input signal
Digital signal processing
Driving valve
Setting analog output signal
Communication with world outside
Memory handling

Each process needs its own specific parameters in order to function correctly.
Most parameter values are accessible through the available interface(s) to influence the process behaviour.
Many parameters may be controlled by end-users for more flexible use of the instruments.

Bronkhorst High-Tech offers special software for these purposes (in combination with FlowDDE):
FlowView : Windows application for operation of max. 8 instruments
FlowPlot : Windows application for adjustment of controllers (Value versus time on screen)

End-users are also free to use their own software using either:
FlowDDE : DDE-server for data exchange with Microsoft Windows applications
FLOWB32.DLL : Dynamic Link Library for Microsoft Windows applications
RS232 interface: Protocol description for instructions with ASCII or Binary HEX telegrams

Each parameter has its own properties, like data-type, size, reading/writing allowance, security.
In general:
all parameters used for operation of instruments are free to be used by end-users.
(e.g.: measure, setpoint, control mode, setpoint slope, fluid number, alarm and counter)
all parameter for settings of instruments are meant for BHT-service personnel only.
(e.g.: calibration settings, controller settings, identification, network/fieldbus settings)

Parameters for settings are secured. They can be read-out, but can not be changed without knowledge of special key-parameters and knowledge of the instrument.

**FlowDDE parameter numbers:**
Reading/changing parameter values via FlowDDE offers the user a different interface to the instrument.
Besides the application name: 'FlowDDE' there is only need of:
topic, used for channel number: 'C(X)'
item, used for parameter number: 'P(Y)'
A DDE-parameter number is a unique number in a special FlowDDE instruments/parameter database and not the same as the parameter number from the process on an instrument.
Node address and process number will be translated by FlowDDE to a channel number.

**When not using FlowDDE for communication with FLOW-BUS, each parameter value needs:**
node address of instrument on FLOW-BUS
process number on instrument
parameter number on instrument
All parameter information needed can be found in the parameter table(s) of FlowDDE.MDB.

Following paragraphs will give short descriptions of how to use parameters for <u>operation</u>.
For a more complete description of the parameters the document “917030 FlowPlot” can be used.

The second line after the parameter name gives some additional information about the parameter.

An explanation of the line is:

***Valve output***

**[unsigned long, RW, secured, 0…16777215, DDEpar. = 55, Process/par. = 114/1]**

**unsigned long** **=** one of the datatypes below.

| | |
|---|---|
| Unsigned char | 1 byte |
| Unsigned int | 2 bytes, MSB first |
| Unsigned long | 4 bytes, MSB first |
| Float | 4 bytes IEEE 32-bit single precision numbers, MSB first |
| Unsigned char [] | array of characters (string) |

| | | |
|---|---|---|
| **RW** | = | R - parameter can be read, W – parameter can be written. |
| **Secured** | = | Parameter is secured. If omitted parameter is not secured. |
| **0…16777215** | = | Parameter range. |
| **DDEpar. = 55** | = | FlowDDE parameter number |
| **Process/par. = 114/1** | = | Process number / process parameter number |

Another example is:

***Fluidname***

**[unsigned char[10], RW,secured,a...Z, 0...9, DDEpar. = 25, Process/par. = 1/17]**

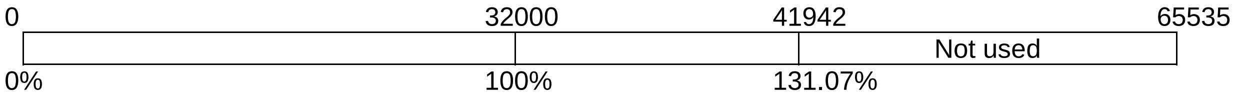
| | | |
|---|---|---|
| **unsigned char[10]** | = | Datatype Unsigned char[], array of characters. [10] = number of characters. |
| **RW** | = | R - parameter can be read, W – parameter can be written. |
| **Secured** | = | Parameter is secured. If omitted parameter is not secured. |
| **a...Z** | = | characters which can be used in the string |
| **0...9** | = | numbers which can be used in the string |
| **DDEpar. = 25** | = | FlowDDE parameter number |
| **Process/par. = 1/17** | = | Process number / process parameter number |

# 5. Normal operation parameters

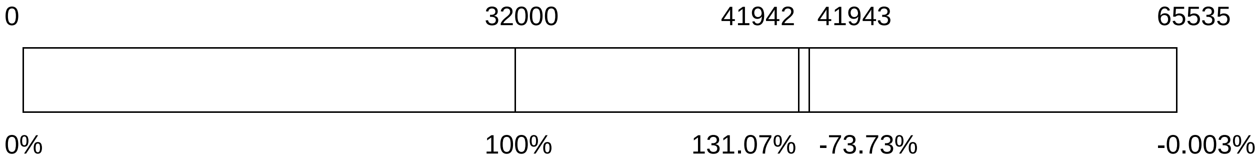
## 5.1. Measured value (measure)

**[unsigned int, R ,0...41942, DDEpar. = 8, Process/par. = 1/0]**
Depending on the type of instrument, measured value indicates the amount of mass flow or pressure metered by the instrument.
Sensor signals at digital instruments will be digitized at the sensor bridge by means of highly accurate AD-converters. Digitized signals will be internally processed by the microcontroller using floating point notation. The sensor signal will be differentiated, linearized and filtered.
At the digital output measured values will be presented as an unsigned integer in the range of 0...65535.

For **Unipolair** mode the signal of 0...100% will be presented in a range of 0...32000.
For the instruments, maximum signal to be expected is 131.07 %, which is: 41942.




For **Bipolair** mode the signal of 0...100% will be presented in a range of 0...32000.
Maximum signal is 131.07 %, which is: 41942, minimum signal is -73.73 %, which is 41943




## 5.2. Setpoint

**[unsigned int, RW,0...32000, DDEpar. = 9, Process/par. = 1/1]**
Setpoint is used to tell the controller of an instrument what the wanted amount of mass flow or pressure is. Signals are in the same range as the measured value, only setpoint is limited between 0 and 100 %.
Setpoint can be given either via optional fieldbus or RS232 or via analog interface.
Parameter control mode selects the active setpoint for the controller. See that paragraph for more detailed information.

## 5.3. Analog input

**[unsigned int, R ,0...65535, DDEpar. = 11, Process/par. = 1/3]**
Depending on the parameter value of analog mode, this signal converts either 0..5Vdc / 0..10Vdc / 0..20mA / 4..20mA. Analog input signals (digitized) are in the same range as measured values (0...32000 = 0...100%).
This input can be used to give setpoint, depending on the value of control mode.

## 5.4. IOStatus

**[unsigned char, RW, secured, 15/79, DDEpar. = 86, Process/par. = 114/11]**
After power-up the parameter value of IOStatus determines whether the instrument operates in analog mode or in digital mode. The control mode is set accordingly.

| IOStatus | Control Mode | Setpoint source |
|---|---|---|
| 15 | 0 | RS232/digital interface |
| 79 | 1 | Analog input |

### 5.4.1. Test and calibration

These are special modes to prepare the instrument for either a test or calibration action.
These modes are used by Bronkhorst High-Tech service personnel only and are not meant for customer use. Putting the instrument in this mode, will disable normal control. The instrument will wait until:

1. Control mode will change again.
2. Instrument receives command (secured parameter) which item should be tested or calibrated.
   When ready performing the wanted action the instrument will return to its previous control mode again.

## 5.5. Fluid number

**[unsigned char, RW,0...7, DDEpar. = 24, Process/par. = 1/16]**
Fluid number is a pointer to the set of calibration parameters. For each fluid (gas or liquid) several parameters get values in order to store the calibration for a specific fluid. This increases accuracy.
Fluid number is an unsigned char parameter (ucFluidnr) in a range of 0...7, where 0 = fluid1 and 7=fluid8.
Up to 8 fluids can be stored in one instrument. Default value = 0 (fluid 1).

## 5.6. Fluid name

**[unsigned char[10], RW,secured,a...Z, 0...9, DDEpar. = 25, Process/par. = 1/17]**
Fluid name consists of the name of the fluid of the actual selected fluidnumber.
Up to 10 characters are available for storage of this name.
Parameter is secured and read-only for normal users. During calibration of the instrument this parameter will get its value. Default value is “Air”.

## 5.7. Valve output

**[unsigned long, RW,secured,0...16777215, DDEpar. = 55, Process/par. = 114/1]**
This parameter is the signal coming out of the controller, going to the DAC for driving the valve.
0...16777215 corresponds with approximately 0...300mAdc.
Maximum output voltage is approximately 14V.

## 5.8. Temperature

**[float, RW, -250..500, DDEpar. = 142, Process/par. = 33/7]**
This parameter shows the temperature in °C of the sensor.

## 5.9. IO switch status (Manifold only)

**[unsigned long, RW, 0..4294967296, DDEpar. = 288, Process/par. = 114/31]**
This 32-bit parameter can be used to set / read up to 16 shut-off valves (16 LSB, Read/Write) and to read the state of up to 16 pressure switches (16 MSB, Readonly).

| PS 15 | PS 14 | ------ | PS 2 | PS 1 | PS 0 | SO 15 | SO 14 | ------ | SO 2 | SO 1 | SO 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| (MSB) | | | | | | | | | | | (LSB) |

Note: this parameter is channel-independent. It contains the data for all pressure switches and shut-off valves connected, not only for the channel at which this parameter is read. E.g. by writing this parameter via the second channel, it is possible to open/close the shut-off valve located at the first channel.

# 6. Direct reading parameters

## 6.1. Sensor type

**[unsigned char, RW,secured,0...4, DDEpar. = 22, Process/par. = 1/14]**
Unsigned char used to select proper set of units for certain sensor, together with Counter unit.
Default settings is 3.

| Value | Description | Controller/Sensor |
|---|---|---|
| 0 | pressure (no counting allowed) | Controller |
| 1 | liquid volume | |
| 2 | liquid/gas mass | |
| 3 | gas volume | |
| 4 | other sensor type (no counting allowed) | |
| 128 | pressure (no counting allowed) | Sensor |
| 129 | liquid volume | |
| 130 | liquid/gas mass | |
| 131 | gas volume | |
| 132 | other sensor type (no counting allowed) | |

## 6.2. Capacity (100%)

**[float, RW,secured,1E-10...1E+10, DDEpar. = 21, Process/par. = 1/13]**
Capacity is the maximum value (span) at 100% for direct reading in sensor base units.
The base unit will be determined by ‘sensor type’.
For each fluid (number) capacity will be stored separately.

## 6.3. Capacity 0%

**[float, RW,secured,-1E-10...1E+10, DDEpar. = 183, Process/par. = 33/22]**
This is the capacity zero point (offset) for direct reading in sensor base units.
The base unit will be determined by ‘sensor type’.
This capacity 0% is the same for all stored fluid (number)s.

## 6.4. Fmeasure

**[float,R, 1E-10...1E+10, DDEpar. = 205, Process/par. = 33/0]**
Floating point version of variable measure (see 4.1).
The user will read-out the measured value in the capacity and capacity unit for which the instrument has been calibrated. These settings depend on variables: capacity, capacity unit, sensor type and capacity 0%.
Fmeasure is a read-only float on (FLOW-BUS) proc 33, par 0.

Value is calculated as follows:

proc33,par0 proc1,par 0 proc1,par13 proc33,par 22 proc33,par22
*fmeasure = ((measure/32000)*(capacity-capacity0%)) + capacity0%*

The value is in units as described in parameter capacity unit (proc 1, par 31).

## 6.5. Fsetpoint

**[float,RW, 1E-10...1E+10, DDEpar. = 206, Process/par. = 33/3]**
With the use of parameter fmeasure, also fsetpoint is often needed. This parameter is R/W as variable in FLOW-BUS proc33, par3. Setpoint can be operated via 2 parameters at the same time. One parameter is setpoint (see 2.1), a short integer. The other one is fsetpoint, a float (in the capacity in which the instrument was calibrated (see also fmeasure). Last received setpoint by the instrument will be valid.
It is not advised to use setpoint and fsetpoint at the same time. Choose either one or the other.

Relation between setpoint and fsetpoint is calculated as follows:

proc1,par1 proc33,par3 proc33,par22 proc1,par13 proc33,par22
*setpoint = ((fsetpoint – capacity0%) / (capacity – capacity0%)) * 32000*

Note: Reading back actual values of fsetpoint is also possible. When a value has been send to proc1, par1 (integer setpoint), then this will be converted to the float setpoint for direct reading in the right capacity and unit.

## 6.6. Capacity unit index

**[unsigned char, RW,secured,0...9, DDEpar. = 23, Process/par. = 1/15]**
Capacity unit is a pointer to select an actual readout unit (see list below).
All capacity units are available for direct reading.

Overview of capacity units in digital instruments and E-7000 for analog instruments:

capacity unit index

| Sensor Type | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | bar | mbar | psi | kPa | cmH2O | cmHg | atm | kgf/cm2 | torr | mmHg | Pa | gf/cm2 |
| 1 | l/min | ml/h | ml/min | l/h | mm3/s | cm3/min | unused | unused | unused | unused | unused | unused |
| 2 | kg/h | kg/min | kg/s | g/h | g/min | g/s | mg/h | mg/min | mg/s | unused | unused | unused |
| 3 | ln/min | mln/h | mln/min | ln/h | m3n/h | mls/min | mls/h | ls/min | ls/h | m3s/h | sccm | slm |
| 4 | usrtype | usrtype | usrtype | unused | Unused | unused | unused | unused | unused | unused | unused | unused |
| 5 | °C | °F | K | unused | Unused | unused | unused | unused | unused | unused | unused | unused |
| 6 | hour | minute | seconds | unused | Unused | unused | unused | unused | unused | unused | unused | unused |
| 7 | kHz | Hz | rpm | unused | Unused | unused | unused | unused | unused | unused | unused | unused |
| 8 | kg | g | mg | ug | Unused | unused` | unused | unused | unused | unused | unused | unused |
| 9 | g/l | kg/l | g/m3 | kg/m3 | Unused | unused | unused | unused | unused | unused | unused | unused |

| **name** | **description** |
|---|---|
| sensor type | Indicator for type of sensor in instrument in relation with a list of units for direct reading |
| capacity unit index | Points to the capacity unit for direct reading in list of available units |

Example:
If you want to readout your instrument in ln/min, then make sure parameter ‘sensor type’ has value 3 and parameter ‘capacity unit index’ has value 0. By means of parameter ‘capacity unit’ the unit string can read-back as a 7 character string.

## 6.7. Capacity unit

**[unsigned char[7], RW,secured, DDEpar. = 129, Process/par. = 1/31]**
Parameter ‘capacity unit’ consisting of 7 characters (string) with selected unit from table.
Can only be written when sensor type = 4: usertype, user unit string of 7 chars can be send.

# 7. Identification parameters

## 7.1. Serial number

**[unsigned char[20], RW,secured, DDEpar. = 92, Process/par. = 113/3]**
This parameter consists of a maximum 20-byte string with instrument serial number for identification.
Example: “M0202123A”

## 7.2. BHTModel number

**[unsigned char[14], RW,secured, DDEpar. = 91, Process/par. = 113/2]**
Bronkhorst High-Tech instrument model number information string.

## 7.3. Firmware version

**[unsigned char[5], R, DDEpar. = 105, Process/par. = 113/5]**
Revision number of firmware. Eg. “V6.01”

## 7.4. Usertag

**[unsigned char[13], RW,secured, DDEpar. = 115, Process/par. = 113/6]**
User definable alias string. Maximum 13 characters allow the user to give the instrument his own tagname.
It is advised here to limit the name up to 7 characters when using E-7000 readout and control modules.
These modules can display the tagname of an instrument only up to 7 characters.
Eg.: “Room1s6”

## 7.5. Customer model

**[unsigned char[16], RW,secured, DDEpar. = 93, Process/par. = 113/4]**
Digital instrument manufacturing configuration information string.
This string can be used by Bronkhorst High-Tech to add extra information to the model number information.

## 7.6. Identification number

**[unsigned char, RW,secured,0...255, DDEpar. = 175, Process/par. = 113/12]**
Bronkhorst High-Tech (digital) device/instrument identification number (pointer).
See list below:

| Value | Description |
|---|---|
| 1 | RS232/FLOW-BUS interface |
| 2 | PC(ISA) interface |
| 3 | ADDA4 (4 channels) |
| 4 | R/C-module, 32 channels |
| 5 | T/A-module |
| 6 | ADDA1: 1 channel ADDA converter module |
| 7 | DMFC: digital mass flow controller |
| 8 | DMFM: digital mass flow meter |
| 9 | DEPC: digital electronic pressure controller |
| 10 | DEPM: digital electronic pressure meter |
| 11 | ACT: single actuator |
| 12 | DLFC: digital liquid flow controller |
| 13 | DLFM: digital liquid flow meter |
| 14 | DSCM-A: digital single channel module for analog instruments |
| 15 | DSCM-D: digital single channel module for digital instr. |
| 16 | FRM: FLOW-BUS rotor meter (calibration-instrument) |
| 17 | FTM: FLOW-BUS turbine meter (calibration-instrument) |
| 18 | FPP: FLOW-BUS piston prover/tube (calibration-instrument) |
| 19 | F/A-module: special version of T/A-module |
| 20 | DSCM-E: evaporator controller module (single channel) |
| 21 | DSCM-C: digital single channel module for calibrators |
| 22 | DDCM-A: digital dual channel module for analog instruments |
| 23 | DMCM-D: digital multi channel module for digital instruments |
| 24 | PRODPS: FLOW-BUS/PROFIBUS DP-slave interface |
| 25 | FCM: FLOW-BUS Coriolis meter |
| 26 | FBI: FLOW-BUS Balance Interface |
| 27 | CORIFC: Cori-Flow Controller |
| 28 | CORIFM: Cori-Flow Meter |

## 7.7. Device type

**[unsigned char[6], R, DDEpar. = 90, Process/par. = 113/1]**
Device type information string: String value in max. 6 chars of descriptions in table above.

# 8. Alarm/status parameters

## 8.1. Alarminfo

**[unsigned char, R,0...255, DDEpar. = 28, Process/par. = 1/20]**
This parameter contains 8 bits with status information about some (alarm) events in the instrument.

| Bit | Meaning |
|---|---|
| 0 | 0 – no error, 1 – Error message in alarm error status register |
| 1 | 0 – no error, 1 – Warning message in alarm warning status register |
| 2 | 0 – no error, 1 – Minimum alarm (sensor signal < minimum limit) |
| 3 | 0 – no error, 1 – Maximum alarm (sensor signal > maximum limit) |
| 4 | 0 – no error, 1 – Batch counter has reached its limit |
| 5 | 0 – no error, 1 – This bit only: Power-up alarm (probably power dip occurred)<br>Together wit bit 2 or bit 3: Response alarm message<br>(setpoint-measure too much difference)<br>(bit 2 or bit 3 indicate if difference is positive or negative) |
| 6 | 0 – no error, 1 – Master/slave alarm: master output signal not received or slave factor out of limits (> 100%) |
| 7 | 0 – no error, 1 – Hardware alarm: check hardware |

## 8.2. Alarm mode

**[unsigned char, RW,secured,0...3, DDEpar. = 118, Process/par. = 97/3]**
Available alarm modes for device:

| Value | Description |
|---|---|
| 0 | Off |
| 1 | alarm on absolute limits |
| 2 | alarm on limits related to setpoint (response alarm) |
| 3 | alarm when instrument powers-up (eg. after power-down) |

## 8.3. Alarm maximum limit

**[unsigned int, RW,secured,0...32000, DDEpar. = 116, Process/par. = 97/1]**
Maximum limit for sensor signal to trigger alarm situation (after delay time).
Note: Minimum limit ≤ Maximum limit ≤ 100 %

## 8.4. Alarm minimum limit

**[unsigned int, RW,secured,0...32000, DDEpar. = 117, Process/par. = 97/2]**
Minimum limit for sensor signal to trigger alarm situation (after delay time).
Note: 0 % ≤ Minimum limit ≤ Maximum limit

## 8.5. Alarm output mode

**[unsigned char, RW,secured,0...2, DDEpar. = 119, Process/par. = 97/4]**
Available alarm output modes for device:

| Value | Description |
|---|---|
| 0 | no relais/TTL-output activity at alarm |
| 1 | relais/TTL-output pulses until reset |
| 2 | relais/TTL-output activated until reset |

## 8.6. Alarm setpoint mode

**[unsigned char, RW,secured,0...1, DDEpar. = 120, Process/par. = 97/5]**
Available alarm setpoint modes for device:

| Value | Description |
|---|---|
| 0 | no setpoint change at alarm |
| 1 | new/safe setpoint at alarm enabled (set at alarm new setpoint) |

## 8.7. Alarm new setpoint

**[unsigned int, RW,secured,0...32000, DDEpar. = 121, Process/par. = 97/6]**
New/safe setpoint during alarm situation until reset.

## 8.8. Alarm delay time

**[unsigned char, RW,secured,0...255, DDEpar. = 182, Process/par. = 97/7]**
Time in seconds alarm action will be delayed when alarm limit has been exceeded.
Also time in seconds automatic reset will be delayed when sensor signal reaches safe level again.

## 8.9. Reset alarm enable

**[unsigned char, RW,secured,0...15, DDEpar. = 156, Process/par. = 97/9]**
Available alarm reset options:

| Value | Description |
|---|---|
| 0 | no reset possible |
| 1 | reset: keyboard/micro-switch |
| 2 | reset: external |
| 3 | reset: keyboard/micro-switch or external |
| 4 | reset: BUS/RS232 |
| 5 | reset: BUS/RS232 or keyboard/micro-switch |
| 6 | reset: BUS/RS232 or external |
| 7 | reset: BUS/RS232 or keyboard/micro-switch or external |
| 8 | reset: automatic |
| 9 | reset: automatic or keyboard/micro-switch |
| 10 | reset: automatic or external |
| 11 | reset: automatic or keyboard/micro-switch or external |
| 12 | reset: automatic or BUS/RS232 |
| 13 | reset: automatic or BUS/RS232 or keyboard/micro-switch |
| 14 | reset: automatic or BUS/RS232 or external |
| 15 | reset: automatic or BUS/RS232 or keyboard/micro-switch or external |

## 8.10. Using an alarm (examples)

Using the alarms will take three steps:
1. Preparing the instrument (setting correct values for mode, limits etc.)
2. Monitoring the alarm info byte (gives info which alarm has occurred)
3. Resetting the alarm (will re-initialize the alarm and set output to normal values again)

All settings needed are secured parameters. These parameters can only be changed if a key-parameter (‘init/reset’) value has been send first to get the instrument in a soft-init mode.
It will stay in this mode until a new power-up situation.

### 8.10.1. Using maximum and minimum alarm

This alarm will check if the measured signal crosses the maximum or minimum limit set by the user.

Example:
Setting maximum alarm on 90%. Setting minimum alarm on 10%.
Relay/TTL output should be on off.
No new setpoint wanted at crossing alarm limit.
Delay on action at output should be 10 seconds.
Reset should be automatically, when signal comes into safe area again.

Through DDE links send following parameter values:

| Parameter | Value |
|---|---|
| Init/reset | 64 |
| alarm maximum limit | 28800 |
| alarm minimum limit | 3200 |
| alarm output mode | 0 |
| alarm setpoint mode | 0 |
| reset alarm enable * | 12 |
| alarm delay time | 10 |
| alarm mode | 1 |

*) Default all reset inputs are enabled, so this command isn’t really necessary

Now the alarm will be active.
Alarm status can be monitored by means of parameter alarm info.

Resetting the alarm will need the following command reset = 0 and then reset=2.

To inactivate the alarm, put it in alarm mode “off”. This will also reset your outputs.
This can be done sending command: alarm mode = 0.

### 8.10.2. Using instrument with response alarm

This alarm will check if the measured value will come within an area limited by maximum limit and minimum limit, related to the setpoint, within a certain delay-time.

Example:
Setting maximum alarm limit on setpoint + 3% .
Setting minimum alarm limit on setpoint – 0.9%.
Relay/TTL output should not be used.
Setpoint wanted at crossing alarm limit = 0 %.
Delay on action at output should be 2 minutes.
Reset via keyboard or BUS/RS232.

Send following parameter values:

| Parameter | Value |
|---|---|
| init/reset | 64 |
| alarm maximum limit | 960 |
| alarm minimum limit | 288 |
| alarm output mode | 0 |
| alarm setpoint mode | 1 |
| alarm new setpoint | 0 |
| reset alarm enable * | 5 |
| alarm delay time | 120 |
| alarm mode | 2 |

*) Default all reset inputs are enabled, so this command isn't really necessary

Now the alarm will be active.
Alarm status can be monitored by means of parameter alarm info.

Resetting the alarm will need following command: reset = 2.

To inactivate the alarm, put it in alarm mode "off". This will also reset your outputs.
This can be done sending command: alarm mode = 0.

# 9. Counter parameters

## 9.1. Counter value

**[float, RW,secured,0...10000000, DDEpar. = 122, Process/par. = 104/1]**
Actual counter value in units selected at counter unit. Value is a float in IEEE 32-bits single precision notation.

## 9.2. Counter mode

**[unsigned char, RW,secured,0...2, DDEpar. = 130, Process/par. = 104/8]**
Available counter modes for device:

| Value | Description |
|---|---|
| 0 | Off |
| 1 | counting upwards continuously |
| 2 | counting up to limit (batchcounter) |

Default value = 0.

## 9.3. Counter setpoint mode

**[unsigned char, RW,secured,0...1, DDEpar. = 126, Process/par. = 104/5]**
Setpoint change enable during counter limit/batch situation (until reset). Default = 0.

| Value | Description |
|---|---|
| 0 | no setpoint change at batch limit allowed |
| 1 | setpoint change at batch limit allowed |

## 9.4. Counter new setpoint

**[unsigned int, RW,secured,0...32000, DDEpar. = 127, Process/par. = 104/6]**
New/safe setpoint at counter limit/batch situation (until reset). See measure for range.
Normally this value is set to 0%.

## 9.5. Counter limit

**[float, RW,secured,0...10000000, DDEpar. = 124, Process/par. = 104/3]**
Counter limit/batch. in units selected at Counter unit. Value is a float in IEEE 32-bits single precision notation. Default setting is 1000000 ln.

## 9.6. Counter output mode

**[unsigned char, RW,secured,0...2, DDEpar. = 125, Process/par. = 104/4]**
Parameter to determine counter relais/TTL-output activity mode when limit/batch has been reached.
Default setting is 0.

| Value | Description |
|---|---|
| 0 | no relais/TTL-output activity at batch limit |
| 1 | relais/TTL-output pulses after reaching batch limit until reset |
| 2 | relais/TTL-output activated after reaching batch limit until reset |

## 9.7. Counter unit

**[unsigned char, RW,secured,0...13, DDEpar. = 123, Process/par. = 104/2]**
Parameter used to select a unit from for certain sensor type (pointer to a unit).
Default setting is 0: 'ln' (for sensor type 3).

| Sensor type | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. |
| 1 | l | mm3 | ml | cm3 | ul | m3 | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. |
| 2 | g | mg | ug | kg | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. |
| 3 | ln | mm3n | mln | cm3n | uln | dm3n | m3n | uls | mm3s | mls | cm3s | ls | dm3s | m3s |
| 4 and > | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. |

## 9.8. Counter unit string

**[unsigned char[4], R, DDEpar. = 128, Process/par. = 104/7]**
Parameter counter unit consisting of 4 characters (string) with selected unit out of table. Read-only parameter.

## 9.9. Using a counter (example)

Using the counter will take three steps:
1. Preparing the instrument (setting correct values for mode, limit etc.)
2. Monitoring the alarm info byte (gives info which alarm has occurred)
3. Resetting the counter (will re-initialize the counter and set output to normal values again)

All settings needed are secured parameters. These parameters can only be changed if a key-parameter ('init/reset') value has been send first to get the instrument in a soft-init mode.
It will stay in this mode until a new power-up situation.

### 9.9.1. Using a batch counter

The measured signal will be integrated in time and there will be a check on a certain limit set by the user.

Example:
You have an instrument with a range of 1 ln/min.
Setting the batch to be reached on 1000 ln.
Relay/TTL output should do nothing.
New setpoint wanted at reaching the limit is 0% (valve should be closed).
Reset should be enabled via BUS/RS232 or by means of keyboard/micro-switch.

Through DDE links send following parameter values:

| Parameter | Value |
|---|---|
| init/reset | 64 |
| counter limit | 1000.0 |
| counter output mode | 0 |
| counter setpoint mode | 1 |
| counter new setpoint | 0 |
| reset counter enable * | 5 |
| counter mode | 2 |

*) Default all reset inputs are enabled, so this command isn't really necessary

Now the counter will be active.
Alarm/counter status can be monitored by means of parameter alarminfo.
Resetting the counter will need following command: reset = 1.
To inactivate the counter, put it in counter mode "off". This will also reset your outputs.
This can be done sending command: counter mode = 0.

# 10. Special parameters

## 10.1. Reset

**[unsigned char, W,0...5, DDEpar. = 114, Process/par. = 115/8]**
Parameter to reset program, counter or alarms. Default value = 0.

| Value | Description |
|---|---|
| 0 | no reset |
| 1 | reset counter value (no mode change) or common reset |
| 2 | reset alarm |
| 3 | restart batch counter |
| 4 | reset counter value (counter off) |
| 5 | Reset module (soft reset) |

Note: To make sure the parameter is accepted send a 0 first.

## 10.2. Init/reset (key-parameter)

**[unsigned char, RW,0...255, DDEpar. = 7, Process/par. = 0/10]**
Init and reset security key command for network/parameter settings.
Make 64 to enable changing of secured parameters. Make 0 again to reset. Default setting is 82.

Note: when an instrument powers-up this value will be reset to 82 always automatically.

## 10.3. Wink

**[unsigned char, W,0...9, DDEpar. = 1, Process/par. = 0/0]**
Unsigned char in range 0...9 enables master to let the instrument connected to that channel wink for several seconds for tracing the physical location. Type of winking depends on instrument. This will be either with red and green LED turn-by-turn or with special characters on a display. Default setting = 0.

# 11. Controller parameters

## 11.1. Controller

The controlling algorithm for the valve handled by the micro-controller consists of several parameters which can be set via the RS232.
Although many parameters could be accessed via RS232, Bronkhorst High-Tech advises not to change these parameters because during manufacturing they have got optimal values for their purposes.
Controller parameters are classified as setting parameters. Changing of controller settings should be performed by or under supervision from trained service personnel only.

Main parameter settings for controller adjustment are listed below:

## 11.2. Open from zero controller response

**[unsigned char, RW,secured,0...255, DDEpar. = 165, Process/par. = 114/18]**
Controller response when starting-up from 0% (when valve opens).
Value 128 is default and means: no correction.
Otherwise controller speed will be adjusted as follows:

New response = old response * $1.05^{(128\text{-RespOpen0})}$

## 11.3. Normal step controller repsonse

**[unsigned char, RW, secured,0...255, DDEpar. = 72, Process/par. = 114/5]**
Controller response during normal control (at setpoint step)

New response = old response * $1.05^{(128\text{-ContrResp})}$

## 11.4. Stable situation controller response

**[unsigned char, RW, secured,0...255, DDEpar. = 141, Process/par. = 114/17]**
Controller response when controller is stable (within band of 2%)

New response = old response * $1.05^{(128\text{-RespStable})}$

## 11.5. PID-Kp

**[float, RW, secured,0...1E+10, DDEpar. = 167, Process/par. = 114/21]**
PID controller response, proportional action, multiplication factor.

## 11.6. PID-Ti

**[float, RW, secured,0...1E+10, DDEpar. = 168, Process/par. = 114/22]**
PID controller response, integration action in seconds.

## 11.7. PID-Td

**[float, RW, secured,0...1E+10, DDEpar. = 169, Process/par. = 114/23]**
PID controller response, differentiation action in seconds.

## 11.8. Sensor differentiator up

**[float, RW, secured,0...1E+10, DDEpar. = 51, Process/par. = 1/12]**
Sensor signal differentiation time in seconds needed to speed-up sensor signals going upwards.

## 11.9. Sensor differentiator down

**[float, RW, secured,0...1E+10, DDEpar. = 50, Process/par. = 1/11]**
Sensor signal differentiation time in seconds needed to speed-up sensor signals going downwards.

## 11.10. Sensor exponential smoothing filter

**[float, RW, secured,0...1, DDEpar. = 74, Process/par. = 117/4]**
This factor is used for filtering the signal coming from the sensor circuitry before it is further processed.

It filters according the following formula:

out = in * Sensor exponential smoothing filter + (1 - Sensor exponential smoothing filter) * out

| Response | Factor setting |
|---|---|
| Slow | 0.05 |
| Normal | 0.1 |
| Fast | 0.2 |
| Very fast | 0.5...1.0 (not advised) |

This filter is in the control loop so it affects the response time.

# 12. Special instrument features

## 12.1. Zero (mass flow instruments only)

The zero procedure is able to remove zero offset signals on the sensor signal automatically.
This automatic procedure can be started through the RS232 or by means of the micro-switch on top of the instrument.

### 12.1.1. Zeroing with the Micro-switch

- **Set process conditions**
  Warm-up, pressure up the system and fill the instrument according to the process conditions.
- **Stop flow**
  Make sure no flow is going through the instrument by closing valves near the instrument.
- **Press and hold**
  With no flow, use the micro-switch (#) to start the zero adjustment procedure.
  Press the micro-switch (#) and hold it, after a short time the red LED will go ON and OFF then the green LED will go ON. At that moment release the micro-switch (#).
- **Zeroing**
  The zeroing procedure will start at that moment and the green LED will blink fast. The zeroing procedure waits for a stable signal and saves the zero. If the signal is not stable zeroing will take long and the nearest point to zero is accepted. The procedure will take approx. 10 sec.
  So make always sure that there is going no flow through the instrument when performing the zeroing procedure.
- **Ready**
  When indication is showing 0% signal and the green indication LED is burning continuously again, then zero has been performed well.

### 12.1.2. Zeroing with digital communication

The following parameters must be used for zeroing an instrument:

| | |
|---|---|
| Initreset | **[unsigned char, RW,0...255, DDEpar. = 7, Process/par. = 0/10]** |
| Control mode | **[unsigned char, RW,0...255, DDEpar. = 12, Process/par. = 1/4]** |
| Calibration mode | **[unsigned char, RW,0...255, DDEpar. = 58, Process/par. = 115/1]** |

- **Set process conditions**
  Warm-up, pressure up the system and fill the instrument according to the process conditions.
- **Stop flow**
  Make sure no flow is going through the instrument by closing valves near the instrument.
- **Send parameters**
  Send the following values to the parameters in this sequence.

  | | |
  |---|---|
  | Initreset | 64 |
  | Control mode | 9 |
  | Calibration mode | 255 |
  | Calibration mode | 0 |
  | Calibration mode | 9 |
- **Zeroing**
  The zeroing procedure will start at that moment and the green LED will blink fast. The zeroing procedure waits for a stable signal and saves the zero. If the signal is not stable zeroing will take long and the nearest point to zero is accepted. The procedure will take approx. 10 sec.
  So make always sure that there is going no flow through the instrument when performing the zeroing procedure.
- **Ready**
  When indication is showing 0% signal and the green indication LED is burning continuously again, then zero has been performed well. Also parameter control mode goes back to its original value.
  As last send 0 to parameter initreset.

This action will be performed already during production at Bronkhorst High-Tech, but may be repeated at wish on site.

## 12.2. Restore parameter settings

All parameter value settings in the instruments are stored in non-volatile memory so each time at power-up these settings are known.
However, several settings can be changed afterwards in the field by a user if needed. Sometimes it may be necessary to get back all original settings.
Therefore a backup of all settings at production final-test will be stored in non-volatile memory also.
Because of this it will be possible to restore these original factory settings at any moment. Of course this will only function as long as there is no memory failure.
Restoring original factory settings can be achieved by means of the micro-switch on top of the instrument or through a command via RS232.
See instructions for manual operation with switch and LED’s for details.

# 13. Manual interface: micro-switch and LED's

## 13.1. General

The micro-switch on top of the digital instrument can be used to start a certain function at the instrument. When the switch is pressed down, both LED's will start indicating different patterns in a loop. The switch has to be pressed down until the 2 LED's are indicating the right pattern. Then the switch has to be released and the choice has been made.

Normally (when the switch is not pressed) the green and red LED are used for mode indication on digital instruments.
(see **table 1 and 2**).

When the switch is pressed-down both LED's will be switched-off for function selection.
As long as the switch will be pressed-down, there will be a change in indication by the 2 LED's after each 4 seconds.
The moment the user recognizes the indication (LED-pattern) for the function he wants, he must release the switch. Now he has triggered the requested function.

In **table 3** is described what the indications are for the function to be performed at normal situation. This is when the switch will be pressed-down during normal operation of the instrument.

In **table 4** is described what the indications are for the functions to be performed at power-up situation of an instrument. This can be realized by pressing the switch first and while pressing, connecting the power. These actions have a more 'initializing' character for the instrument.

Using this 1 switch and 2 LED's this way, offers the user a maximum of possibilities, even if this instrument is only operated by an analog interface.

**Table 1: Green LED indication modes for digital instrument (no switch used)**

| Green LED | Time | Indication |
|---|---|---|
| off | Continuously | Power-off or program not running |
| on | Continuously | Normal running/operation mode |
| short flash | 0.1 sec on, 2 sec off | Special mode: Initialization mode<br>Secured params can be changed |
| normal flash | 0.2 sec on, 0.2 sec off | Special function mode<br>Instrument is busy performing a special function.<br>E.g. auto-zero or self-test |
| long flash | 2 sec on, 0.1 sec off | Not used |
| fast flash | 0.1 sec on, 0.1 sec off | Flow < -1% |
| slow wink | 0.2 sec on, 0.2 sec off | Wink mode<br>By a command send to the instrument, the instrument can "wink" with LED's to indicate its position in a (large) system |
| normal wink | 1 sec on, 1 sec off | Alarm indication: minimum alarm, limit/maximum alarm; power-up alarm or limit exceeded or batch reached. |
| fast wink | 0.1 sec on, 0.1 sec off | Switch-released, selected action started |

***Note: wink = green-red-green-red turn-by-turn***

**Table 2: Red LED indication modes for digital instrument (no switch used)**

| Red LED | Time | Indication |
|---|---|---|
| off | Continuously | No error |
| short flash | 0.1 sec on, 2 sec off | Not used |
| normal flash | 0.2 sec on, 0.2 sec off | Not used |
| long flash | 2 sec on, 0.1 sec off | Not used |
| on | Continuously | Critical error message<br>A serious error occurred in the instrument<br>Instrument needs service before further using |

**Table 3: LED indications using micro-switch at normal running mode of an instrument**

| Green LED | Red LED | Time | Indication |
|---|---|---|---|
| off | off | 0 – 1 sec | Pressing a switch shortly by accident will not cause unwanted reactions of instrument.<br>Pressing the switch 3x briefly with intervals of max. 1 sec. will force instrument to indicate its bus-address/MAC-ID and evt. baudrate.<br>See paragraph 10.2 for more details. |
| off | off | 1 – 4 sec | In case of min/max alarm or counter batch reached:<br>Reset alarm (only if reset by keyboard has been enabled) |
| off | on | 4 – 8 sec | Reset instrument<br>Instrument program will be restarted and all warning and error message will be cleared<br>During (new) start-up instrument will perform a (new) self-test |
| on | off | 8 – 12 sec | Auto-zero<br>Instrument will be re-adjusted for measurement of zero-flow (not for pressure meter/controller)<br>NOTE: First make sure there is no flow and instrument is connected to power for at least 30 minutes ! |
| on | on | 12 – 16 sec | Prepare instrument for FLASH mode<br>At next power-up FLASH mode will be active<br>This mode will be indicated by both LED’s off when instrument is normally powered |

**Note: short flash = 0.1 sec on, 2 sec off**

**Table 4: LED indications using micro-switch at power-up situation of an instrument**

| Green LED | Red LED | Time | Indication |
|---|---|---|---|
| off | off | 0 – 4 sec | No action<br>Pressing a switch shortly by accident will not cause unwanted reactions of the instrument |
| off | normal flash | 4 – 8 sec | Restore parameters<br>All parameter settings (except fieldbus settings) will be restored to situation of final test at BHT production |
| normal flash | off | 8 – 12 sec | Not used |
| normal flash | normal flash | 12 – 16 sec | Not used |

**Note: normal flash = 0.2 sec on, 0.2 sec off**

# 14. Communication: RS232 FLOW-BUS protocol

## 14.1. General

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

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

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

The instrument- RS232 module automatically recognises the protocol used by the PC and adapts its behaviour to the protocol in use. The type of protocol is determined by the first character of a message.

- the first character is >:= (0x3A)    existing type of message.

- the first character is DLE (0x10)    enhanced type of message.

Via the FLOW-BUS DLL (FLOWB32.DLL) the PC determines which protocol is in use.

The communication relation is always master (PC) and slave (instrument). The instrument will always respond on a request from the PC. (Single channel instruments only: if the node address is set to 10 or higher, the instrument will not respond with an error message, when it receives a message with wrong node address).

## 14.2. Initialisation

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

***Manifold instruments:***
The baudrate is selectable with a rotary switch: 38K4 or 115K2 baud. With the other 2 rotary switches the node address can be set. The selected address will be the address for the first channel. Channel 2 and channel 3 will get adress of channel 1 + 1 and address of channel 1 + 2.
When you send a message to node address 128 the instrument will always respond to your message with data from channel 1.

<table>
<tr><th colspan="2">Rotary Switch 1</th><th>Rotary Switch 2</th><th>Rotary Switch 3</th></tr>
<tr><td>0</td><td>RS485<br>9K6</td><td rowspan="5">Address<br>MSD 0-9</td><td rowspan="5">Address<br>LSD 0-9</td></tr>
<tr><td>1</td><td>RS485<br>19K2</td></tr>
<tr><td>2</td><td>RS485<br>38K4</td></tr>
<tr><td>3</td><td>RS232<br>38K4</td></tr>
<tr><td>4</td><td>RS232<br>115K2</td></tr>
</table>

## 14.3. Interface structure

### 14.3.1. Basic datalink format

The basic datalink message format has the following fields:

| | |
|---|---|
| node | message destination |

| | |
|---|---|
| length | data field length |

| | | | |
|---|---|---|---|
| data | data | Data | etc. |

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

### 14.3.2. RS232 ASCII protocol

A basic datalink message is coded in ASCII as follows:

| | len | | node | | | | data | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **:** | len1 | len2 | node1 | node2 | | | data1 | data2 | | | CR |

>**:**= (semicolon)    initial character
len1, len2    length of message including the node address in *bytes*, so (len1,len2) is the basic message length +1.
node1, node2    node address of destination (PC to instrument)
node address of source (instrument to PC)
data1, data2    message field
CR    termination character

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

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

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

| | 0x01 | | error | | |
|---|---|---|---|---|---|
| **:** | 0x30 | 0x31 | error1 | error2 | CR |

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

The error code can have the following values:

| Value | Meaning |
|---|---|
| 1 | no >:" at the start of the message |
| 2 | error in first byte |
| 3 | error in second byte or number of bytes is 0 or message to long |
| 4 | error in received message (receiver overrun, framing error etc.) |
| 5 | FLOW-BUS communication error: timeout or message rejected by receiver |
| 8 | time out during sending |
| 9 | no answer received within time out |

### 14.3.3. Enhanced binary protocol

*Binary coding and control sequences*

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

| First byte | Second byte | Function |
|---|---|---|
| DLE (0x10) | STX (0x02) | Start of message |
| DLE (0x10) | ETX (0x03) | End of message |
| DLE (0x10) | DLE (0x10) | Data byte 0x10 |
| DLE (0x10) | any other character | Not allowed. Messages that contain such a sequence will be ignored. The receiver waits until a new DLE STX sequence. |

The [DLE DLE] sequence is used to prevent possible DLE bytes in the transmitted binary data stream from being recognised as the start of a control sequence. The sender replaces any DLE bytes in the data by two DLE bytes. The datalink of the receiver will convert a [DLE DLE] sequences to one DLE byte.

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

*Enhanced message coding*

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

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| DLE | STX | seq | node | len | | | data | | | DLE | ETX |

DLE, STX start sequence
seq message sequence number
node node address of destination (PC to instrument)
node address of source (instrument to PC)
len length of data field in bytes
data message field
DLE, ETX end sequence

The enhanced protocol allows the transmission of more than one request at a time. The sequence number makes it possible to associate the answer to the according request. The instrument has more than one message buffer where messages may be stored. When the message buffers are full, the instrument responds with an error message.

The responses from the instrument to the PC have the same message format as the request. An error message has a special format:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| DLE | STX | seq | node | 0x00 | error | DLE | ETX |

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

The error code can have the following values:

| Value | Meaning |
|---|---|
| 3 | Message rejected by instrument, receiver buffer overflow |
| 5 | FLOW-BUS communication error: timeout or message rejected by FLOW-BUS node |
| 8 | time out during sending |
| 9 | no response on a request received within time out |

## 14.4. Communication messages

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

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

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

*Communication commands*

| Command | Description |
|---|---|
| 00 | Status message |
| 01 | Send parameter with destination address, will be answered with type 00 command |
| 02 | Send parameter with destination address, no status requested |
| 03 | Send parameter with source address, no status requested |
| 04 | Request parameter, will be answered with type 02 or 00 command |
| 05 | Instruction: send parameter repeatedly (followed by byte with repeating time) |
| 06 | Stop process |
| 07 | Start process |
| 08 | Claim process |
| 09 | Unclaim process |

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

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

*Parameter types*

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

## 14.5. Chaining

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

## 14.6. Status message

*status*

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

***Note: Value from byte 5 of status message may be neglected if value of byte 4 = 0 !**

## 14.7. Send parameters

*Send*

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

## 14.8. Request parameter

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

*Request*

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

*Answer of request*

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

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

## 14.9. Examples

### 14.9.1. Sending setpoint

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

*Send parameters to node 3*

| Nr | Byte | Layout | Description | | |
|---|---|---|---|---|---|
| 0 | ':' | | Start character | | |
| 1 | 06 | | Length 6 | | |
| 2 | 03 | | Node 3 | | |
| 3 | 01 | | Command write with status response | | |
| 4 | 01 | 00000001 | C | 00 | Process not chained |
| | | | P | 01 | Process 1 |
| 5 | 21 | 00100001 | C | 00 | Parameter not chained |
| | | | T | 20 | Parameter type ‘integer’ |
| | | | P | 01 | Parameter number (FBnr.) 1 |
| 6 | 3E | | Setpoint 16000 = 3E80h | | |
| 7 | 80 | | | | |
| 8 | ‘\r’ | | Carriage Return | | |
| 9 | ‘\n’ | | Line Feed | | |

*Answer from node 3*

| Nr | Byte | Description |
|---|---|---|
| 0 | ‘:’ | Start character |
| 1 | 04 | Fixed message length 4. |
| 2 | 01 | Node address 01 |
| 3 | 00 | Command status |
| 4 | 00 | Status ok. |
| 5 | 05 | Status ok, value points to end of send message. |
| 6 | ‘\r’ | Carriage Return |
| 7 | ‘\n’ | Line Feed |

#### 14.9.2. Sending chained parameters

Interface sends following parameters to module at node 3:
Process 0: INIT MODE (10), 64 = soft init
Process 1: FLUID NUMBER (16). 1
Process 1: POLYNOMIAL CONSTANT A (5), 0.0
Process 1: POLYNOMIAL CONSTANT B (6), 1.0
Process 1: POLYNOMIAL CONSTANT C (7), 0.0
Process 1: POLYNOMIAL CONSTANT D (8), 0.0
Process 0: INIT MODE (10), 82 = reset initmode.

*Send parameters to node 3*

| Nr | Byte | Layout | Description | | |
|---|---|---|---|---|---|
| 0 | ':' | | | | |
| 1 | 1D | | Length 29 | | |
| 2 | 03 | | Node 3 | | |
| 3 | 01 | | Command write with status response | | |
| 4 | 80 | 10000000 | C | 80 | Process chained |
| | | | P | 00 | Process 0 |
| 5 | 0A | 00001010 | C | 00 | Parameter not chained |
| | | | T | 00 | Parameter type 'character' |
| | | | N | 0A | Parameter number (FBnr.)10 |
| 6 | 40 | 01000000 | Parameter value 64 set soft init mode | | |
| 7 | 81 | 10000001 | C | 80 | Process chained |
| | | | P | 01 | Process 1 |
| 8 | C5 | 11000101 | C | 80 | Parameter chained |
| | | | T | 40 | Parameter type 'float' |
| | | | N | 05 | Parameter number (FBnr.) 5 |
| 9 | 00 | | Parameter value 'float' 0.0 | | |
| 10 | 00 | | | | |
| 11 | 00 | | | | |
| 12 | 00 | | | | |
| 13 | C6 | 11000110 | C | 80 | Parameter chained |
| | | | T | 40 | Parameter type 'float' |
| | | | N | 06 | Parameter number (FBnr.) 6 |
| 14 | 3F | | Parameter value 'float' 1.0 | | |
| 15 | 80 | | | | |
| 16 | 00 | | | | |
| 17 | 00 | | | | |
| 18 | C7 | 1100111 | C | 80 | Parameter chained |
| | | | T | 40 | Parameter type 'float' |
| | | | N | 07 | Parameter number (FBnr.) 7 |
| 19 | 00 | | Parameter value 'float' 0.0 | | |
| 20 | 00 | | | | |
| 21 | 00 | | | | |
| 22 | 00 | | | | |
| 23 | C8 | 11001000 | C | 80 | Parameter chained |
| | | | T | 40 | Parameter type 'float' |
| | | | N | 08 | Parameter number (FBnr.) 8 |
| 24 | 00 | | Parameter value 'float' 0.0 | | |
| 25 | 00 | | | | |
| 26 | 00 | | | | |
| 27 | 00 | | | | |
| 28 | 00 | 00000000 | C | 00 | Process not chained |
| | | | P | 00 | Process 0 |
| 29 | 0A | 00001010 | C | 00 | Parameter not chained |
| | | | T | 00 | Parameter type 'character' |
| | | | N | 0A | Parameter number (FBnr.) 10 |
| 30 | 52 | 01010010 | Parameter value 82, reset init mode | | |
| 31 | '\r' | | Carriage Return | | |
| 32 | '\n' | | Line Feed | | |

*Answer from node 3*

| Nr | Byte | Description |
|---|---|---|
| 0 | ‘:’ | Start character |
| 1 | 04 | Fixed message length 4. |
| 2 | 03 | Node address |
| 3 | 00 | Command status |
| 4 | 00 | Status ok. |
| 5 | 1C | Status ok, value points to end of send message. |
| 6 | ‘\r’ | Carriage Return |
| 7 | ‘\n’ | Line Feed |

### 14.9.3. Request setpoint

*Request setpoint from node 3 process 1, type integer.*

| Nr | Byte | Layout | Description | | |
|---|---|---|---|---|---|
| 0 | ':' | | | | |
| 1 | 06 | | Length 6 | | |
| 2 | 03 | | Node 3 | | |
| 3 | 04 | | Command read | | |
| 4 | 01 | 00000001 | C | 00 | Process not chained (return) |
| | | | P | 01 | Process 1 (return) |
| 5 | 21 | 00100001 | C | 00 | Parameter not chained (return) |
| | | | T | 20 | Parameter type ‘integer’ (return) |
| | | | N | 01 | Parameter index 1 (return) |
| 6 | 01 | -0000001 | P | 01 | Process 1 |
| 7 | 21 | -0100001 | T | 20 | Parameter type ‘integer’ |
| | | | P | 01 | Parameter number (FBnr.) 1 (setpoint) |
| 8 | ‘\r’ | | | | Carriage Return |
| 9 | ‘\n’ | | | | Line Feed |

*Answer by node 3*

| Nr | Byte | Layout | Description | | |
|---|---|---|---|---|---|
| 0 | ':' | | | | |
| 1 | 06 | | Length 6 | | |
| 2 | 03 | | Node 3 | | |
| 3 | 02 | | Command write | | |
| 4 | 01 | 00000001 | C | 00 | Process not chained |
| | | | P | 01 | Process 1 (receiving process) |
| 5 | 21 | 00100001 | C | 00 | Parameter not chained |
| | | | T | 20 | Parameter type ‘integer’ |
| | | | N | 01 | Parameter index 1 |
| 6 | 3$^{E}$ | | Value 3E80h = 16000 = 50% | | |
| 7 | 80 | | | | |
| 8 | ‘\r’ | | Carriage Return | | |
| 9 | ‘\n’ | | Line Feed | | |

### 14.9.4. Request chained parameters

Interface sends a request for the following parameters to module at node 3:
Process 113: Serial number (3), Usertag (6)

Process 1: Measure (0), Capacity (13), Capacity unit (31), Fluid name (17)
*Request by node 3*

| Nr | Byte | Layout | Description | | |
|---|---|---|---|---|---|
| 0 | ':' | | | | |
| 1 | 1A | | Length 26 | | |
| 2 | 03 | | Node 3 | | |
| 3 | 04 | | Command read | | |
| 4 | F1 | 11110001 | C | 80 | Process chained (return) |
| | | | P | 71 | Process 113 (return) |
| 5 | EC | 11101100 | C | 80 | Parameter chained (return) |
| | | | T | 60 | Parameter type ‘string’ (return) |
| | | | N | 0C | Parameter index 12 (return) |
| 6 | 71 | -1110001 | P | 71 | Process 113 |
| 7 | 63 | -1100011 | T | 60 | Parameter type ‘string’ |
| | | | P | 03 | Parameter number (FBnr.) 3 – Serial number |
| 8 | 14 | 10000100 | | 14 | String length 20 |
| 9 | 6D | 01101101 | C | 00 | Parameter not chained (return) |
| | | | T | 60 | Parameter type ‘string’ (return) |
| | | | N | 0D | Parameter index 13 (return) |
| 10 | 71 | -1110001 | P | 71 | Process 113 |
| 11 | 66 | -1100110 | T | 60 | Parameter type ‘string’ |
| | | | P | 06 | Parameter number (FBnr.) 6 – Usertag |
| 12 | 00 | 00000000 | | | String length 00, length not defined |
| 13 | 01 | 00000001 | C | 00 | Parameter not chained (return) |
| | | | P | 01 | Process 1 (return) |
| 14 | AE | 10101110 | C | 80 | Parameter chained (return) |
| | | | T | 20 | Parameter type ‘integer’ (return) |
| | | | N | 0E | Parameter index 14 (return) |
| 15 | 01 | -0000001 | P | 00 | Process 1 |
| 16 | 20 | -0100000 | T | 20 | Parameter type ‘integer’ |
| | | | P | 00 | Parameter number (FBnr.) 0 – Measure |
| 17 | CF | 11001111 | C | 80 | Process chained (return) |
| | | | T | 40 | Parameter type ‘float’ (return) |
| | | | N | 0F | Parameter index 15 (return) |
| 18 | 01 | -0000001 | P | 01 | Process 1 |
| 19 | 4D | -1001101 | T | 40 | Parameter type ‘float’ |
| | | | P | 0D | Parameter number (FBnr.) 13 – Capacity |
| 20 | F0 | 11110000 | C | 80 | Parameter chained (return) |
| | | | T | 60 | Parameter type ‘string’ (return) |
| | | | N | 10 | Parameter index 16 (return) |
| 21 | 01 | -0000001 | P | 01 | Process 1 |
| 22 | 7F | -1111111 | T | 60 | Parameter type ‘string’ |
| | | | P | 1F | Parameter number (FBnr.) 31 – Capacity unit |
| 23 | 07 | 00001110 | | 07 | String length 7 |
| 24 | 71 | 01110001 | C | 00 | Parameter not chained (return) |
| | | | T | 60 | Parameter type ‘string’ (return) |
| | | | N | 11 | Parameter index 17 (return) |
| 25 | 01 | -0000001 | P | 01 | Process 1 |
| 26 | 71 | 01110001 | T | 60 | Parameter type ‘string’ |
| | | | P | 11 | Parameter number (FBnr.) 17 – Fluid name |

| | | | | | |
|---|---|---|---|---|---|
| 27 | 0A | | | | String length 10 |
| 28 | '\r' | | | | Carriage Return |
| 29 | '\n' | | | | Line Feed |

*Answer by node 3*

| Nr | Byte | Layout | Description |
|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| 0 | ':' | | | | |
| 1 | 41 | | Number of bytes which do follow: 65 bytes | | |
| 2 | 03 | | Node 3 | | |
| 3 | 02 | | Command write | | |
| 4 | F1 | 11110001 | C | 80 | Process chained |
| | | | P | 71 | Process 113 (receiving process) |
| 5 | EC | 11101100 | C | 80 | Parameter chained |
| | | | T | 60 | Parameter type 'string' |
| | | | N | 0C | Parameter index 12 |
| 6 | 14 | | Length of the answer 20 Bytes | | |
| 7-26 | 4D 36 32 31 32 33 34 35 41 20 20 20 20 20 20 20 20 20 20 20<br>Parameter value converted from hex to ASCII : M6212345A | | | | |
| 27 | 6D | 01101101 | C | 00 | Process not chained |
| | | | T | 60 | Parameter type 'string' |
| | | | N | 0D | Parameter index 13 |
| 28 | 00 | | String length 00, length not defined | | |
| 29-36 | 55 53 45 52 54 41 47 00<br>Parameter value converted from hex to ASCII, the values do read: USERTAG | | | | |
| 37 | 01 | 00000001 | C | 00 | Process not chained |
| | | | P | 01 | Process 1 (receiving process) |
| 38 | AE | 10101110 | C | 80 | Parameter chained |
| | | | T | 20 | Parameter type 'integer' |
| | | | N | 0E | Parameter index 14 |
| 39 | 1C | | Parameter value is: 1CD8 (hex)<br>Measure Value is: 7384 (dec) | | |
| 40 | D8 | | | | |
| 41 | CF | 11001111 | C | 80 | Parameter chained |
| | | | T | 40 | Parameter type 'float' |
| | | | N | 0F | Parameter index 15 |
| 42 | 3F | | Parameter Value: 3F 80 00 00 (IEEE-floating point notation, 32-bit single precision )<br>Parameter value converted from float to decimal, the values do read: 1.0 | | |
| 43 | 80 | | | | |
| 44 | 00 | | | | |
| 45 | 00 | | | | |
| 46 | F0 | 11110000 | C | 80 | Parameter chained |
| | | | T | 60 | Parameter type 'string |
| | | | N | 10 | Parameter index 16 |
| 47 | 07 | | Length of the answer 7 Bytes | | |
| 48-54 | 6D 6C 6E 2F 6D 69 6E Parameter value converted from hex to ASCII, the values do read: mln/min | | | | |
| 55 | 71 | 01110001 | C | 00 | Parameter not chained |
| | | | T | 60 | Parameter type 'string' |
| | | | N | 11 | Parameter index 17 |
| 56 | 0A | | Length of the answer 10 Bytes | | |
| 57-66 | 4E 32 20 20 20 20 20 20 20 20<br>Parameter value converted from hex to ASCII, the values do read: N2 | | | | |

### 14.9.5. Request measure

*Request measure from node 3 process 1, type integer.*

| Nr | Byte | Layout | Description |
|---|---|---|---|
| 0 | ':' | | |
| 1 | 06 | | Length 6 |
| 2 | 03 | | Node 3 |

| | | | | | |
|---|---|---|---|---|---|
| 3 | 04 | | | | Command read |
| 4 | 01 | 00000001 | C | 00 | Process not chained (return) |
| | | | P | 01 | Process 1 (return) |
| 5 | 21 | 00100001 | C | 00 | Parameter not chained (return) |
| | | | T | 20 | Parameter type ‘integer’ (return) |
| | | | N | 01 | Parameter index 1 (return) |
| 6 | 01 | -0000001 | P | 01 | Process 1 |
| 7 | 20 | -0100000 | T | 20 | Parameter type ‘integer’ |
| | | | P | 00 | Parameter number (FBnr.) 0 (measure) |
| 8 | ‘\r’ | | | | Carriage Return |
| 9 | ‘\n’ | | | | Line Feed |

*Answer by node 3*

| Nr | Byte | Layout | | | Description |
|---|---|---|---|---|---|
| 0 | ':' | | | | |
| 1 | 06 | | | | Length 6 |
| 2 | 03 | | | | Node 3 |
| 3 | 02 | | | | Command write |
| 4 | 01 | 00000001 | C | 00 | Process not chained |
| | | | P | 01 | Process 1 (receiving process) |
| 5 | 21 | 00100001 | C | 00 | Parameter not chained |
| | | | T | 20 | Parameter type ‘integer’ |
| | | | N | 01 | Parameter index 1 |
| 6 | $3^{E}$ | | | | Value 3E80h = 16000 = 50% |
| 7 | 80 | | | | |
| 8 | ‘\r’ | | | | Carriage Return |
| 9 | ‘\n’ | | | | Line Feed |

### 14.9.6. Request counter value

*Request counter value from node 3 process 104, type float.*

| Nr | Byte | Layout | | | Description |
|---|---|---|---|---|---|
| 0 | ':' | | | | |
| 1 | 06 | | | | Length 6 |
| 2 | 03 | | | | Node 3 |
| 3 | 04 | | | | Command read |
| 4 | 68 | 01101000 | C | 00 | Process not chained (return) |
| | | | P | 68 | Process 104 (return) |
| 5 | 41 | 01000001 | C | 00 | Parameter not chained (return) |
| | | | T | 40 | Parameter type ‘float’ (return) |
| | | | N | 01 | Parameter index 1 (return) |
| 6 | 68 | -1101000 | P | 68 | Process 104 |
| 7 | 41 | -1000001 | T | 40 | Parameter type ‘float’ |
| | | | P | 01 | Parameter number (FBnr.) 1 (counter value) |
| 8 | ‘\r’ | | | | Carriage Return |
| 9 | ‘\n’ | | | | Line Feed |

*Answer by node 3*

| Nr | Byte | Layout | Description | | |
|---|---|---|---|---|---|
| 0 | ':' | | | | |
| 1 | 08 | | Length 8 | | |
| 2 | 03 | | Node 3 | | |
| 3 | 02 | | Command write | | |
| 4 | 68 | 01101000 | C | 00 | Process not chained |
| | | | P | 68 | Process 104 (receiving process) |
| 5 | 41 | 01000001 | C | 00 | Parameter not chained |
| | | | T | 40 | Parameter type ‘float’ |
| | | | N | 01 | Parameter index 1 |
| 6 | 45 | | Parameter value ‘float’ = 5023.96 dec. | | |
| 7 | 9C | | | | |
| 8 | FF | | | | |
| 9 | AE | | | | |
| 10 | ‘\r’ | | Carriage Return | | |
| 1 | ‘\n’ | | Line Feed | | |

# 15. Communication: RS485 Modbus protocol

The Modbus interface offers a direct connection to Modbus. This chapter describes the interface between the instrument and a master device.

More detailed information about Modbus can be found at:

| | |
|---|---|
| Website of the Modbus organisation | www.modbus.org |

or any website of the (local) Modbus organisation of your country (when available).

The implementation of the Modbus interface is based on the following standards:
- MODBUS Application Protocol Specification V1.1b, December 28, 2006, www.modbus.org
- MODBUS over Serial Line specification and implementation guide V1.02, December 20, 2006, www.modbus.org

There is no mutual communication between Modbus slaves; only between master and slave.
Each slave should have it's own unique slave address on the bus, otherwise there is no communication possible

Note: only manifold instruments support the RS485 Modbus protocol.

## 15.1. Address and baudrate settings

The baudrate is selectable with a rotary switch: 9K6, 19K2 or 38K4 baud. With the other 2 rotary switches the node address can be set. The selected address will be the address for the first channel. Channel 2 and channel 3 will get adress of channel 1 + 1 and address of channel 1 + 2.
When you send a message to node address 0 the instrument will always respond to your message with data from channel 1.

| Rotary Switch 1 | | Rotary Switch 2 | Rotary Switch 3 |
|---|---|---|---|
| 0 | RS485 9K6 | Address MSD 0-9 | Address LSD 0-9 |
| 1 | RS485 19K2 | | |
| 2 | RS485 38K4 | | |
| 3 | RS232 38K4 | | |
| 4 | RS232 115K2 | | |

## 15.2. Available parameters

Modbus registers (in the data model) are numbered from 1 to 65536. In a Modbus PDU (Protocol Data Unit) these registers are addressed from 0 to 65535.

The following table lists the most commonly used parameters.

| MODBUS REGISTERS | | | | | | |
|---|---|---|---|---|---|---|
| PARAMETER NAME | PARAMETER TYPE | ACCESS | PDU ADDRESS hex | REGISTER NUMBER | | REMARK |
| | | | | Hex | Dec | |
| Wink | Unsigned char | W | 0x0000 | 0x0001 | 1 | Value 14592 |
| Init/reset | Unsigned char | RW | 0x000A | 0x000B | 11 | |
| Valve output | Unsigned int | RW | 0x001F | 0x0020 | 32 | Range 0..32767 |
| Measure | Unsigned int | R | 0x0020 | 0x0021 | 33 | |
| Setpoint | Unsigned int | RW | 0x0021 | 0x0022 | 34 | |
| Setpoint slope | Unsigned int | RW | 0x0022 | 0x0023 | 35 | |
| Setpoint/control modes | Unsigned char | RW | 0x0024 | 0x0025 | 37 | |
| Sensor type | Unsigned char | RW secured | 0x002E | 0x002F | 47 | |
| Capunit | Unsigned char | RW secured | 0x002F | 0x0030 | 48 | |
| Fluid number | Unsigned char | RW | 0x0030 | 0x0031 | 49 | |
| Alarminfo | Unsigned char | R | 0x0034 | 0x0035 | 53 | |
| Temperature | Unsigned int | R | 0x0427 | 0x0428 | 1064 | See addr 0xA138 |
| Alarm maximum limit | Unsigned int | RW | 0x0C21 | 0x0C22 | 3106 | |
| Alarm minimum limit | Unsigned int | RW | 0x0C22 | 0x0C23 | 3107 | |
| Alarm mode | Unsigned char | RW | 0x0C23 | 0x0C24 | 3108 | |
| Alarm output mode | Unsigned char | RW | 0x0C24 | 0x0C25 | 3109 | |
| Alarm setpoint mode | Unsigned char | RW | 0x0C25 | 0x0C26 | 3110 | |
| Alarm new setpoint | Unsigned int | RW | 0x0C26 | 0x0C27 | 3111 | |
| Alarm delay time | Unsigned char | RW | 0x0C27 | 0x0C28 | 3112 | |
| Reset alarm enable | Unsigned char | RW | 0x0C29 | 0x0C2A | 3114 | |
| Counter value | Unsigned int | RW | 0x0D01 | 0x0D02 | 3330 | See addr 0xE808 |
| Counter unit | Unsigned char | RW | 0x0D02 | 0x0D03 | 3331 | |
| Counter limit | Unsigned int | RW | 0x0D03 | 0x0D04 | 3332 | See addr 0xE818 |
| Counter output mode | Unsigned char | RW | 0x0D04 | 0x0D05 | 3333 | |
| Counter setpoint mode | Unsigned char | RW | 0x0D05 | 0x0D06 | 3334 | |
| Counter new setpoint | Unsigned int | RW | 0x0D06 | 0x0D07 | 3335 | |
| Counter mode | Unsigned char | RW | 0x0D08 | 0x0D09 | 3337 | |
| Identnumber | Unsigned char | RW secured | 0x0E2C | 0x0E2D | 3629 | |
| ContrResp | Unsigned char | RW secured | 0x0E45 | 0x0E46 | 3654 | |
| RespStable | Unsigned char | RW secured | 0x0E51 | 0x0E52 | 3666 | |
| RespOpen0 | Unsigned char | RW secured | 0x0E52 | 0x0E53 | 3667 | |
| Calibration mode | Unsigned char | RW secured | 0x0E61 | 0x0E62 | 3682 | |
| Monitor mode | Unsigned char | RW secured | 0x0E62 | 0x0E63 | 3683 | |
| Reset | Unsigned char | W | 0x0E68 | 0x0E69 | 3689 | |
| Sensor zero potmeter | Unsigned char | RW secured | 0x0E85 | 0x0E86 | 3718 | |
| Modbus slave address | Unsigned char | RW secured | 0x0FAA | 0x0FAB | 4011 | |
| Polycnst A | Float | RW secured | 0x8128..0x8129 | 0x8129..0x812A | 33065..33066 | |
| Polycnst B | Float | RW secured | 0x8130..0x8131 | 0x8131..0x8132 | 33073..33074 | |
| Polycnst C | Float | RW secured | 0x8138..0x8139 | 0x8139..0x81A | 33081..33082 | |
| Polycnst D | Float | RW secured | 0x8140..0x8141 | 0x8141..0x8142 | 33089..33090 | |
| TdsDn | Float | RW secured | 0x8158..0x8159 | 0x8159..0x815A | 33113..33114 | |
| TdsUp | Float | RW secured | 0x8160..0x8161 | 0x8161..0x8162 | 33121..33122 | |
| Capacity | Float | RW secured | 0x8168..0x8169 | 0x8169..0x816A | 33129..33130 | |
| Fluid name | String (10 bytes) | RW secured | 0x8188..0x818C | 0x8189..0x818D | 33161..33165 | |
| Capacity unit string | String (7 bytes) | RW secured | 0x81F8..0x81FB | 0x81F9..0x81FC | 33273..33276 | |
| Fmeasure | Float | R | 0xA100..0xA101 | 0xA101..0xA102 | 41217..41218 | |
| Fsetpoint | Float | RW | 0xA118..0xA119 | 0xA119..0xA11A | 41241..41242 | |
| Temperature | Float | R | 0xA138..0xA139 | 0xA139..0xA13A | 41273..41274 | See addr 0x0427 |
| Capacity 0% | Float | RW secured | 0xA1B0..0xA1B1 | 0xA1B1..0xA1B2 | 41393..41394 | |
| Counter value | Float | RW | 0xE808..0xE809 | 0xE809..0xE80A | 59401..59402 | See addr 0x0D01 |
| Counter limit | Float | RW | 0xE818..0xE819 | 0xE819..0xE81A | 59417..59418 | See addr 0x0D03 |
| Counter unit string | String (4 bytes) | R | 0xE838..0xE839 | 0xE839..0xE83A | 59449..59450 | |
| Device type | String (6 bytes) | R | 0xF108..0xF10A | 0xF109..0xF10B | 61705..61707 | |
| Modelnumber | String (14 bytes) | RW secured | 0xF110..0xF116 | 0xF111..0xF117 | 61713..61719 | |
| Serialnumber | String (16 bytes) | RW secured | 0xF118..0xF11F | 0xF119..0xF120 | 61721..61728 | |

| PARAMETER NAME | PARAMETER TYPE | ACCESS | PDU ADDRESS hex | REGISTER NUMBER | | REMARK |
|---|---|---|---|---|---|---|
| | | | | Hex | Dec | |
| Manufacturer config | String (16 bytes) | RW secured | 0xF120..0xF127 | 0xF121..0xF128 | 61729..61736 | |
| Firmware version | String (5 bytes) | R | 0xF128..0xF12A | 0xF129..0xF12B | 61737..61739 | |
| Usertag | String (13 bytes) | RW | 0xF130..0xF136 | 0xF131..0xF137 | 61745..61751 | |
| PID Kp | Float | RW secured | 0xF2A8..0xF2A9 | 0xF2A9..0xF2AA | 62121..62122 | |
| PID Ti | Float | RW secured | 0xF2B0..0xF2B1 | 0xF2B1..0xF2B2 | 62129..62130 | |
| PID Td | Float | RW secured | 0xF2B8..0xF2B9 | 0xF2B9..0xF2BA | 62137..62138 | |
| IO switch status | Long integer | RW | 0xF2F8..0xF2F9 | 0xF2F9..0xF2FA | 62201..62202 | |
| Dynamic display factor | Float | RW secured | 0xF508..0xF509 | 0xF509..0xF50A | 62729..62730 | |
| Static display factor | Float | RW secured | 0xF510..0xF511 | 0xF511..0xF512 | 62737..62738 | |
| Exponential smoothing | Float | RW secured | 0xF520..0xF521 | 0xF521..0xF522 | 62753..62754 | |
| Modbus baudrate | Long integer | RW secured | 0xFD48..0xFD49 | 0xFD49..0xFD4A | 64841..64842 | |

Notes:

- Access indicates whether parameter can be Read and/or Written.
- When a byte parameter is read, the upper 8-bits of the Modbus register will be 0. When a byte parameter is written, the upper 8-bits must be set to 0.
- Long integer parameters have a length of 4 bytes and are mapped on two consecutive Modbus registers. The first register contains bit 31-16, the second register contains bit 15-0.
- Floating point parameters have a length of 4 bytes and are mapped on two consecutive Modbus registers. Floats are in single precision IEEE format (1 sign bit, 8 bits exponent and 23 bits fraction). The first register contains bit 31-16, the second register contains bit 15-0.
- String parameters can have a length of maximal 16 bytes and can take up to 8 Modbus registers where each register contains two characters (bytes). The upper byte of the first register contains the first character of the string. When writing strings, the write action should always start from the first register as a complete block (it is not possible to write a part of a string). If the string is shorter than the specified maximum length the string should be terminated with an 0.
- Parameters Temperature, Counter value and Counter limit can be found in the parameter table as an unsigned integer variant and as a floating point variant. Only the floating point variant supports the full parameter range and resolution.

# 16. Testing and diagnostics

All digital instruments have facilities to run self-test procedures for diagnostics.
Most of the instrument functions will be tested automatically during start-up or normal running mode of the instrument.
The red LED on top of the instrument is used to indicate if there is something wrong.
The longer the LED is burning (blinking) red, the more is wrong with the instrument.
See chapter 13 for more details about the LED’s and other documents for more specific troubleshooting.

# 17. Troubleshooting

## 17.1. General

For a correct analysis of the proper operation of a flow controller it is recommended to remove the unit from the process line and check it without applying fluid supply pressure. In case the unit is dirty, this can be ascertained immediately by loosening the compression type couplings.
Furthermore remove the cover and check if all connectors are fixed properly. By disabling and enabling the power supply the instrument indicates whether there is an electronic failure. After that, fluid pressure is to be applied in order to check behavior. If there should be suspicion of leakage in case of a gas unit, do not check for bubbles with a leak detection liquid under the cover as this may lead to a short-circuit in the sensor or p.c.board.

## 17.2. Return shipment

When returning material, always describe the problem and if possible the work to be done, in a covering letter.

**It is absolutely required to notify the factory if toxic or dangerous fluids have been metered with the instrument!**

This is to enable the factory to take sufficient precautionary measures to safe-guard the staff in their repair department. Take care of proper packing, if possible use the original packing box; seal instrument in plastic etc.

**Contaminated instruments must be dispatched with a completely filled in 'declaration on contamination form'.**
**Contaminated instruments without this declaration will not be accepted.**

**Note:**
If the instruments have been used with toxic or dangerous fluids the customer should pre-clean the instrument.

**Important:**
Clearly note, on top of the package, the customer clearance number of Bronkhorst High-Tech B.V., namely:

NL801989978B01

If applicable, otherwise contact your distributor for local arrangements.

## 17.3. Service

If the equipment is not properly serviced, serious personal injury and/or damage to the equipment could be the result. It is therefore important that servicing is performed by trained and qualified service personnel. Bronkhorst High-Tech B.V. has a trained staff of servicemen available.

## 17.4. Troubleshooting summary

| Symptom | Possible cause | Action |
|---|---|---|
| No output signal | Bad communication | 1a) check communication |
| | | 1b) check cabling |
| | No power supply | 1c) check power supply |
| | | 1d) check cable connection |
| | Supply pressure too high | 1d) return to factory |
| | Output stage blown-up due to long lasting shortage and/or high-voltage peaks | 1e) return to factory |
| | Valve blocked/contaminated | 1f) return to factory |
| | Sensor/capillary failure | 1g) return to factory |
| No communication possible with PC-software program | Interface settings could be wrong | 2) Check settings e.g. for RS232 (COM-port and baudrate) |
| Red or green LED's are blinking on top of the instrument | Instrument is in a special operation mode or is indicating some kind of error/warning | 3) See chapter 12 for more details |
| No/bad reaction to manual instructions by means of micro push-button switch on top | Some actions can only be used under special conditions | 4) See chapter 12 for more details |
| Maximum output signal | Output stage blown-up | 5a) return to factory |
| | Sensor failure | 5b) return to factory |
| Output signal much lower than setpoint signal or desired flow | Sensor blocked/contaminated | 6a) return to factory |
| | Valve blocked/contaminated | 6b) return to factory |
| | Incorrect type of gas is used and/or pressure/diff. pressure | 6c) try instrument on conditions for which it was designed |
| Oscillation | Supply pressure/diff. pressure too high | 7a) look at the instrument for the correct pressure |
| | | 7b) lower pressure |
| | Pressure regulator is oscillating | 7c) lower pressure or return factory |
| | Controller adjustment wrong | 7d) adjust controller |
| Small flow at zero setpoint | Valve leaks due to damaged plunger or dirt in orifice | 8a) return to factory |
| | Pressure too high or much too low | 8b) apply correct pressure |
| No or slow response to setpoint changes | Slope value could be set too high (ramp function for setpoint signal) | 9) Check and/or change slope value with PC software program (on request) |
| | Control mode/setpoint source for instrument could point to a different setpoint source | 10) Check if correct control mode/setpoint source is selected for the instrument |
| Inaccuracy of measurement | | 11) Check if powering of instrument is done correctly |
| Controller doesn't function as wanted | Settings for controller are for other behaviour than you want | 12) Adjust parameters |
| | | 13) Change response factor settings for controller (see manual of readout/ control module or PC-software) |
| Other troubles | One or more parameter settings could be wrong | 14) Get back original factory settings by means of micro push-button on top (see chapter 13) |
| | | 15) Contact sales representative for assistance if the above doesn't help |