8030730/AE00/V3-0/2016-10

Products Solutions Services

Technical Information
BCU Basic Control Unit

GMS800 Series
Advanced Functions, Configuration, Modbus Functions

Endress+Hauser
People for Process Automation

Described Product

Product name:	BCU
Basic device:	GMS800 series gas analyzers

Manufacturer

Endress+Hauser SICK GmbH+Co. KG
Bergener Ring 27
01458 Ottendorf-Okrilla
Germany

Legal information

Original document

This document is an original document of Endress+Hauser SICK GmbH+Co. KG.

Contents

1 About this document

1.1 Warning symbols

⚠ **DANGER**
Hazard (general)

1.2 Warning levels / signal words

CAUTION

Hazard or unsafe practice which could result in less severe or minor injuries.

Notice

Hazard which could result in property damage.

Note

Hints

1.3 Information symbols

❗ **NOTICE**
Important technical information for this product

ℹ **NOTE**
Hint, link to information at another place

2 Important information

2.1 Responsibility of user

2.1.1 Intended users

This Technical Information is aimed at competent persons who are familiar with GMS800 series gas analyzers and, based on their device-specific training and knowledge as well as knowledge of the relevant regulations, can assess the tasks given and recognize the dangers involved.

NOTE
This Manual is only valid in combination with the “GMS800 Operating Instructions”.

NOTE
Observe all safety information in the “GMS800 Operating Instructions”.

2.1.2 Special local requirements

Follow all local laws, regulations and company-internal operating directives applicable at the respective installation location of the equipment.

2.2 Additional documentation/information

This document supplements the Operating Instructions for GMS800 series gas analyzers. It extends the “ GMS800 Series Operating Instructions” with technical information on the BCU Basic Control Unit running with SOPAS ET.

- ▶ Observe the Operating Instructions provided for the “GMS800 Series Gas Analyzers” and “BCU Basic Control Unit”.

NOTE
The “GMS800 Series Gas Analyzers” Operating Instructions also specify all further documents belonging to the individual device.

NOTICE
- ▶ Pay primary attention to any individual information provided.

3 SOPAS ET connection

3.1 System requirements

- Network: Ethernet (LAN)
- To connect via switch or hub: Standard LAN cable (1:1, RJ45 plug-in connector)
- For direct connection to a PC: Crossover LAN cable or standard LAN cable with crossover adapter

NOTE
There is a risk of unauthorized changes ("hacks") when the BCU is run within an Ethernet network.

▶ Only connect the BCU to networks properly protected (e.g. with a firewall).

3.2 IP address

Each device in a network must have an individual address. Two options are available to specify the IP address for the BCU:

Option	Procedure	Instructions
Manual setting:	▶ Set a suitable IP address with a menu function on the BCU display.	→ Operating Instructions "BCU Basic Control Unit"
Automatic setting:	▶ Set the IP address using SOPAS ET.	see "Creating a network connection", page 9

NOTE
The BCU supports the "Auto-IP" function to set the IP address via the network.

3.3 Creating a network connection

3.3.1 Connect a PC directly (Peer-to-Peer)

1. Connect the Ethernet interface of the GMS800 (see the Supplementary Operating Instructions of the enclosure used) to the LAN interface of the PC.
   ▷ Use either a crossover cable or a standard LAN cable + crossover adapter.

NOTE
If the PC is fitted with an Ethernet controller that can switch to "crossover" electronically, it may be possible to connect using a standard LAN cable.

2. Start SOPAS ET on the PC.
3. In SOPAS ET:
   a) Select **Device search** when the device search does not start automatically.
   b) Locate and connect the GMS800.

NOTE
- Use the **Search settings** when no device is found even though these are connected.
- Further information on searching for devices and on the search settings → Help function in SOPAS ET

3.3.2 Connect a PC via a network (Ethernet/LAN)

1. Connect the Ethernet interface of the GMS800 (see the Supplementary Operating Instructions of the enclosure used) to the hub/switch/router/gateway of the network (standard LAN cable).
2. Connect the PC to the network.
3. Start SOPAS ET on the PC.
4. In SOPAS ET:
   a) Select **Device search** when the device search does not start automatically.
   b) Locate and connect the GMS800.

NOTE

- Use the **Search settings** when no device is found even though these are connected.
- Further information on searching for devices and on the search settings → Help function in SOPAS ET

4 Function overview

4.1 Menu tree in SOPAS ET

User level:
O = This menu can be accessed in user level “Operator”.
A = This menu can be accessed in user level “Authorized Customer”.
Access rights:
○ = User can view the function.
● = User can set/start the function.

BCU/... menu level	O	A	Explanation
Measuring screen			
Measuring screen 1 ... 8	○	○	see "Selecting a measuring screen", page 13
Diagnosis			
Logbook	○	○	see "Viewing the logbook", page 15
Status	○	○	see "Checking the status", page 16
System overview	-	○	see "Displaying the system overview", page 16
I/O module	○	○	see "Checking the I/O modules", page 17
Adjustment results			
Adjustment	○	○	see "Adjustment", page 17
Validation	○	○	see "Validation", page 18
Parameter			
Measuring screen			
Measuring screen 1 ... 8	-	●	see "Measuring screens in SOPAS ET", page 30
I/O			
Digital inputs (DIi)	○	●	see "Configuring digital inputs (DIi)", page 37
Digital outputs (DOi)	○	●	see "Configuring digital outputs (DOi)", page 38
Analog inputs (AIi)	○	●	see "Configuring analog inputs (AIi)", page 39
Analog outputs (AOi)	○	●	see "Configuring analog outputs (AOi)", page 41
Modbus inputs (MBIi)	○	●	see "Identifying Modbus inputs", page 66
Modbus outputs (MBOi)	○	●	see "Configuring Modbus outputs", page 62
Formulas			
Formulas	○	●	see "Formulas", page 80
Function buttons	○	○	see "Programming the Function buttons", page 91
Timers			
Cyclic trigger (CTi)	○	●	see "Setting a cyclic trigger", page 98
Countdown (SCCDi)	○	●	see "Setting countdown timers", page 99
Variables			
Real values (RVi)	○	●	see "Floating point variables (RVi)", page 85
Integer values (IVi)	○	●	see "Integer variables (IVi)", page 86
Boolean values (BVSi) - input signals (system)	○	●	see "Boolean variables (BVSi) - input signals (system)", page 86
Boolean values (BVIi) – input signals	○	●	see "Boolean variables (BVIi) – input signals", page 88
Boolean values (BVOi) – output signals	○	○	see "Boolean variables (BVOi) – output signals", page 90
Boolean values (BVi)	○	●	see "Boolean variables (BVi)", page 91

BCU/… menu level				O	A	Explanation
		Help values (SjHVk)		○	●	see "Configuring help values", page 55
	Sequence control programs					
		Number		○	●	see "Determining the number of sequence control programs", page 94
		Sequence control programs				
			Sequence control 1 … 4 (SC1 … SC4)[1]	○	●	see "Programming sequence control programs", page 95
	Measured values (MVi)			○	●	see "Configuring measured values", page 52
	Measuring point automatic			○	●	see "Measuring point automatic", page 53
	Test Gas Table			○	●	see "Configuring test gases (Test gas Table)", page 43
	Adjustment / validation			○	●	see "Automatic adjustments/validations", page 48
	Pump control			○	●	see "Controlling the pump", page 34
	Modbus			○	○	see "Checking Modbus activation", page 29
	Device			○	○/●	see "Configuring device (BCU) parameters", page 29
	System			○		see "Check system (measuring system)/enter user identifier", page 34
Maintenance						
	Tests					
		Digital inputs		-	○	see "Testing digital inputs (DIi)", page 19
		Digital outputs		-	●	see "Testing digital outputs (DOi)", page 20
		Analog inputs		-	○	see "Testing analog inputs (AIi)", page 22
		Analog outputs		-	●	see "Testing analog outputs (AOi)", page 23
	System functions			○	●	see "System functions (BCU)", page 24
	Maintenance mode			○	○/●	see "Signaling maintenance mode", page 25
	Function buttons			○	○	see "Using the Function buttons", page 25
	Operator commands			-	●	see "Starting adjustment/validation (operator commands)", page 26
	Manual adjust			●	●	see "Performing manual adjust", page 45
	Data backup					
		User settings		-	●	see "Backing up/restoring data", page 27
		Factory settings		-	●	see "Backing up/restoring data", page 27

[1] When a corresponding number is set

5 Operating functions

5.1 Measuring screen

5.1.1 Selecting a measuring screen

Function

This selection only applies to the measuring screens in SOPAS ET. Measuring screens on the BCU display are not affected.

NOTE

- Configuring measuring screen contents, see "Configuring measuring screen contents", page 31
- Configuring single measuring screens, see "Configuring single measuring screens", page 33
- Setting display contents → “BCU Basic Control Unit” Operating Instructions

Procedure

1. Call up menu: **BCU/Measuring screen**.
2. Select desired measuring screen.

NOTE

- If menu branch **Measuring screen** is not displayed:
  Configure at least one measuring screen (see "Configuring measuring screen contents", page 31).
- Measuring screens are refreshed at one second intervals.

5.1.2 Measuring screen colors

The measuring screen color identifies the current measured value status.

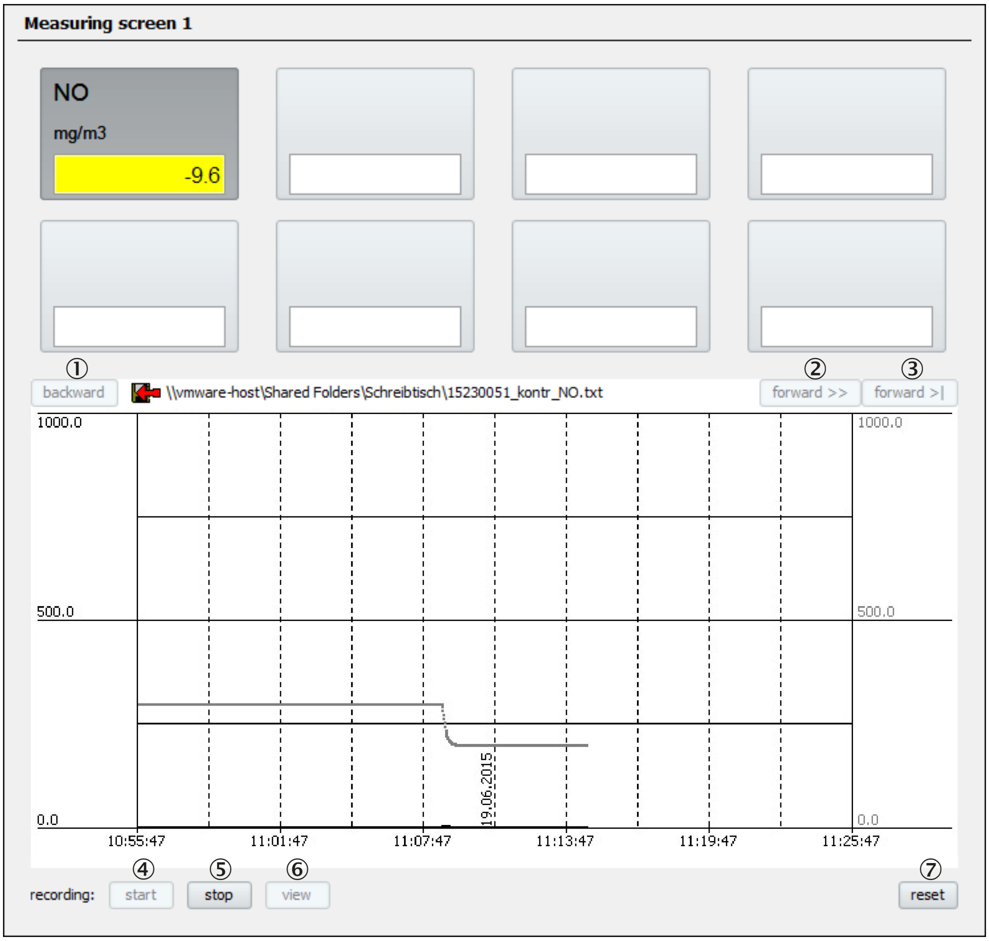
Color	Status flag[1]
None / green	–
Yellow	M C U[2]
Red	F[3]

Table 1: Colors for measured value status

[1] Explanation, see "Status flags", page 73
[2] At least one of these status flags must be activated.
[3] Has priority over other status flags.

5.1.3 Functions in the measuring screen “Chart recorder”

NOTE
The measuring screen “Chart recorder” serves to record current measured values and then display these values later.



Measuring screen 1

NO
mg/m3
-9.6

\\vmware-host\Shared Folders\Schreibtisch\15230051_kontr_NO.txt

Figure 1: Measuring screen “Chart recorder” (example)

① Shift the displayed time interval backwards.
② Shift the displayed time interval forwards.
③ Shift the displayed time interval forwards to the current time.
④ Start recording current measured values:

▶ Select the storage location for the recorded data (select path, enter file name).

⑤ Stop recording current measured values.
⑥ View recorded measured values:

▶ Select file containing the desired data (path, file name)

⑥[1] Stop display of recorded measured values.
⑦ Start current display again.

[1] “Live view” appears instead of “view” when recorded measured values are displayed.

5.2 Diagnosis

NOTE
General information about the BCU, see "Configuring device (BCU) parameters", page 29

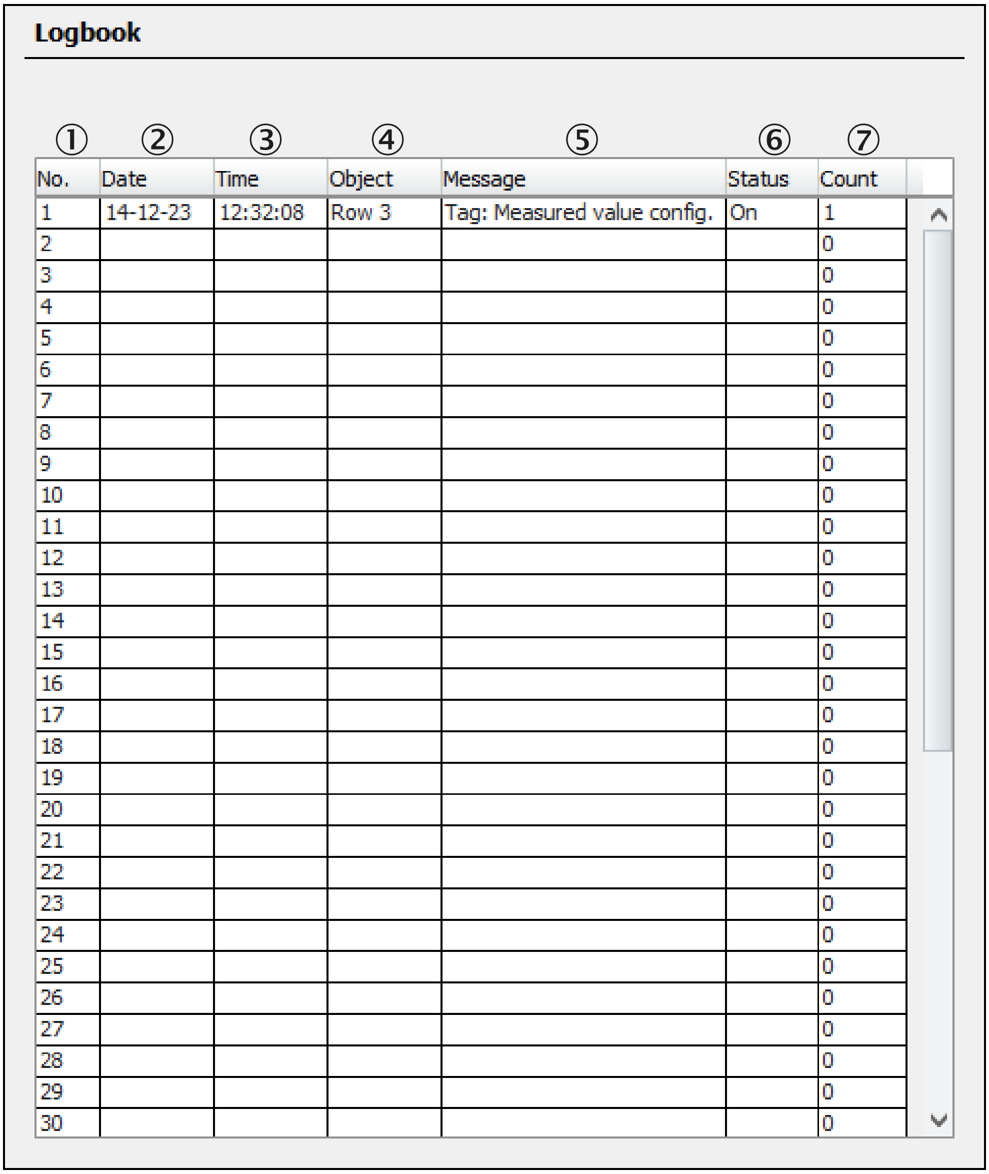
5.2.1 Viewing the logbook

Function

The Logbook Table contains the last 50 internal function and error messages. The context help explains error classes and error numbers.

Procedure

1. Call up menu: **BCU/Diagnosis/Logbook**.



No.	Date	Time	Object	Message	Status	Count
1	14-12-23	12:32:08	Row 3	Tag: Measured value config.	On	1
2						0
3						0
4						0
5						0
6						0
7						0
8						0
9						0
10						0
11						0
12						0
13						0
14						0
15						0
16						0
17						0
18						0
19						0
20						0
21						0
22						0
23						0
24						0
25						0
26						0
27						0
28						0
29						0
30						0

Figure 2: Menu "Logbook" (example)

① Sequential number in Logbook
② Time of last message change
③ Time of last message change
④ Message source
⑤ Message description and classification
⑥ Current message status (on/off)
⑦ Total count of activations

▶ To delete existing logbook entries: Select **Delete all entries**.

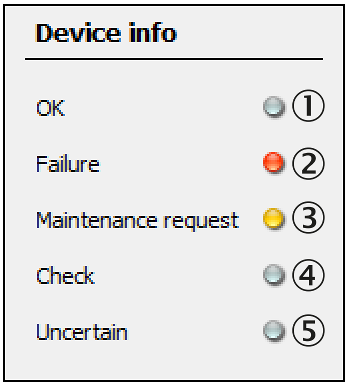
5.2.2 Checking the status

Function

Five LED symbols indicate the current operating state of the complete GMS800.

Procedure

▶ Call up **BCU/Diagnosis/Device info**.



Device info

OK ● ①
Failure ● ②
Maintenance request ● ③
Check ● ④
Uncertain ● ⑤

Figure 3: Menu "Status"

Status	Significance	Consequence
①	GMS800 in correct operating state.	Normal measuring operation running.
②	Malfunction: At least one output signal is permanently invalid.	GMS800 defective. ▶ Arrange for repair.
③	Maintenance request: Output signal is still valid but the wear limit will soon be reached.	▶ Arrange for maintenance.
④	At least one output signal is temporarily invalid. GMS800 is in a specifically induced operating state.	GMS800 measured values do not represent the real values. ▶ Inform locations affected as required.
⑤	Self-monitoring: Current operating conditions or internal malfunctions probably make measurement uncertainty higher than expected.	Normal measuring operation not possible at this time. ▶ Check ambient temperature. ▶ Check peripheral devices. ▶ Check measuring conditions.

5.2.3 Displaying the system overview

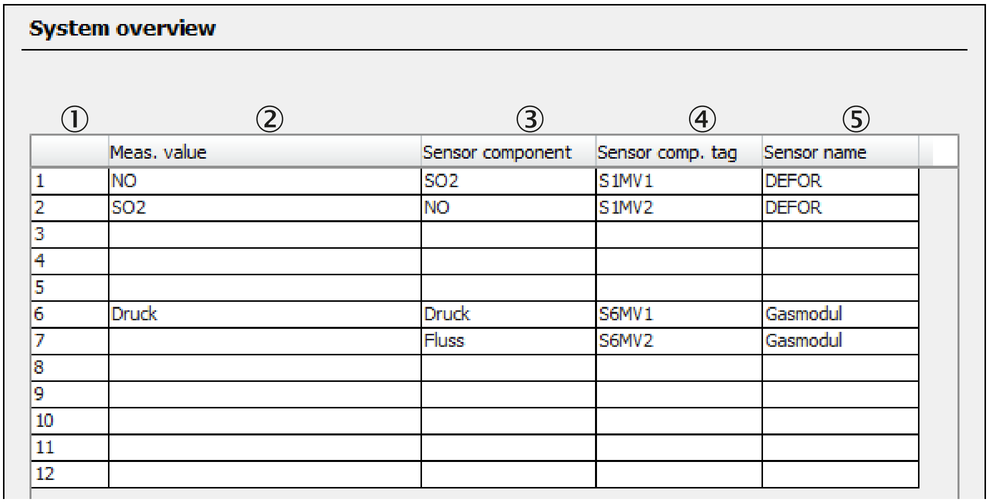
Function

Menu function **System overview** displays:

- Internal components
- Associated tags (identification character)
- Module delivering the component values.

Procedure

▶ Call up menu: **BCU/Diagnosis/System overview**.



System overview

	Meas. value	Sensor component	Sensor comp. tag	Sensor name
1	NO	SO2	S1MV1	DEFOR
2	SO2	NO	S1MV2	DEFOR
3				
4				
5				
6	Druck	Druck	S6MV1	Gasmodul
7		Fluss	S6MV2	Gasmodul
8				
9				
10				
11				
12				

Figure 4: Menu “System overview” (example)

①	Consecutive number
②	Measuring component name in the BCU
③	Component name in the Sensor module
④	Sensor component tag in the BCU
⑤	Identifier of the Sensor module from which the measured value originates

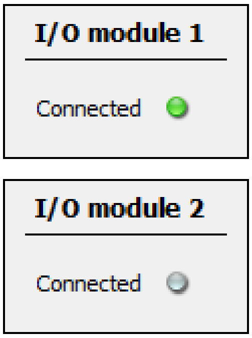
5.2.4 Checking the I/O modules

Function

Menu function **I/O-Module** displays which I/O modules are connected. I/O module 2 is an option.

Procedure

▶ Call up menu: **BCU/Diagnosis/I/O Module.**



I/O module 1

Connected ●

I/O module 2

Connected ○

Figure 5: Menu “I/O overview” (example)

LED symbol	Status of I/O module
Green	Connected and usable
Gray	Not present / not connected properly / not ready for operation

5.2.5 Adjustment results

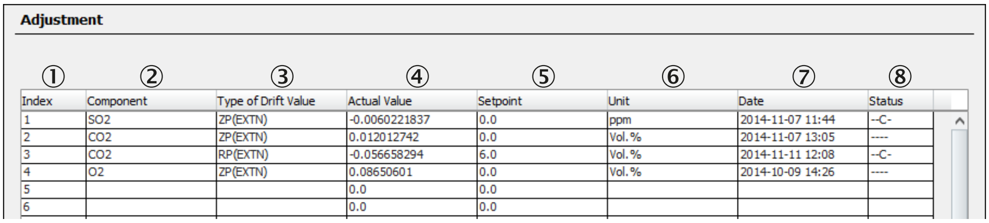
Function

Menu function “Adjustment results” displays the adjustment or validation results.

5.2.5.1 Adjustment

Procedure

1. Call up menu: **BCU/Diagnosis/Adjustment results/Adjustment.**



Adjustment

Index	Component	Type of Drift Value	Actual Value	Setpoint	Unit	Date	Status
1	SO2	ZP(EXTN)	-0.0060221837	0.0	ppm	2014-11-07 11:44	--C-
2	CO2	ZP(EXTN)	0.012012742	0.0	Vol.%	2014-11-07 13:05	----
3	CO2	RP(EXTN)	-0.056658294	6.0	Vol.%	2014-11-11 12:08	--C-
4	O2	ZP(EXTN)	0.08650601	0.0	Vol.%	2014-10-09 14:26	----
5			0.0	0.0			
6			0.0	0.0			

Figure 6: Menu “Adjustment” – List (example)

① Consecutive number
② Measuring component of the Sensor module
③ Zero point or reference point and type of test medium

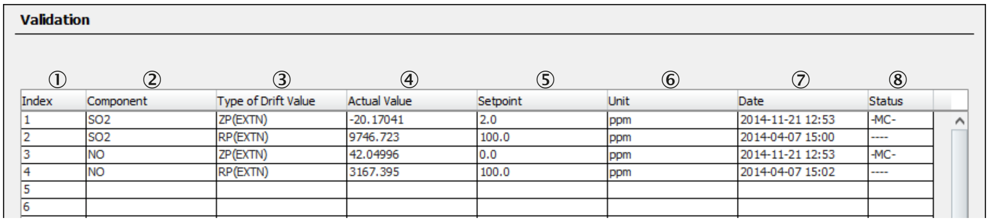
- ZP: Zero point
- RP: Reference point
- EXTN: External test medium was used for the adjustment (test gas).
- INTN: Internal test medium (adjustment cuvette)

④ Actual measured value measured for the component before the drift correction
⑤ Set point value specified in the Test gas Table (test gas or adjustment cuvette concentration)
⑥ Unit for component
⑦ Date and time when adjustment was completed
⑧ NAMUR status of the measuring component during completion (see "Checking the status", page 16)

5.2.5.2 Validation

Procedure

1. Call up menu: **BCU/Diagnosis/Adjustment results/Validation.**



Validation

Index	Component	Type of Drift Value	Actual Value	Setpoint	Unit	Date	Status
1	SO2	ZP(EXTN)	-20.17041	2.0	ppm	2014-11-21 12:53	-MC-
2	SO2	RP(EXTN)	9746.723	100.0	ppm	2014-04-07 15:00	----
3	NO	ZP(EXTN)	42.04996	0.0	ppm	2014-11-21 12:53	-MC-
4	NO	RP(EXTN)	3167.395	100.0	ppm	2014-04-07 15:02	----
5							
6							

Figure 7: Menu “Validation“ – List (example)

① Consecutive number
② Measuring component of the Sensor module
③ Zero point or reference point and type of test medium

- ZP: Zero point
- RP: Reference point
- EXTN: External test medium was used for the validation (test gas).
- INTN: Internal test medium (adjustment cuvette)

④ Actual measured value measured for the component before the drift correction
⑤ Set point value specified in the rows Test gas Table (test gas or adjustment cuvette concentration)
⑥ Unit for component
⑦ Date and time when validation was completed
⑧ NAMUR status of the measuring component during completion (see "Checking the status", page 16)

6 Maintenance functions

NOTE
Manual adjust (adjustment function), see "Performing manual adjust", page 45

6.1 I/O tests

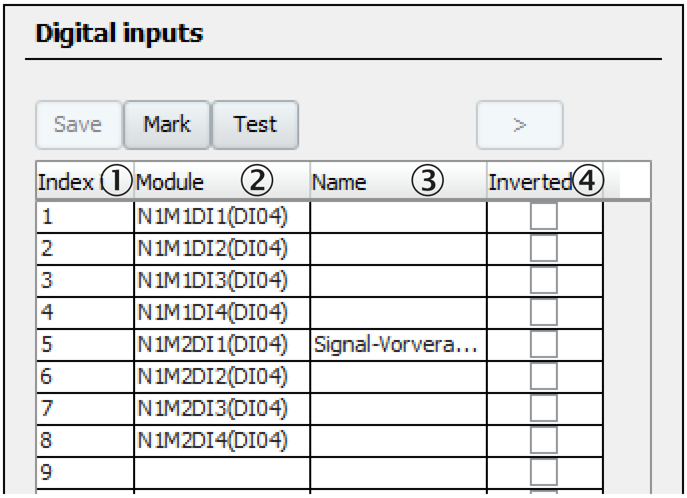
6.1.1 Testing digital inputs (DIi)

Function

These menus serve to view the current state of each digital input of the GMS800. Internal assignment and programmed switching logic are displayed.

Procedure

1. Call up menu: **BCU/Maintenance/Tests/Digital inputs.**



Index	Module	Name	Inverted
1	N1M1DI1(DI04)		□
2	N1M1DI2(DI04)		□
3	N1M1DI3(DI04)		□
4	N1M1DI4(DI04)		□
5	N1M2DI1(DI04)	Signal-Vorvera...	□
6	N1M2DI2(DI04)		□
7	N1M2DI3(DI04)		□
8	N1M2DI4(DI04)		□
9			□

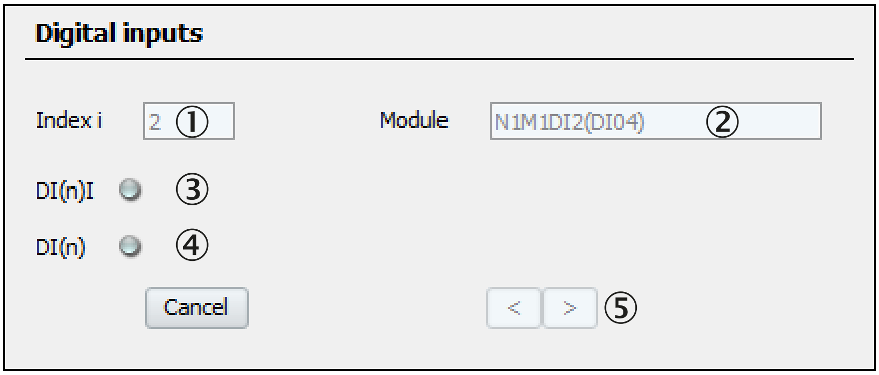
Figure 8: Menu “Digital inputs” – list (example)

① Consecutive number
② Internal identifier of digital input[1]
③ Internal name of connection[2]
④ ☑= logical input status is inverse to electronic status[2]

[1] Only digital inputs with a module identifier are present in the GMS800 (see "Number of I/O connections", page 36).

[2] Parameter, see "Configuring digital inputs (DIi)", page 37.

2. Mark one or several Table rows.
3. Select **Test.**



Digital inputs

Index i: 2 ①

Module: N1M1DI2(DI04) ②

Figure 9: Menu “Digital inputs” – Menu (example)

① Index number of digital input

② Internal identifier of digital input

③ Current electronic state of digital input:

- Green LED symbol: Input electronically activated.
- Gray LED symbol: Input not electronically activated.

④ Current logical state of digital input:

- Green LED symbol: Input internally regarded as activated.
- Gray LED symbol: Input not internally regarded as activated.

⑤ Calls up the menu for a different digital input.[1]

[1] Only when several Table rows were selected.

6.1.2 Testing digital outputs (DOi)

Function

These menus serve to control and test each single digital output of the BCU. This allows testing the function of these outputs and the interaction with connected devices. Internal assignment and programmed switching logic are displayed.

The test function is only used on one single digital output at a time. All the other digital outputs remain in operation during this time.

CAUTION

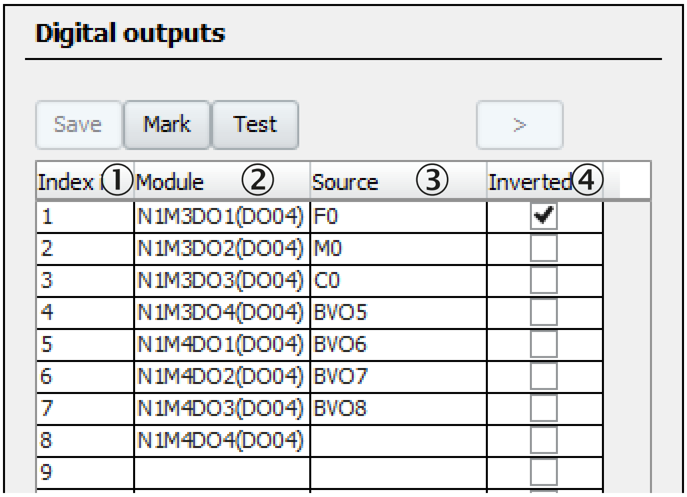
Risk for connected systems

The operating function of the selected digital output is deactivated as long as the Test menu is called up for a digital output. When the menu for processing the single digital outputs is open (see figure 11, page 21), the electronic state of the digital output corresponds to the selected test value.

▶ Ensure the digital output test does not cause any problems on connected locations.

Procedure

1. Call up menu: **BCU/Maintenance/Tests/Digital outputs**.



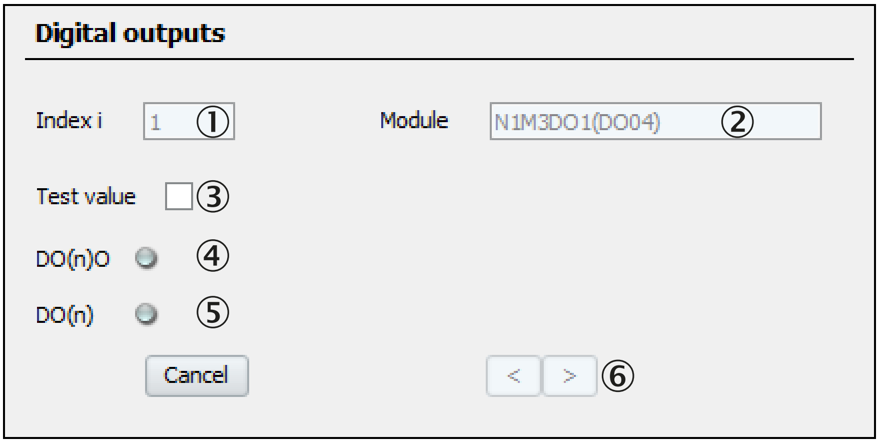
Index	Module	Source	Inverted
1	N1M3DO1(DO04)	F0	☑
2	N1M3DO2(DO04)	M0	☐
3	N1M3DO3(DO04)	C0	☐
4	N1M3DO4(DO04)	BVO5	☐
5	N1M4DO1(DO04)	BVO6	☐
6	N1M4DO2(DO04)	BVO7	☐
7	N1M4DO3(DO04)	BVO8	☐
8	N1M4DO4(DO04)		☐
9			☐

Figure 10: Menu "Digital outputs" – List (example)

① Consecutive number
② Internal identifier of digital output[1]
③ Tag of assigned signal source[2]
④ ☑ = electronic output status is inverse to logical source status[2]

[1] Only digital inputs with a module identifier are present in the GMS800 (see "Number of I/O connections", page 36).
[2] Parameters, see "Configuring digital outputs (DOi)", page 38.

2. Mark one or several Table rows.
3. Select **Test.**
4. To invert the current state of the digital output: Activate **Test value.**



Index i: 1
Module: N1M3DO1(DO04)
Test value ☐
DO(n)O ●
DO(n) ●

Figure 11: Menu "Digital outputs" – Menu (example)

① Index number of digital output
② Internal identifier of digital output
③ ☑ = output activated logically
④ Current electronic state of digital output:

- Green LED symbol: Output is electronically activated (relay energized).
- Gray LED symbol: Output not electronically activated.

⑤ Current logical state of digital output:

- Green LED symbol: Output internally regarded as activated.
- Gray LED symbol: Output not internally regarded as activated.

⑥ Calls up the menu for a different digital output.[1]

[1] Only when several Table rows were selected.

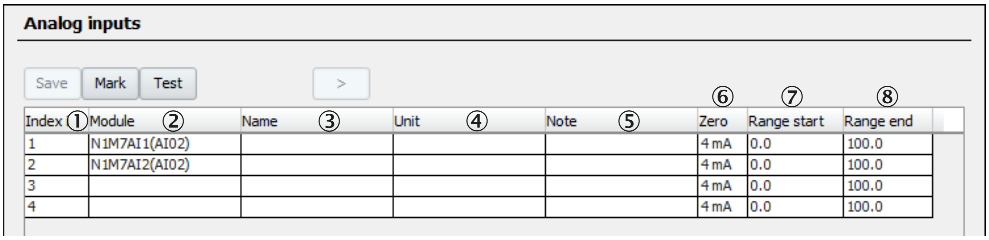
6.1.3 Testing analog inputs (AIi)

Function

These menus serve to view the current input signal of each analog input of the GMS800. The assigned physical measurement span and the current physical input value according to this conversion are also displayed.

Procedure

1. Call up menu: **BCU/Maintenance/Tests/Analog inputs.**



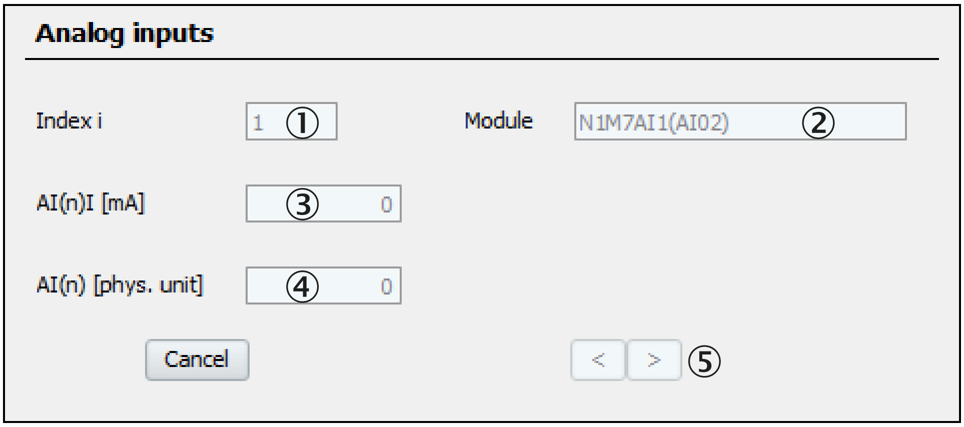
Index ①	Module ②	Name ③	Unit ④	Note ⑤	Zero	Range start	Range end
1	N1M7AI1(AI02)				4 mA	0.0	100.0
2	N1M7AI2(AI02)				4 mA	0.0	100.0
3					4 mA	0.0	100.0
4					4 mA	0.0	100.0

Figure 12: Menu “Analog inputs” – List (example)

① Consecutive number
② Internal identifier of analog input[1]
③ Name of input variable[2]
④ Physical unit[2]
⑤ Comment[2]
⑥ Electronic zero point of the input signal (0/2/4 mA)[2]
⑦ Physical value corresponding to electronic zero point
⑧ Physical value corresponding to input signal “20 mA”

[1] Only analog inputs with a module identifier are present in the GMS800 (see "Number of I/O connections", page 36).
[2] Setting, see "Configuring analog inputs (AIi)", page 39.

2. Mark one or several Table rows.
3. Select **Test.**



Index i: 1 ①
Module: N1M7AI1(AI02) ②
AI(n)I [mA]: ③ 0
AI(n) [phys. unit]: ④ 0

Figure 13: Menu “Analog inputs” – Menu (example)

① Index number of digital input
② Internal identifier of digital input
③ Current electronic input signal
④ Input signal in physical unit after programmed conversion[1]

⑤ Calls up the menu for a different analog input.[2]

[1] Under consideration of the electronic zero point.
[2] Only when several Table rows were selected.

6.1.4 Testing analog outputs (AOi)

Function

These menus serve to test the analog outputs of the GMS800 by setting the desired output value. The electronic zero point and the usage parameters are displayed.

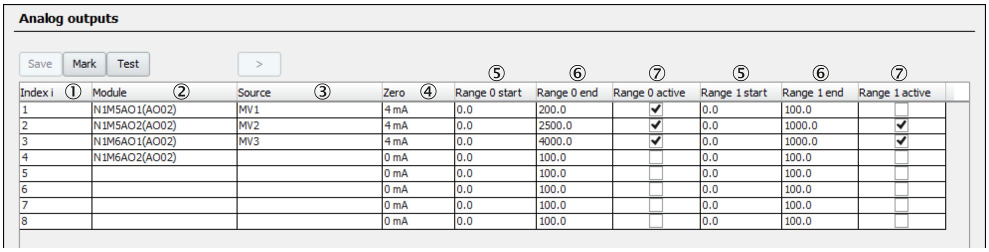
CAUTION
Risk for connected systems

The operating function of the selected analog output is deactivated as long as the Test menu is called up for a digital output. When the menu for processing the single analog outputs is open (see figure 15, page 24), the electronic state of the analog output corresponds to the selected test value.

▶ Ensure the analog output test can not cause any problems on connected locations.

Procedure

1. Call up menu: **BCU/Maintenance/Tests/Analog outputs.**



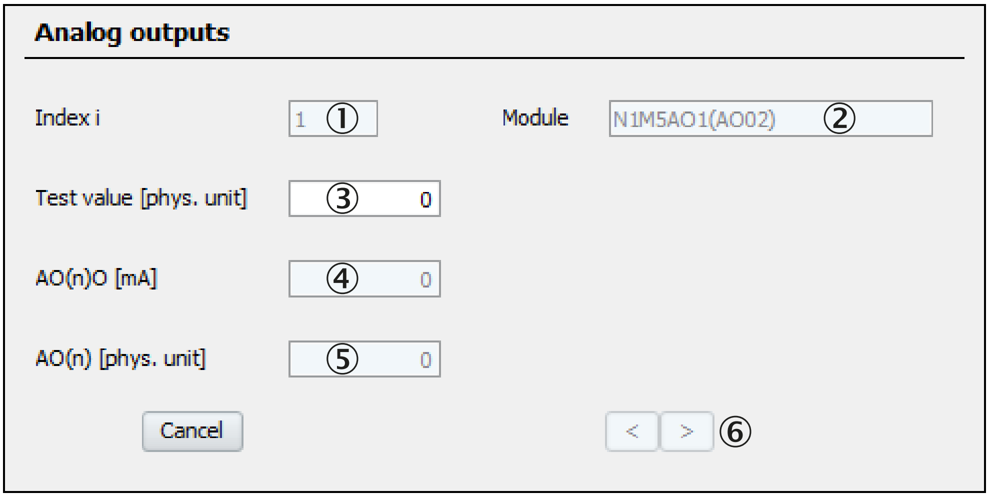
Index i ①	Module ②	Source ③	Zero ④	⑤ Range 0 start	⑥ Range 0 end	⑦ Range 0 active	⑤ Range 1 start	⑥ Range 1 end	⑦ Range 1 active
1	N1M5AO1(AO02)	MV1	4 mA	0.0	200.0	☑	0.0	100.0	□
2	N1M5AO2(AO02)	MV2	4 mA	0.0	2500.0	☑	0.0	1000.0	☑
3	N1M6AO1(AO02)	MV3	4 mA	0.0	4000.0	☑	0.0	1000.0	☑
4	N1M6AO2(AO02)		0 mA	0.0	100.0	□	0.0	100.0	□
5			0 mA	0.0	100.0	□	0.0	100.0	□
6			0 mA	0.0	100.0	□	0.0	100.0	□
7			0 mA	0.0	100.0	□	0.0	100.0	□
8			0 mA	0.0	100.0	□	0.0	100.0	□

Figure 14: Menu “Analog outputs” – List (example)

① Consecutive number
② Internal identifier of analog output[1]
③ Tag of assigned signal source[2]
④ Electronic zero point of output signal (0/2/4 mA)[2]
⑤ Physical value corresponding to electronic zero point
⑥ Physical value corresponding to output signal “20 mA”
⑦ ☑ = output range used during operation[2]

[1] Only analog outputs with a module identifier are present in the GMS800 (see "Number of I/O connections", page 36).
[2] Parameters, see "Configuring analog outputs (AOi)", page 41.

2. Mark one or several Table rows.
3. Select **Test.**
4. Enter desired output value in field “Test value [phys. unit]”.



Analog outputs

Index i: 1 ①

Module: N1M5AO1(AO02) ②

Test value [phys. unit]: ③ 0

AO(n)O [mA]: ④ 0

AO(n) [phys. unit]: ⑤ 0

Figure 15: Menu “Analog outputs” – Menu (example)

①	Index number of analog output
②	Internal identifier of analog output
③	Input field for desired output value of analog output
④	Current electronic output signal of analog output
⑤	Current output signal of analog output in physical unit
⑥	Calls up the menu for a different analog output.[1]

[1] Only when several Table rows were selected.

6.2 System functions (BCU)

Warm start

Function	Usage
Restart the BCU.	When unclear malfunctions occur.

Parameter upload

Function	Usage
Load all parameters and process values of the connected modules in the BCU.	When module parameters have been changed using SOPAS ET or when a module has been added.[1]

[1] Not necessary when the BCU has been started again afterwards.

Update mode

Service function. Updates may only be carried out by a trained Service technician.

When the “Active” display is yellow, a firmware update can be carried out via the serial interface. Service access rights are required for this function.

Procedure

1. Call up menu: **BCU/Maintenance/System functions.**
2. Activate the desired function.

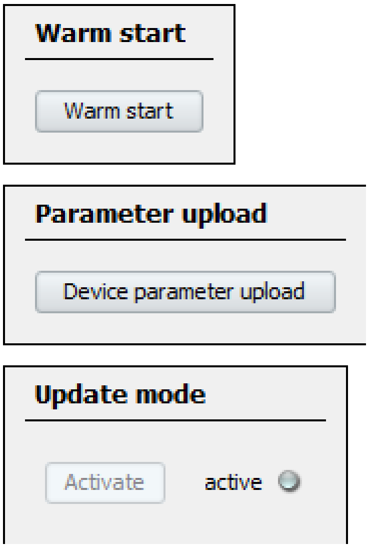


Figure 16: Menus in “System functions”

6.3 Signaling maintenance mode

Function

Maintenance mode can be activated and deactivated by setting the checkmark.

When “Maintenance mode” is activated, the “MAINTENANCE REQUEST“ LED is on and status flag “CO” of the BCU is activated (see "Applicable values and states (read tags)", page 74). If this status flag triggers a digital output, an external location can be signaled that the GMS800 is not in normal measuring operation.

Procedure

1. Call up menu: **BCU/Maintenance/Maintenance mode.**

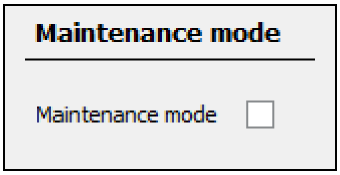


Figure 17: Menu “Maintenance mode”

2. Activate/deactivate **Maintenance mode.**

6.4 Using the Function buttons

Function

Service buttons are only available when programmed by trained personnel.

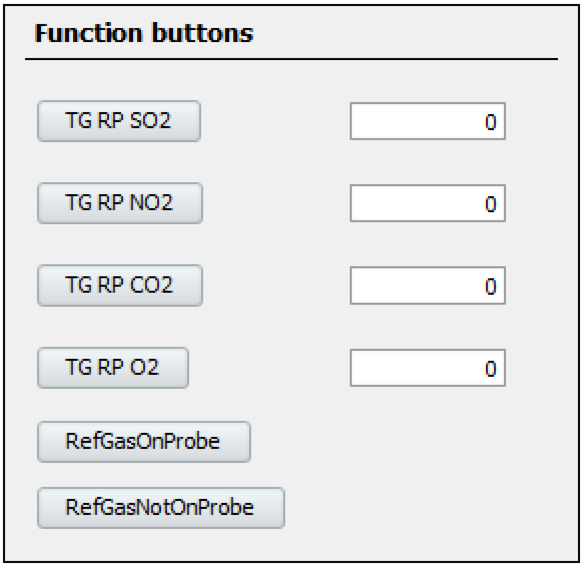
Many different data assignments and function calls are possible and should only be configured with appropriate technical knowledge. Additional documentation is necessary when Function buttons are configured on the device.

A maximum of 8 Function buttons are available. Each Function button executes a function that can be individually programmed (see "Defining variables", page 85).

Only those Function buttons with a name and function are displayed.

Procedure

1. Call up menu: **BCU/Maintenance/Function buttons.**



Function buttons

TG RP SO2 | 0

TG RP NO2 | 0

TG RP CO2 | 0

TG RP O2 | 0

RefGasOnProbe

RefGasNotOnProbe

Figure 18: Menu “Function buttons” (example)

2. Select desired Function button.

✓ The programmed function of the Function button is performed.

Example

In the example shown (see figure 18, page 26), the span gases for the corresponding components (1 - 4) can be assigned concentration values and span gas feed assigned or activated via the sampling probe (activation of the corresponding valve) (5 - 6).

When numeric values are entered, the value shown next is always zero because the current value can not be read out for technical reasons.

6.5 Starting adjustment/validation (operator commands)

Function

Menu **Operator commands** serves to start programmed adjustment or validation procedures manually (see "Automatic adjustments/validations", page 48). Procedures marked as “active” are available. The buttons show the respective function. The button names can be set as required and are taken directly from the “Adjustment / Validation” Table.

A running adjustment or validation procedure can be aborted with a mouse click.

Procedure

1. Call up menu: **BCU/Maintenance/Operator commands.**
2. To run the desired function: Click the appropriate button.

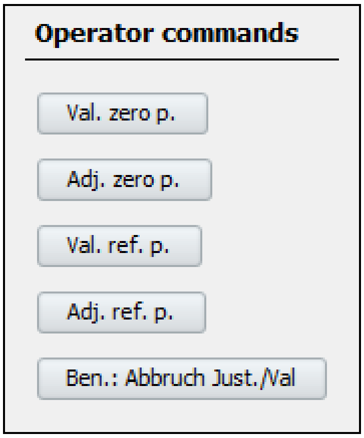


Figure 19: Menu “Operator commands” (example)

NOTE
The procedure phases are displayed on the BCU display.

Function rules

- Only one procedure can run at any one time.
- Clicking the button starts or aborts the procedure immediately.
- A running procedure cannot be started again (start command is ignored).
- A start command for a different procedure does not abort the running procedure.

6.6 Backing up/restoring data

Functions

Function	Effect
Backup	• A copy of the current settings is stored as “last backup”. • The previous “last backup” (If present) is renamed as “next to last backup”.
Restore last user settings[1]	• Current settings are replaced by the last saved copy. • A warm start is then done automatically.
Restore next to last user settings[1]	

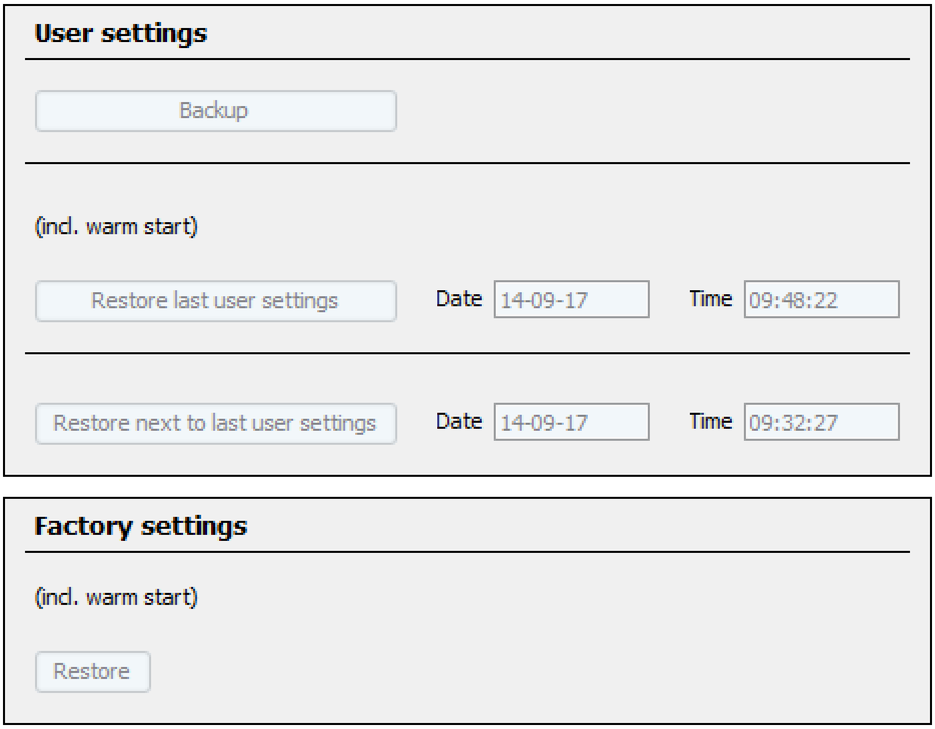
[1] Only available when a backup has been created (see entries for date and time).

Restore factory settings	• Current settings are replaced by the original factory settings. • A warm start is then done automatically.

Procedure

1. Call up menu: **BCU/Maintenance/User settings**.
2. Select desired function.

✓ The function is performed.



User settings

(incl. warm start)

Restore last user settings

Date 14-09-17 Time 09:48:22

Restore next to last user settings

Date 14-09-17 Time 09:32:27

Factory settings

(incl. warm start)

Figure 20: Menu "User settings"

7 Basic settings

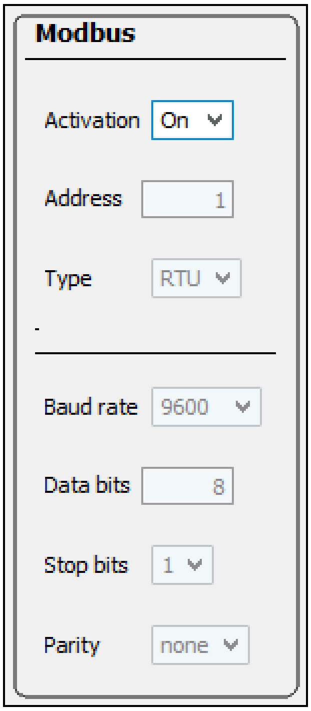
7.1 Checking Modbus activation

Function

This function shows which parameters are set for the Modbus functions of the GMS800. The Modbus functions are deactivated in state **Off**.

Procedure

▶ Call up menu: **BCU/Parameter/Modbus.**



Modbus

Activation: On

Address: 1

Type: RTU

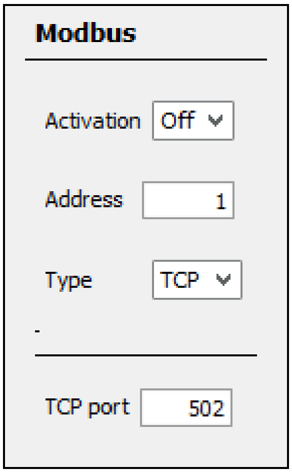
Baud rate: 9600

Data bits: 8

Stop bits: 1

Parity: none

Figure 21: Menu “Modbus” (example 1)



Modbus

Activation: Off

Address: 1

Type: TCP

TCP port: 502

Figure 22: Menu “Modbus” (example 2)

NOTE
Modbus parameters can be changed in user level **Service.**

7.2 Configuring device (BCU) parameters

Function

Menu **Device** serves to set the individual designation **Location** which can serve to identify the GMS800 in SOPAS ET. Certain internal information is also shown in this menu.

NOTICE
Memory storage (buffer) for the internal clock
Clock settings are retained for 3 ... 5 days after shutdown or mains failure.

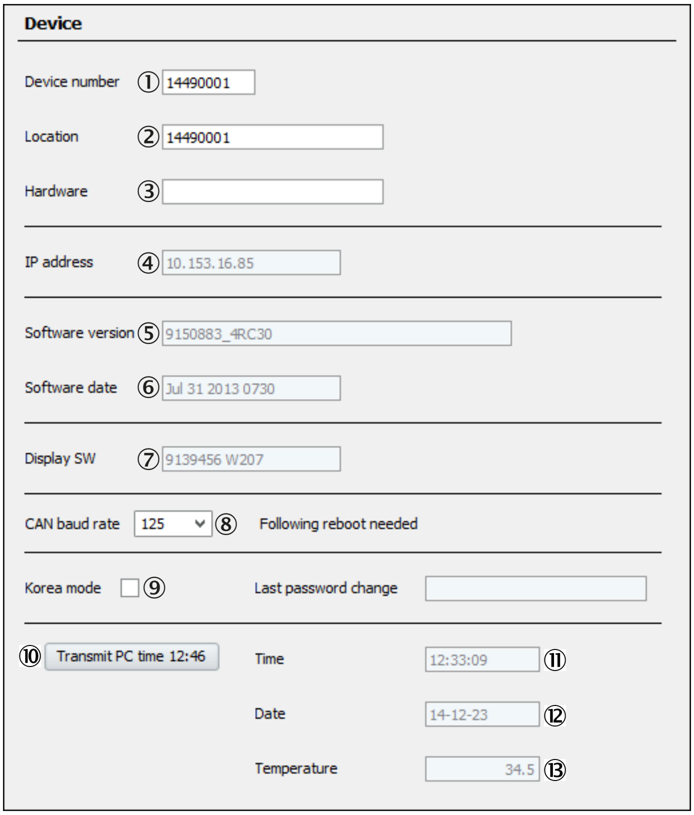
▶ If the GMS800 has been out of operation longer than 3 ... 5 days: Set the internal clock again after start-up.

Only observe this note when

- Logbook entries (see "Viewing the logbook", page 15) are evaluated using their time
- Procedures to be started by the internal clock are set up (e.g. automatic adjustments, sequence control programs with start by cyclic triggers).

Procedure

▶ Call up menu: **BCU/Parameter/Device.**



Device

Device number ① 14490001

Location ② 14490001

Hardware ③

IP address ④ 10.153.16.85

Software version ⑤ 9150883_4RC30

Software date ⑥ Jul 31 2013 0730

Display SW ⑦ 9139456 W207

CAN baud rate 125 ⑧ Following reboot needed

Korea mode □ ⑨ Last password change

⑩ Transmit PC time 12:46

Time 12:33:09 ⑪

Date 14-12-23 ⑫

Temperature 34.5 ⑬

Figure 23: Menu “Device” (example)

① Determined automatically
② Freely selectable text
③ Hardware identifier of BCU fitted
④ Current IP address of BCU[1]
⑤ Software version of BCU
⑥ Software version date
⑦ Display version
⑧ Transfer speed on internal CAN bus[2]
⑨ Korea mode
⑩ Transfer PC time to the BCU
⑪ Current time in GMS800 (in the BCU)
⑫ Current date in GMS800 (in the BCU)
⑬ Current temperature in GMS800 (in the BCU)

[1] Explanation and setting, see "IP address", page 9

[2] Can be set in user level “Service”.

7.3 Measuring screens in SOPAS ET

7.3.1 Configuring measuring screen contents

Function

- 8 different settings for measuring screens can be stored.
- 8 different layouts are available.
- Individual settings for data to be displayed.

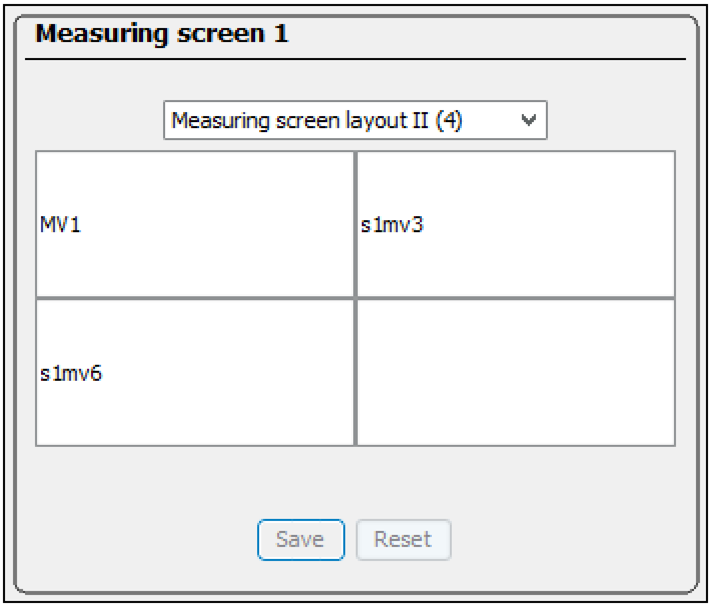
NOTE
These settings are only valid for measuring screens in SOPAS ET and not for the display in the GMS800.

Procedure

1. Call up menu: **BCU/Parameter/Measuring Screen/Measuring screen X** (X = 1 ... 8).
2. Select desired layout for the measuring screen.

✓ The layout fields are shown symbolically (example, see figure 24, page 31).

3. Enter the name of the tag (identification character sequence) of the value or state in the measured value field to be displayed there (Tag list, see "Applicable values and states (read tags)", page 74).
4. Select **Save**.



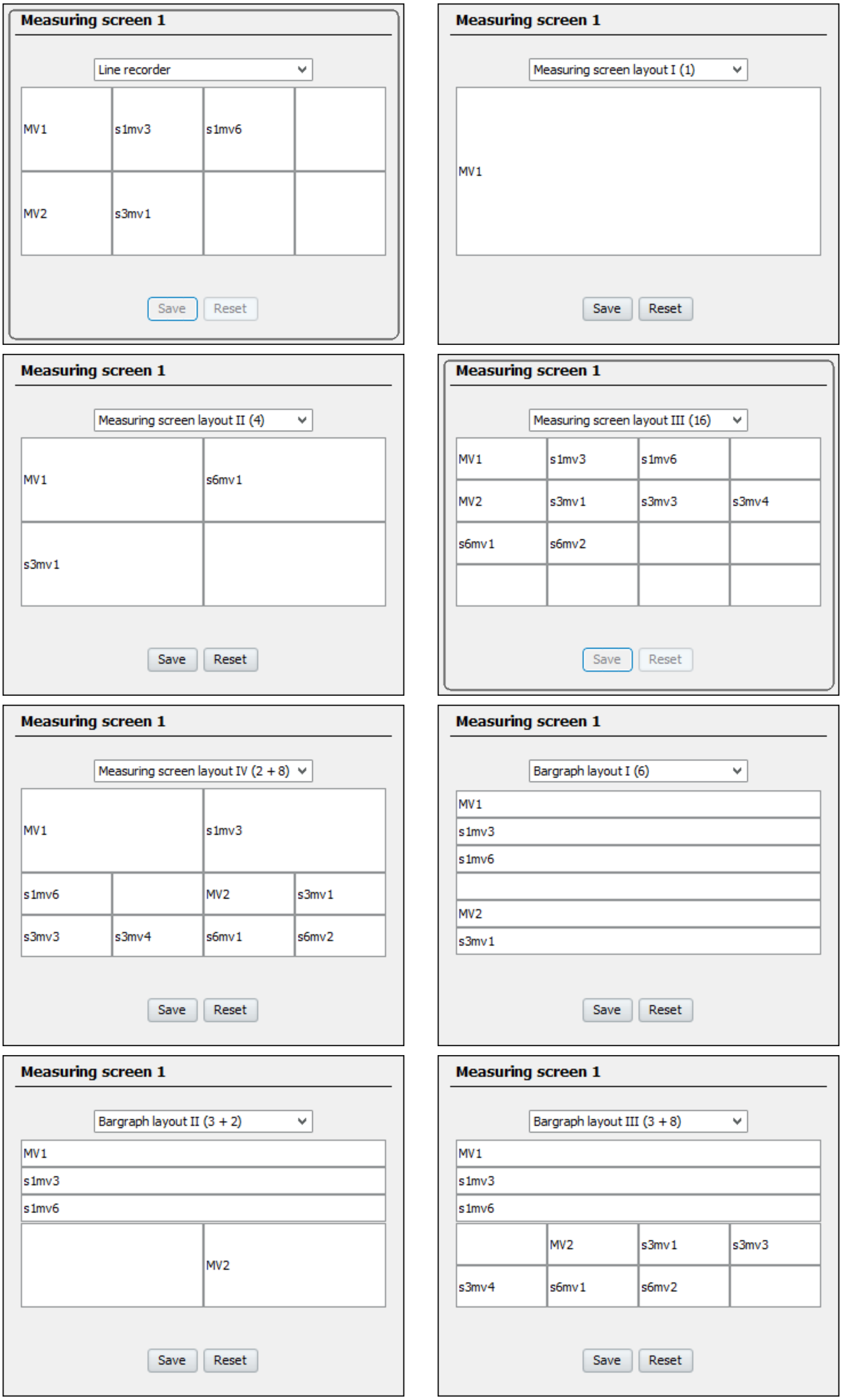
Measuring screen 1

Measuring screen layout II (4)

MV1	s1mv3
s1mv6	

Figure 24: Menu “Measuring screen” (example)

Designation (example)	Significance
MV1	Measured value No. 1, calculated by the BCU software
s1mv3	Measured value No. 3 from Sensor module No. 1



Measuring screen 1

Line recorder

MV1	s1mv3	s1mv6	
MV2	s3mv1		

Save Reset

Measuring screen 1

Measuring screen layout I (1)

MV1

Save Reset

Measuring screen 1

Measuring screen layout II (4)

MV1	s6mv1
s3mv1	

Save Reset

Measuring screen 1

Measuring screen layout III (16)

MV1	s1mv3	s1mv6	
MV2	s3mv1	s3mv3	s3mv4
s6mv1	s6mv2		

Save Reset

Measuring screen 1

Measuring screen layout IV (2 + 8)

MV1		s1mv3	
s1mv6		MV2	s3mv1
s3mv3	s3mv4	s6mv1	s6mv2

Save Reset

Measuring screen 1

Bargraph layout I (6)

MV1
s1mv3
s1mv6
MV2
s3mv1

Save Reset

Measuring screen 1

Bargraph layout II (3 + 2)

MV1	
s1mv3	
s1mv6	
	MV2

Save Reset

Measuring screen 1

Bargraph layout III (3 + 8)

MV1			
s1mv3			
s1mv6			
	MV2	s3mv1	s3mv3
s3mv4	s6mv1	s6mv2	

Save Reset

Table 2: Layouts for measuring screens in SOPAS ET (with example tags)

7.3.2 Configuring single measuring screens

Function

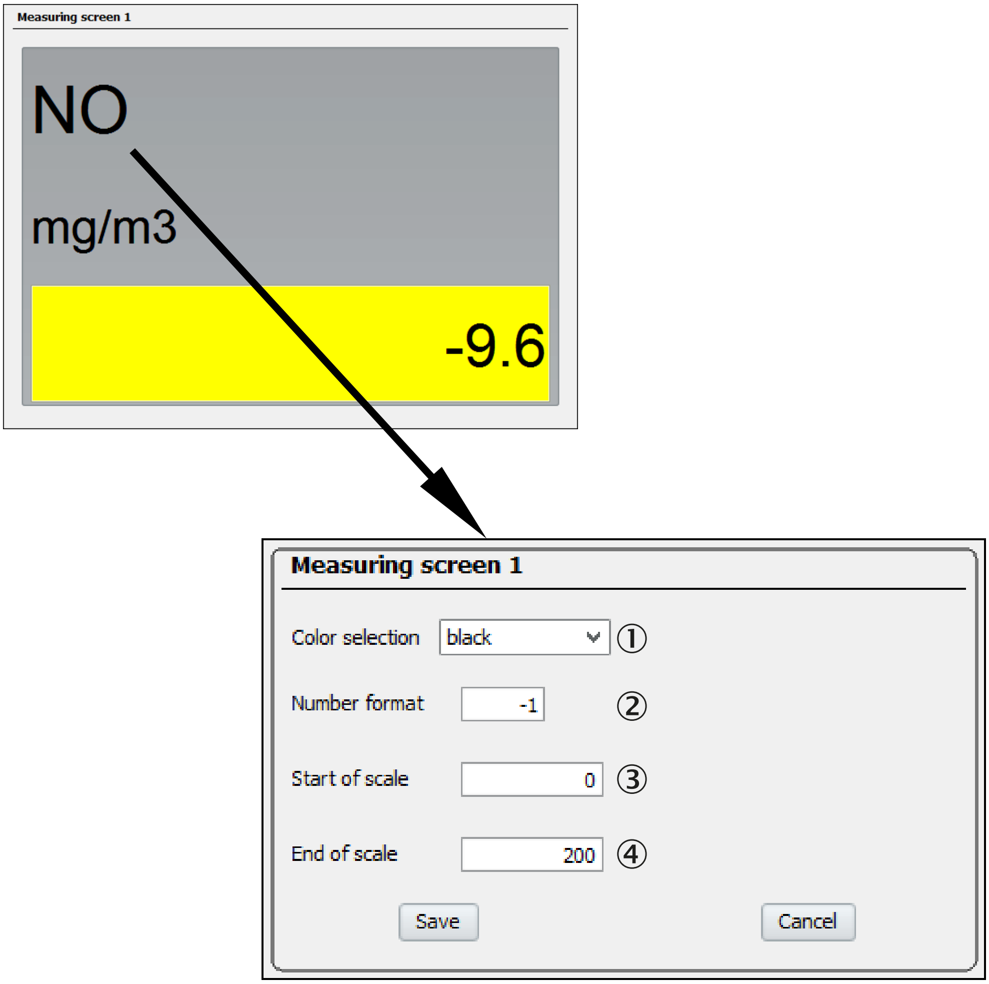
The following can be set for each single measuring screen:

- Font color for name and physical unit
- Number of decimal places
- Display area for bargraphs.

These settings are not stored permanently. The measured value configuration (see "Configuring measured values", page 52) overwrites the number of decimal places setting.

Procedure

1. Select a measuring screen (see "Selecting a measuring screen", page 13).
2. Doubleclick in the upper area of the measuring screen.
3. Make the desired settings.
4. Select **Save**.



Measuring screen 1

NO

mg/m3

-9.6

Measuring screen 1

Color selection: black

Number format: -1

Start of scale: 0

End of scale: 200

Figure 25: Menu for configuring a single measuring screen (example)

① Font color for name and physical unit

②
- "1": Decimal format
- "2": Exponential format for basis E6
- Negative number: Number of decimal places

③ Start value for bargraph display and chart recorder[1]

④ End value for bargraph display and chart recorder[1]

[1] For this measured value. Has no effect on numeric measuring screens.

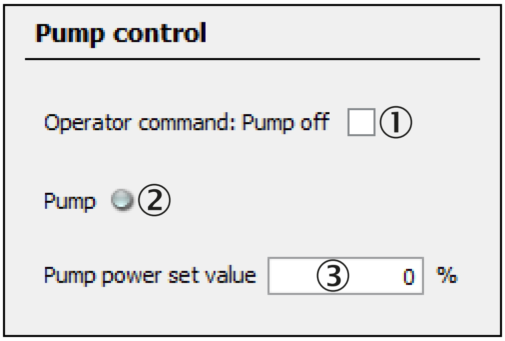
7.4 Controlling the pump

Functions

- The pump can be switched off with a menu function.
- The capacity supply for the fitted pump (option) is adjustable. This serves to determine the pump flow rate.

Procedure

1. Call up menu: **BCU/Parameter/Pump control.**
2. Make the desired settings.



Pump control

Operator command: Pump off □ ①

Pump ● ②

Pump power set value ③ 0 %

Figure 26: Menu “Pump control” (example)

① ☑ = pump switched off
② LED symbol “on” = pump switched on
③ Electronic set point value for pump capacity[1]

[1] Set the set point value so that the desired volume flow is attained.

NOTICE
Even when the pump is switched on in this menu, the pump can actually be deactivated by further safety functions.

NOTE
If the GMS800 has a sample gas pump fitted:

▶ Use this menu function to set the desired sample gas volume flow.

This minimizes the pump load as far as possible and lengthens the service life.

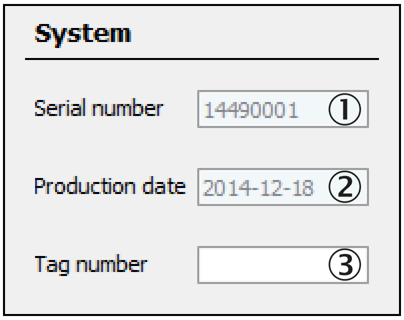
7.5 Check system (measuring system)/enter user identifier

Function

This menu displays the serial number and production date of the GMS800. An individual user identifier can be entered.

Procedure

1. Call up menu: **BCU/Parameter/System.**
2. As required: Enter text for **Tag Number.**



System

Serial number 14490001

Production date 2014-12-18

Tag number

Figure 27: Menu “System” (example)

①	Serial number
②	Production date
③	User identifier for this GMS800[1]

[1] Any text. Not used internally.

8 Configuring the I/O

8.1 Introduction to configuring the I/O

8.1.1 Number of I/O connections

The number of inputs and outputs available depends on whether the GMS800 is fitted with one or two I/O modules (display, see "Checking the I/O modules", page 17). Menus are automatically adapted accordingly.

8.1.2 Live view

If the “Live view” option is activated in the menus, the menus always show the current GMS800 parameters, even when the parameters are changed by an external location (e.g. via the BCU display).

“Live view” must be deactivated when the parameters are to be changed using SOPAS ET.

8.1.3 Tags and formulas in I/O parameters

For some parameters for inputs and outputs, the internal values and states of the tags (identification characters) are used. Apart from that, it is possible to use parameters determined by the programmed formulas.

NOTE

Recommendation:

▶ Read the Section “Introduction to formulas” before configuring I/O parameters (see "Introduction to formulas", page 80).

8.1.4 Source of an output signal

The current state of a digital or analog output corresponds to the current value of the assigned “source”.

Rules

- Input is only possible when the “Live view” is deactivated.
- Usable tags are shown in the context help.

Calculated value as source

How to use a value calculated with a formula as “source”:

1. Define the desired value calculation as assignment to a variable (for digital outputs: BVi, for analog outputs: MVi).
   Schema: XVi=[formelterm]
2. Enter this variable as “source” during configuration.

NOTE

A calculated value in a formula assigned directly to an output has no effect.

8.2 Menu functions for configuring I/O

8.2.1 Configuring digital inputs (DIi)

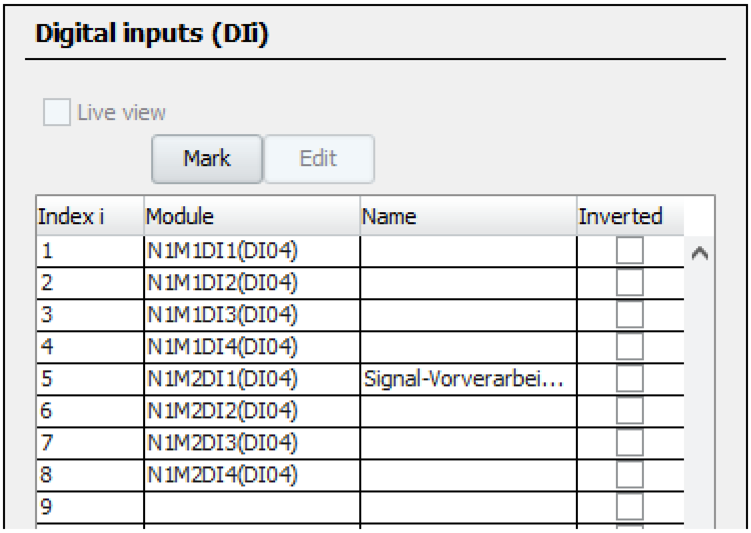
Function

The internal switching logic of the digital inputs (switching inputs) can be inverted. Each digital input can be named for identification.

NOTE
The current state of the digital inputs can be processed in formulas with tags “DIi” and “DIiI” (see "Applicable values and states (read tags)", page 74).

Procedure

1. Call up menu: **BCU/Parameter/I/O/Digital inputs (DIi).**



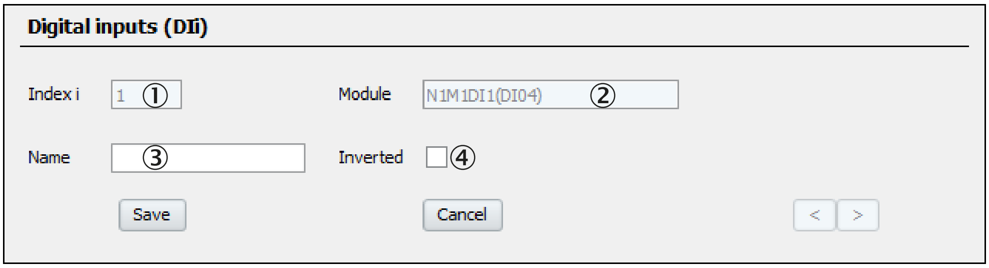
Digital inputs (DIi)

□ Live view

Index i	Module	Name	Inverted
1	N1M1DI1(DI04)		□
2	N1M1DI2(DI04)		□
3	N1M1DI3(DI04)		□
4	N1M1DI4(DI04)		□
5	N1M2DI1(DI04)	Signal-Vorverarbei...	□
6	N1M2DI2(DI04)		□
7	N1M2DI3(DI04)		□
8	N1M2DI4(DI04)		□
9			□

Figure 28: Menu “Digital inputs (DIi)” - Table (example)

2. Deactivate **Live View.**
3. Mark desired Table rows.
4. Select **Edit.**



Digital inputs (DIi)

Index i	1 ①	Module	N1M1DI1(DI04) ②
Name	③	Inverted	□ ④

Figure 29: Menu “Digital inputs (DIi)” - Edit (example)

① Connection number (1 = DI1, 2 = DI2 etc.)
② Internal I/O address[1]
③ Connection name (freely selectable text, max. 16 characters)
④ ☑= logical input status is inverse to electronic status

[1] Only digital inputs with a module identifier are present in the GMS800 (see "Number of I/O connections", page 36).

5. Enter desired data.
6. Select **Save.**

8.2.2 Configuring digital outputs (DOi)

Function

Digital outputs are activated by an internal “source” (see table 3, page 39). The activation logic can be inverted (see "Switching logic", page 38). Each digital output can be named for identification.

Switching logic

Each digital output can function with normal or inverted activation logic:

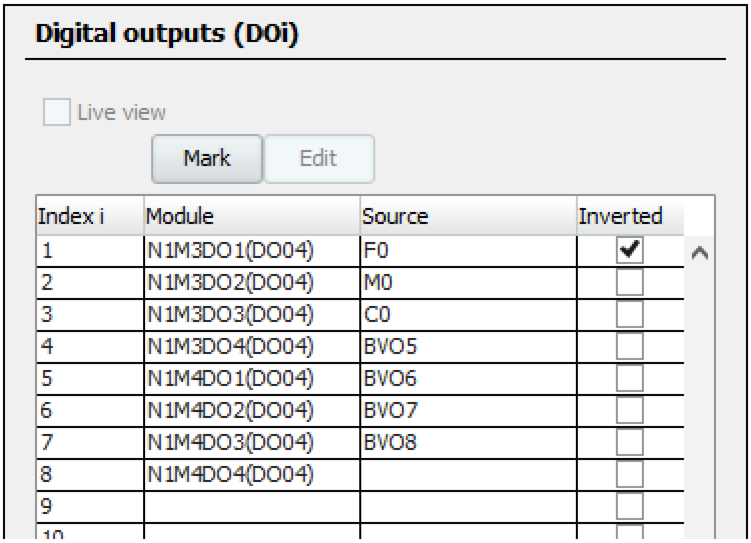
- Normal activation logic: The digital output is activated electronically (open-circuit principle) when the associated switching function is in an active state.
- Inverse activation logic: The digital output is activated electronically (closed-circuit principle) when the associated switching function is logically inactive. The switching output is in an inactive electronic state when the switching function is logically active.

CAUTION
Risks for connected devices/systems

- ▶ Before using the digital outputs, clarify the safety-relevant consequences of the following operational malfunctions:
  - – BCU power failure (e.g. local power failure or accidental switching-off or defective fuse)
  - – Electronic defect on switching output
  - – Interruption of the electrical connection
- ▶ Select the switching logic considering safety aspects:
  - – Switching outputs with normal activation logic signal that the respective switching function is not active when the mains voltage fails.
  - – Switching outputs with inverse activation logic signal that the respective switching function is triggered when the mains voltage fails.
- ▶ Carefully clarify the consequences and make sure no dangerous situation can arise when a failure or defect occurs.

Procedure

1. Call up menu: **BCU/Parameter/I/O/Digital outputs (DOi)**.



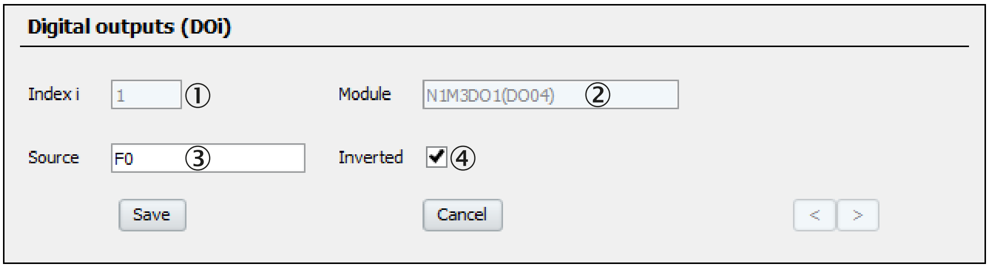
Digital outputs (DOi)

☐ Live view

Index i	Module	Source	Inverted
1	N1M3DO1(DO04)	F0	☑
2	N1M3DO2(DO04)	M0	☐
3	N1M3DO3(DO04)	C0	☐
4	N1M3DO4(DO04)	BVO5	☐
5	N1M4DO1(DO04)	BVO6	☐
6	N1M4DO2(DO04)	BVO7	☐
7	N1M4DO3(DO04)	BVO8	☐
8	N1M4DO4(DO04)		☐
9			☐
10			☐

Figure 30: Menu “Digital outputs (DOi)” - Table (example)

2. Deactivate **Live view**.
3. Mark desired Table rows.
4. Select **Edit**.



Digital outputs (DOi)

Index i: 1 ①
Module: N1M3DO1(DO04) ②
Source: F0 ③
Inverted: ☑ ④

Figure 31: Menu “Digital outputs (DOi)” – Edit (example)

① Connection number (1 = DO1, 2 = DOI2 etc.)
② Internal I/O address[1]
③ Source, the state of which controls the output (see table 3)
④ ☑ = electronic output status is inverse to logical source status

[1] Only digital inputs with a module identifier are present in the GMS800 (see "Number of I/O connections", page 36).

Tag	Function	Indices
X0	Group status for “X” messages (BCU status)	X = C/E/F/M/U
AOiR	Current measuring range of analog output AOi[1]	i = 1 ... 8
BVi	Value of Boolean variable BVi (see "Variables", page 80)	i = 1 ... 24
BVOi	Value of Boolean variable BVOi (see "Boolean variables (BVOi) – output signals", page 90)	i = 1 ... 24
MPS	Operating state for measuring point switchover[2]	–
MPiS	Operating state of measuring point i[3]	i = 0 ... 8
MPiPPS	Advance extraction	i = 0 ... 8

MViLIj	Limit value message for measured value MVi – limit value j	i = 1 ... 12
		j = 1 ... 2

MViX0	Group status for “MViXj” messages	i = 1 ... 12
		X = C/E/F/M/U

SiX0	Group status for “SiXj” messages	i = 1 ... 6
		X = C/E/F/M/U

Table 3: Sources for digital output control

[1] Measuring range 1 : AOiR = 0; measuring range 2: AOiR = 1
[2] “0” = flush, “1” = measure.
[3] “0” = off, “1” = active.

5. Enter desired data.
6. Select **Save**.

8.2.3 Configuring analog inputs (AIi)

Function

The analog inputs cover the signal range 0 ... 20 mA. The input signal can be used in measuring screens and formulas, either as electronic value (0 ... 20 mA) or as physical value according to the conversion (see "Applicable values and states (read tags)", page 74).

NOTE

The NAMUR status of each measured value is provided as information.

- 0 mA: Failure (F0)
- 2 mA: Check (C0)
- 4 - 20 mA: Measured value in selected display range

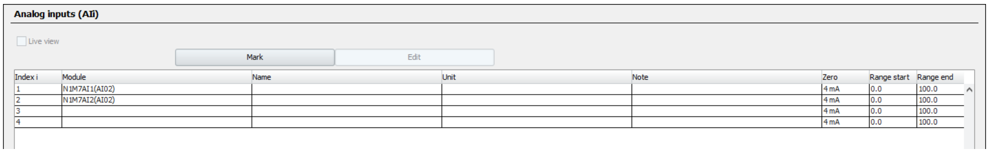
Conversion

- The “Live Zero” (zero) of the input signal is adjustable (0/2/4 mA or 4 mA, F = 0 mA, C = 2 mA).
- The following formula calculates the physical input value (AIi) internally:

$$\text{AIi} = \left[(\text{MB Ende - MB Anfang}) * \frac{\text{AIiI [mA] - Zero [mA]}}{\text{20 - Zero [mA]}} \right] + \text{MB Anfang}$$

Procedure

1. Call up menu: **BCU/Parameter/I/O/Analog inputs (AIi).**



Analog inputs (AIi)

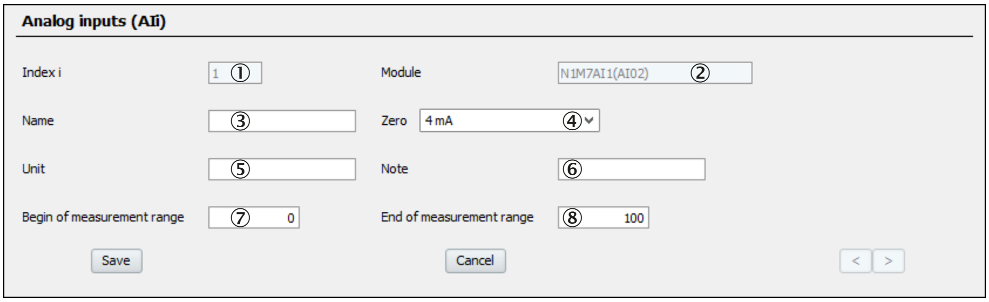
□ Live view

Mark | Edit

Index i	Module	Name	Unit	Note	Zero	Range start	Range end
1	N1M7AI1(AI02)				4 mA	0.0	100.0
2	N1M7AI2(AI02)				4 mA	0.0	100.0
3					4 mA	0.0	100.0
4					4 mA	0.0	100.0

Figure 32: Menu “Analog inputs (AIi)” - Table (example)

2. Deactivate **Live View.**
3. Mark desired Table rows.
4. Select **Edit.**



Analog inputs (AIi)

Index i	1 ①	Module	N1M7AI1(AI02) ②
Name	③	Zero	4 mA ④
Unit	⑤	Note	⑥
Begin of measurement range	⑦ 0	End of measurement range	⑧ 100

Figure 33: Menu “Analog inputs (AIi)” - Edit (example)

① Connection number (1 = AI1, 2 = AI2 etc.)
② Internal I/O address[1]
③ Name (freely selectable text, max. 16 characters)[2]
④ Electronic zero point of the input signal (0/2/4 mA or 4 mA, F = 0 mA, C = 2 mA)
⑤ Physical unit (freely selectable text, max. 16 characters)
⑥ Comment (freely selectable text, max. 16 characters)
⑦ Physical value corresponding to electronic zero point
⑧ Physical value corresponding to input signal “20 mA”

[1] Only analog inputs with a module identifier are present in the GMS800 (see "Number of I/O connections", page 36).

[2] Recommendation: Use the designation of the physical variable (e.g. “pressure”).

5. Enter desired data.
6. Select **Save.**

8.2.4 Configuring analog outputs (AOi)

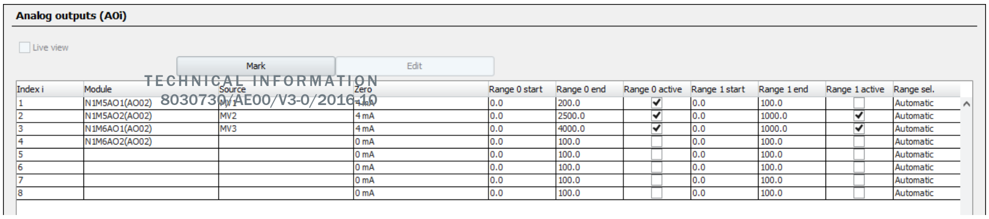
Function

The following can be set for each analog output:

- Measured value to be output (internal source of the measured value)
- Whether one or two output ranges are to be used
- Start and end values of the output ranges
- The "live zero" (zero)

Procedure

1. Call up menu: **BCU/Parameter/I/O/Analog outputs (AOi).**



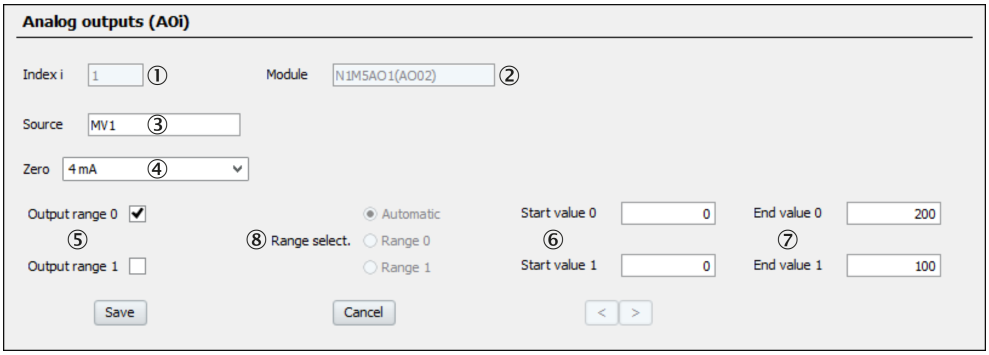
Analog outputs (AOi)

□ Live view

Index i	Module	Source	Zero	Range 0 start	Range 0 end	Range 0 active	Range 1 start	Range 1 end	Range 1 active	Range sel.
1	N1M5AO1(AO02)	MV1	4 mA	0.0	200.0	☑	0.0	100.0	□	Automatic
2	N1M5AO2(AO02)	MV2	4 mA	0.0	2500.0	☑	0.0	1000.0	☑	Automatic
3	N1M6AO1(AO02)	MV3	4 mA	0.0	4000.0	☑	0.0	1000.0	☑	Automatic
4	N1M6AO2(AO02)		0 mA	0.0	100.0	□	0.0	100.0	□	Automatic
5			0 mA	0.0	100.0	□	0.0	100.0	□	Automatic
6			0 mA	0.0	100.0	□	0.0	100.0	□	Automatic
7			0 mA	0.0	100.0	□	0.0	100.0	□	Automatic
8			0 mA	0.0	100.0	□	0.0	100.0	□	Automatic

Figure 34: Menu "Analog outputs (AOi)" – Table (example)

2. Deactivate **Live View.**
3. Mark desired Table rows.
4. Select **Edit.**



Analog outputs (AOi)

Index i: 1 ①

Module: N1M5AO1(AO02) ②

Source: MV1 ③

Zero: 4 mA ④

Output range 0 ☑ ⑤ Output range 1 □

⑧ Range select. ● Automatic ○ Range 0 ○ Range 1

Start value 0: 0 ⑥ Start value 1: 0

End value 0: 200 ⑦ End value 1: 100

Figure 35: Menu "Analog outputs (AOi)" – Edit (example)

① Connection number connection (1 = AO1, 2 = AO2 etc.)
② Internal I/O address[1]
③ Internal source of output value (see table 4, page 42)
④ Electronic zero point of output signal[2]
⑤ ☑ = output range X used during operation[3]
⑥ Physical value corresponding to electronic zero point
⑦ Physical value corresponding to output signal "20 mA"[4]
⑧ **Automatic,** see "Range selection automatic", page 42
Range 0/Range 1 = output range set fixed

[1] Only analog outputs with a module identifier are present in the GMS800 (see "Number of I/O connections", page 36).

[2] 4 mA (F: 0 mA / C: 2 mA) means: Analog output outputs 0 mA when status flag "F" (fault) is activated. Analog output outputs 2 mA when status flag "C" (check) is activated.

[3] Output range 1 is used automatically when no output range is marked "active"; automatic output range switch-over is active when both output ranges are "active" (see "Range selection automatic", page 42).

[4] For automatic output range switch-over, end value 1 must be larger than end value 0.

Tag	Function	Indices
MVi	Current measured value MVi[1]	i = 1 ... 12

MViMPj	Measured value MVi from measuring point MPj[1]	i = 1 ... 12
		j = 1 ... 8

Table 4: Signal value sources for analog outputs

[1] Explanations, see "Measured value configuration", page 51.

5. Enter desired data.
6. Select **Save**.

Range selection automatic

- The smallest output range is used automatically when the current measured value is within this output range (resp. in the output range with the smallest measurement span).
- The larger output range is activated automatically when the measured value exceeds the smaller output range.
- The smaller output range is activated again when the measured value is within the smaller output range again and underflows the limit of the smaller output range by 10% (relative to the measurement span of the smaller output range).

9 Adjustment functions

9.1 Configuring test gases (Test gas Table)

Function

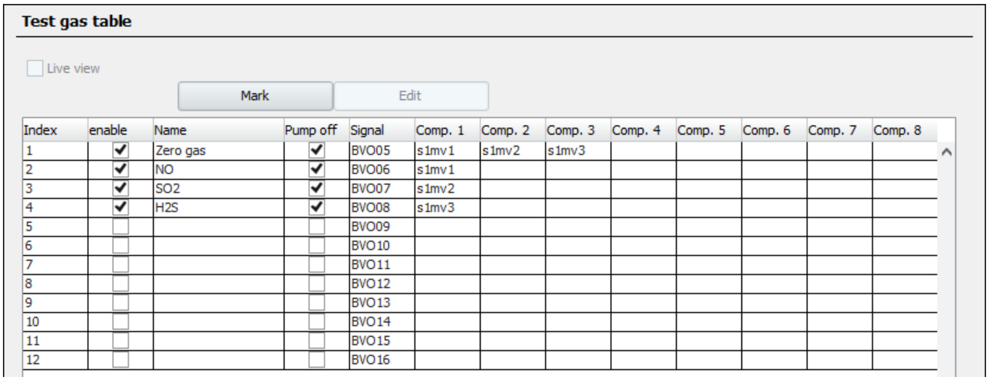
The Test gas Table is the basis for adjustments. 12 different test gas settings can be programmed. Each test gas setting can be used for up to 8 components. The test gas settings can also be used for validation measurements.

NOTE

- The same real test gas can be used in several test gas settings. This allows using a certain test gas for different adjustment procedures.
- Recommendation: Only program one adjustment or validation function in each test gas setting.
- Suitable test gas settings are normally programmed at the manufacturer's factory.

Procedure

1. Cal up menu: **BCU/Parameter/Test Gas Table.**

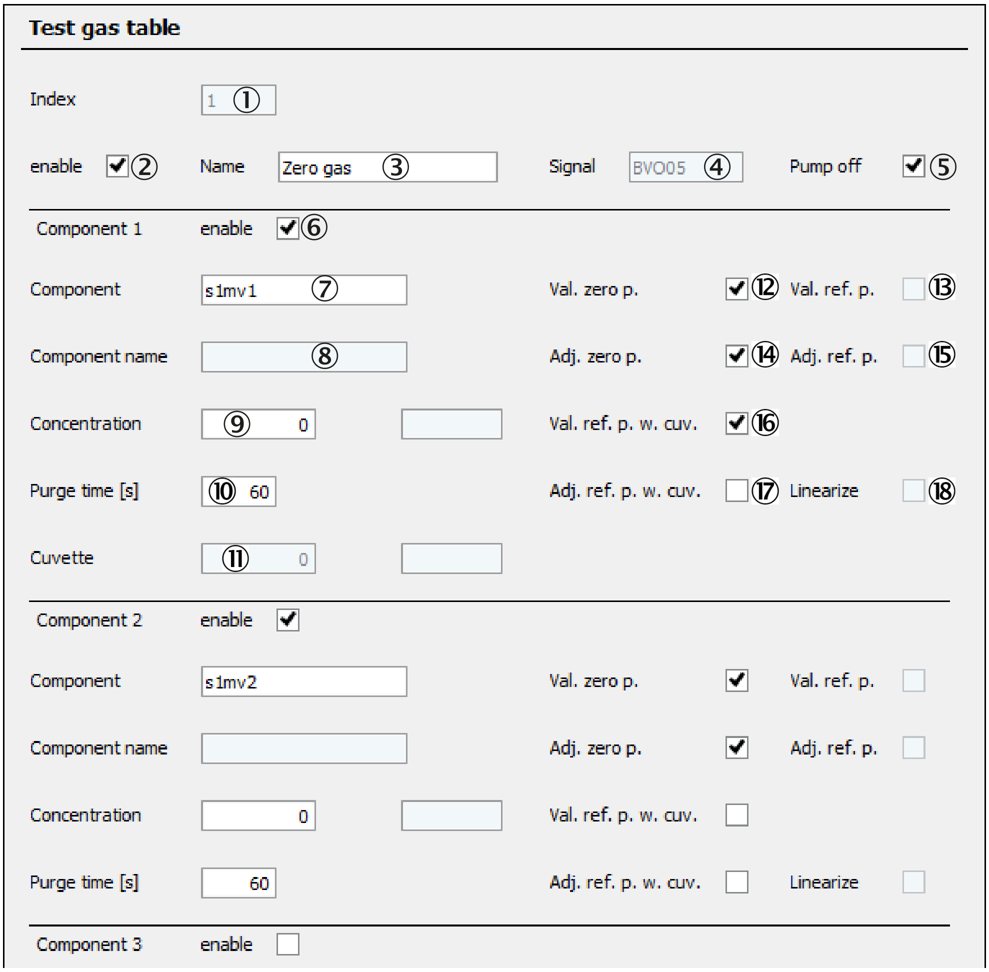


Test gas table

Index	enable	Name	Pump off	Signal	Comp. 1	Comp. 2	Comp. 3	Comp. 4	Comp. 5	Comp. 6	Comp. 7	Comp. 8
1	☑	Zero gas	☑	BVO05	s1mv1	s1mv2	s1mv3					
2	☑	NO	☑	BVO06	s1mv1							
3	☑	SO2	☑	BVO07	s1mv2							
4	☑	H2S	☑	BVO08	s1mv3							
5	□		□	BVO09								
6	□		□	BVO10								
7	□		□	BVO11								
8	□		□	BVO12								
9	□		□	BVO13								
10	□		□	BVO14								
11	□		□	BVO15								
12	□		□	BVO16								

Figure 36: Menu “Test Gas Table“ – Table (example)

2. Deactivate Live View
3. Mark desired Table rows.
4. Select **Edit.**



Test gas table

Index	1						
enable	☑	Name	Zero gas	Signal	BVO05	Pump off	☑

Component 1		enable	☑				
Component	s1mv1			Val. zero p.	☑	Val. ref. p.	□
Component name				Adj. zero p.	☑	Adj. ref. p.	□
Concentration	0			Val. ref. p. w. cuv.	☑		
Purge time [s]	60			Adj. ref. p. w. cuv.	□	Linearize	□
Cuvette	0						
Component 2		enable	☑				
Component	s1mv2			Val. zero p.	☑	Val. ref. p.	□
Component name				Adj. zero p.	☑	Adj. ref. p.	□
Concentration	0			Val. ref. p. w. cuv.	□		
Purge time [s]	60			Adj. ref. p. w. cuv.	□	Linearize	□
Component 3		enable	□				

Figure 37: Menu “Test gas Table” – Edit (example)

① Test gas number (cannot be changed).

② ☑ = this test gas can be used for adjustments/validations.

③ Test gas name (freely selectable text, max. 20 characters).

④ Boolean variable for controlling this test gas (cannot be changed).

⑤ ☑ = gas pump switched off automatically when this test gas is used.

⑥ ☑ = this test gas can be used for the components specified.

⑦ Tag of component for which this test gas is to be used.[1]

⑧ Component name in Sensor module (cannot be changed).

⑨ Test gas set point value in the measuring component physical unit.
▸ Use the decimal point (.) as decimal character.

⑩ Wait time after switching to this test gas. The measurement for the adjustment / validation first starts after the flush time / wait time.[2]

⑪ Cuvette concentration of the adjustment cuvette

⑫ ☑ = test gas used for zero point validations.
Simultaneous use for reference point validations not possible.

⑬ ☑ = test gas used for reference point validations.
Simultaneous use for zero point validations or reference point validations with adjustment cell not possible.

⑭ ☑ = test gas used for zero point adjustments.
Simultaneous use for reference point adjustments not possible.

⑮ ☑ = test gas used for reference point adjustments.
Simultaneous use for zero point adjustments or reference point adjustments with adjustment cell not possible.

⑯ ☑ = test gas used for reference point validations with adjustment cuvette. Simultaneous use for reference point validations not possible.

⑰ ☑ = test gas used for reference point adjustments with adjustment cuvette. Simultaneous use for reference point adjustments not possible.

⑱ ☑ = test gas used for linearity adjustments. Simultaneous use for other adjustments not possible.

[1] Tag schema: SiMVj (i = Sensor module number, j = measured value number in Sensor module)

[2] The longest flush time is effective when the test gas is used for several components.

NOTE
Settings are first effective after **Save** has been selected.

9.2 Performing manual adjust

Function

For "manual adjust", a single adjustment or validation procedure is selected and started manually.

NOTE
The standard setting with checkbox "Automatic" selected (see figure 38, page 47) runs with the test gas fed automatically via solenoid valves controlled by the GMS800 digital outputs. The controlling output is defined in each test gas setting ("Action on start", see "Configuring test gases (Test gas Table)", page 43). The test gas can however also be fed manually.

NOTE
Detailed information concerning adjustments and validations → "GMS800 Series" Operating Instructions

Procedure

▶ Inform connected stations that measuring operation will be interrupted.

1. Call up menu: **BCU**/**Maintenance**/**Manual adjust**.
2. Select the measuring component for which this procedure is to be applicable ([<<] [>>]).
3. Select desired function ([<<] [>>]).

✓ The **Start** button is active when a suitable test gas setting has been found for the measuring component and function combination.

4. Select **Start**.

With manual test gas feed

✓ Current status = **Test gas**

5 Feed suitable test gas into the sample gas inlet of the GMS800.

6 Wait until **Sample gas** is shown as current status.

7 Feed the sample gas again into the sample gas inlet.[1]

8 Wait until **Stop** is shown as current status.

With automatic test gas feed

✓ The automatic procedure starts. Current status = part of the procedure actually running (see table 5, page 46)

5 Wait until **Stop** is shown as current status.

✓ Manual adjust is completed.

✓ Manual adjust is completed.

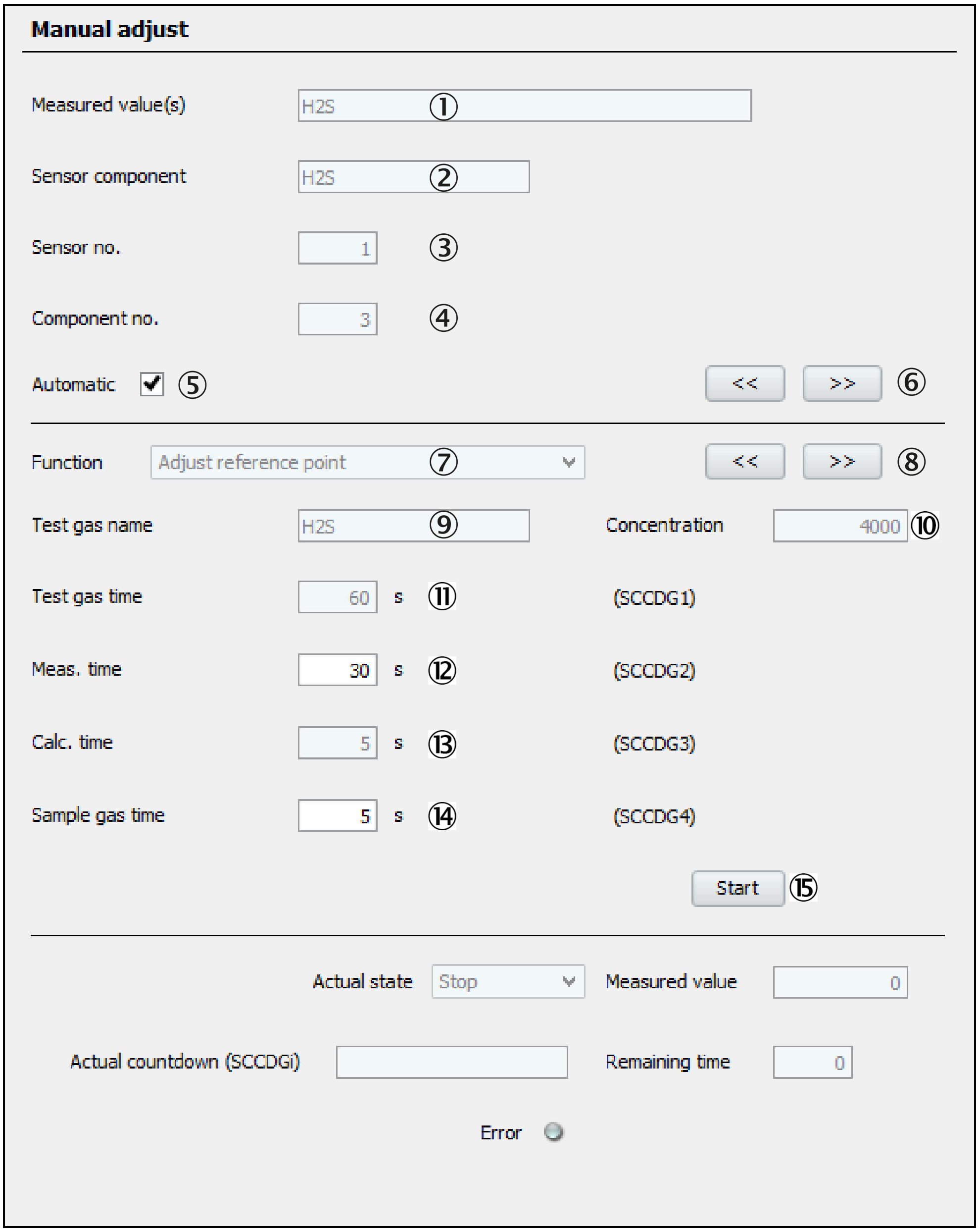
[1] Alternative (if a further manual adjust should follow): The test gas for the next manual adjust

Actual state	Internal function
Stop	Function paused
Test gas	Wait until flush time elapses (after switching to test gas)
Measure	Determine measured values with test gas
Calculate	Determine the mean value during the measurement time, calculate deviation from set point value, adapt adjustment
Sample gas	Wait until flush time elapses (after switching to sample gas)

Table 5: Process flow phases during manual adjust

Designation	Significance
Actual state	Part of procedure currently running (see table 5, page 46)
Measured value	Current measured value of component
Actual countdown (SCCGi)	Name of active countdown timer
Remaining time	Remaining time of active countdown timer

Table 6: Information in menu ***Manual adjust*** *during the procedure*



Manual adjust

Measured value(s): H2S ①

Sensor component: H2S ②

Sensor no.: 1 ③

Component no.: 3 ④

Automatic ☑ ⑤ << >> ⑥

Function: Adjust reference point ⑦ << >> ⑧

Test gas name: H2S ⑨ Concentration: 4000 ⑩

Test gas time: 60 s ⑪ (SCCDG1)

Meas. time: 30 s ⑫ (SCCDG2)

Calc. time: 5 s ⑬ (SCCDG3)

Sample gas time: 5 s ⑭ (SCCDG4)

Start ⑮

Actual state: Stop Measured value: 0

Actual countdown (SCCDGi): Remaining time: 0

Error ○

Figure 38: Menu “Manual adjust” (example)

① Name of measured values calculated from this sensor component
② Measuring component for which this manual adjust is applicable
③ Number of Sensor module with which the measuring component is measured
④ Number of the sensor component in the Sensor module
⑤ ☑ = standard setting: Adjustment runs with preset time sequences.
□ = adjustment steps started singly by the user (manual test gas feed).
⑥ Select measured value(s)
⑦ Adjustment or validation to be performed (options, see figure 37, page 44)
⑧ Select function
⑨ Name of test gas to be used[1]
⑩ Set point of test gas / concentration of adjustment cuvette
⑪ Delay time after switching to the test gas; measurement starts when this delay has elapsed
⑫ Determines how long measured values of the test gas are measured[2]
⑬ Internal interval to compute the values measured

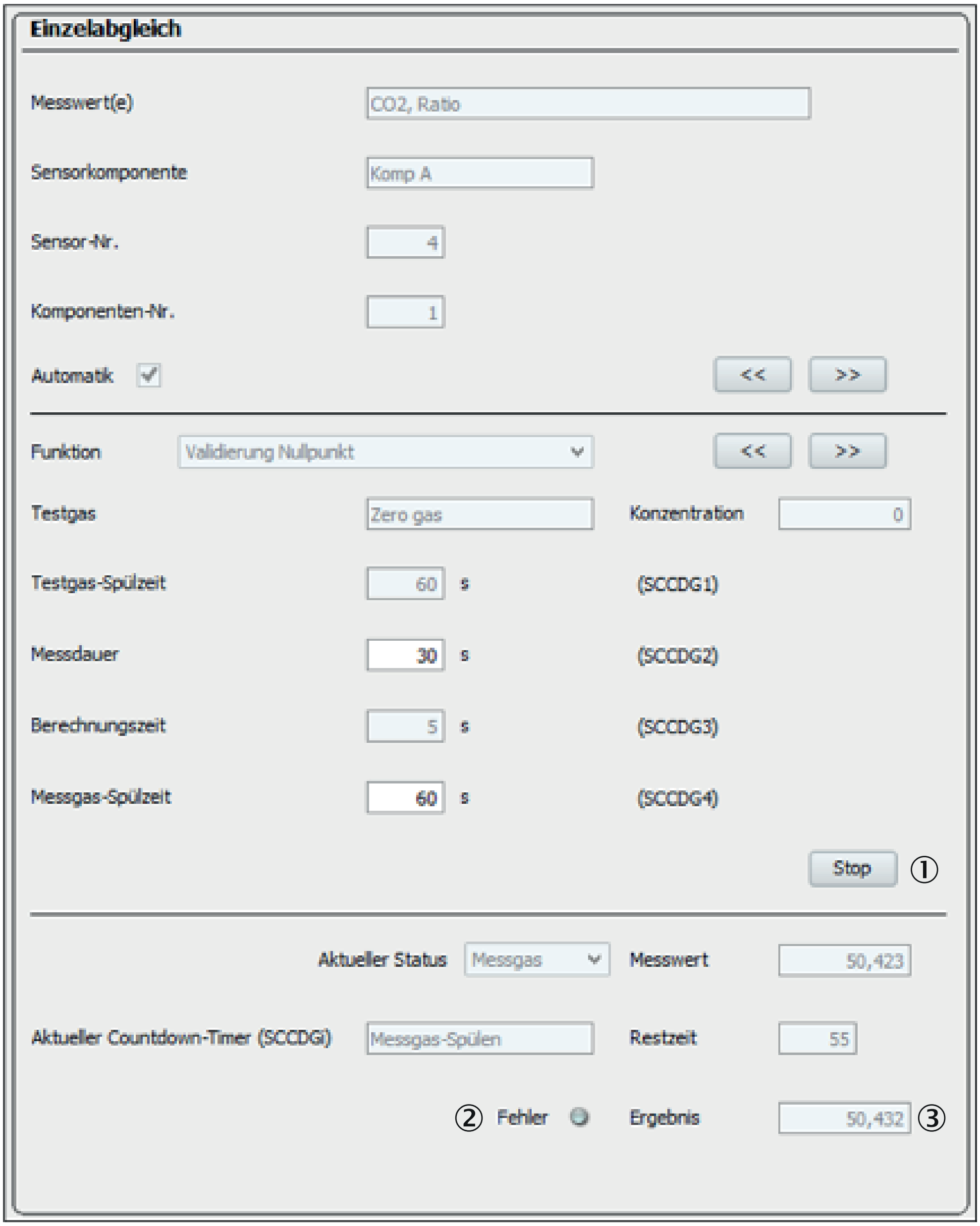
⑭ Delay time after switching to sample gas; the manual adjust has then completed after this time has elapsed

⑮ Start selected manual adjust

[1] Empty box: No suitable test gas setting programmed for the selected function

[2] Actual value for the adjustment = mean value of measured values within the measuring duration

Result



Einzelabgleich

Messwert(e): CO2, Ratio

Sensorkomponente: Komp A

Sensor-Nr.: 4

Komponenten-Nr.: 1

Automatik ☑

Funktion: Validierung Nullpunkt

Testgas: Zero gas

Konzentration: 0

Testgas-Spülzeit: 60 s (SCCDG1)

Messdauer: 30 s (SCCDG2)

Berechnungszeit: 5 s (SCCDG3)

Messgas-Spülzeit: 60 s (SCCDG4)

Aktueller Status: Messgas

Messwert: 50,423

Aktueller Countdown-Timer (SCCDG): Messgas-Spülen

Restzeit: 55

Fehler ●

Ergebnis: 50,432

Figure 39: Menu “Manual adjust” with result (example)

① Cancel active manual adjust

② Error is displayed when no result is available. Erroneous sequence or drift exceeded.

③ The measured actual measured value

9.3 Automatic adjustments/validations

9.3.1 Function of automatic adjustments/validations

8 adjustment and validation procedures that can run fully automatically can be programmed. Each procedure uses one of the test gases for which parameters have been set in the Test Gas Table (see "Configuring test gases (Test gas Table)", page 43).

Which measuring components are to be adjusted or validated with the procedure and which adjustment or validation function is to be performed can be programmed. Several components can be adjusted in one function.

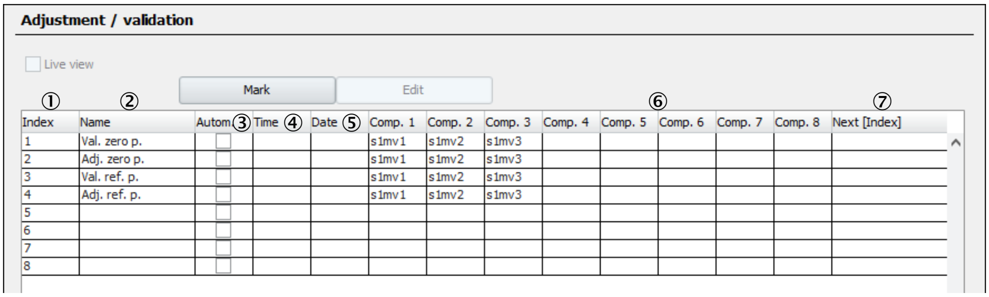
9.3.2 Start options

- Manual start:
  Use menu **Operator commands** (see "Starting adjustment/validation (operator commands)", page 26).
- Automatic start (in regular intervals):
  Use a cyclic trigger in the start condition (see "Setting a cyclic trigger", page 98).
- Remote controlled start:
  Use the value of a Boolean variable (BVIi) to control the value of a digital input (see "Boolean variables (BVIi) – input signals", page 88).
- Programmed start conditions:
  Calculate the value of a Boolean variable (BVIi) with a formula (see "Boolean variables (BVIi) – input signals", page 88).

Layout of adjustment and validation results, see "Adjustment results", page 17.

9.3.3 Programming automatic adjustments/validations

1. Call up menu: **BCU/Parameter/Adjustment/Validation.**



Adjustment / validation

□ Live view

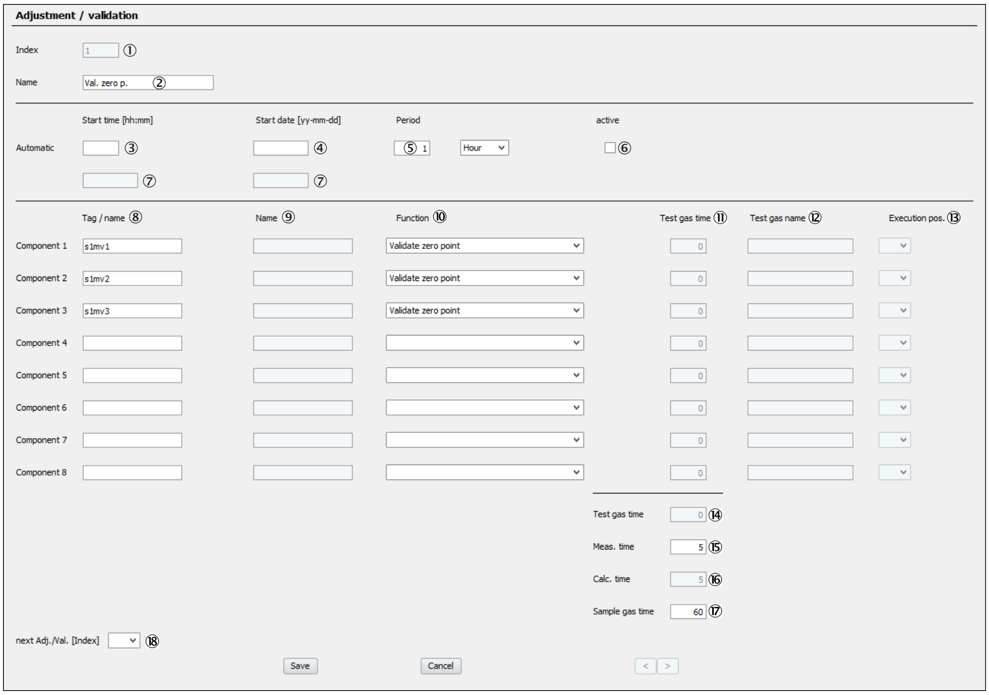
Mark | Edit

Index	Name	Autom.	Time	Date	Comp. 1	Comp. 2	Comp. 3	Comp. 4	Comp. 5	Comp. 6	Comp. 7	Comp. 8	Next [Index]
1	Val. zero p.	□			s1mv1	s1mv2	s1mv3						
2	Adj. zero p.	□			s1mv1	s1mv2	s1mv3						
3	Val. ref. p.	□			s1mv1	s1mv2	s1mv3						
4	Adj. ref. p.	□			s1mv1	s1mv2	s1mv3						
5		□											
6		□											
7		□											
8		□											

Figure 40: Menu “Adjustment/Validation” – Table (example)

① Index number of adjustment or validation procedure
② Programmed name
③ ☑ = automatic starts activated
④ Start time of next automatic function
⑤ Start date of next automatic function
⑥ Component X addressed by the respective function
⑦ Number of the next sequential function that starts (only when procedures are chained)

2. Deactivate **Live View.**
3. Mark desired Table rows.
4. Select **Edit.**



Adjustment / validation

Index 1 ①

Name Val. zero p. ②

	Start time [hh:mm]	Start date [yy-mm-dd]	Period		active
Automatic	③	④	⑤ 1	Hour	□ ⑥
	⑦	⑦			

	Tag / name ⑧	Name ⑨	Function ⑩	Test gas time ⑪	Test gas name ⑫	Execution pos. ⑬
Component 1	s1mv1		Validate zero point	0		
Component 2	s1mv2		Validate zero point	0		
Component 3	s1mv3		Validate zero point	0		
Component 4				0		
Component 5				0		
Component 6				0		
Component 7				0		
Component 8				0		

Test gas time 0 ⑭

Meas. time 5 ⑮

Calc. time 5 ⑯

Sample gas time 60 ⑰

next Adj./Val. [Index] ⑱

Figure 41: Menu “Adjustment / validation” – Edit (example)

① Index number of adjustment or validation procedure
② Name of this procedure (freely selectable text)

Settings for automatic starts of this procedure:

③ Time of first start of this procedure (format: hh:mm = hours:minutes)
④ Date of first start of this procedure (format: yy:mm:dd = year-month-day)
⑤ Time interval in which this procedure starts automatically (hours, days or weeks).
⑥ ☑ = starts are activated (time-controlled starts).
⑦ The fields under Start time and Start date show the next start.

Component X (up to 8 components can be managed by this procedure.):

⑧ Component tag
⑨ Component name in the Sensor module
⑩ Function selected for this component[1]
⑪ Flush time set in test gas settings (information)
⑫ Test gas name in test gas settings (information)
⑬ Sequence in procedure execution[2]
⑭ Test gas flush time for this adjustment/validation procedure[3]
⑮ Duration of measurements (seconds)[4]
⑯ Internal processing time (drift calculation, data storage)
⑰ Flush time with sample gas after the adjustment (seconds)[5]
⑱ As required: Index number of procedure to be started automatically after this procedure.[6]

[1] When the same function is selected for all components and the same test gas is planned, the function is performed within the procedure simultaneously for all components. Otherwise the functions are performed sequentially within the procedure.

[2] Functions with identical “execution pos.” run at the same time.

[3] When the same test gas is used for all functions: The longest individual test gas flush time (will be used automatically by the procedure). Otherwise: “0” (= individual test gas flush times are applicable).

[4] The mean value of the measured values during the measuring time is used as actual value for the measurement.

[5] Procedure status is first terminated after this flush time.

[6] Starts immediately after this procedure has terminated.

Layout of adjustment and validation results, see "Adjustment results", page 17.

10 Measured value configuration

10.1 Functions for measured values

10.1.1 Programming measured values (information)

All measured values displayed and output by the BCU are created by the assignment of a value to an internal measured value variable (MVi). The value is normally the measurement signal of a Sensor module (assignment: MVi = SiMVj).

Formulas can however also be programmed to define a measured value. This allows conversions, e.g. with constant factors or dynamic computations with other measured variables, e.g. with the pressure or an external measured value.

NOTE
These options are also available with “help values” (see "Configuring help values", page 55).

10.1.2 Displaying measured values

Hiding a measured value

Measured values not marked as “active” are neither displayed nor output.

Decimal places

The number of decimal places with which a measured value is displayed on the BCU can be set in a menu. This entry is also valid for the measuring screens in SOPAS ET.

NOTE

- Configuring measuring screens, see "Measuring screens in SOPAS ET", page 30
- Selecting a measuring screen, see "Measuring screen", page 13

10.1.3 Limit values

Two limit values can be programmed for each measured value. A message is activated when the measured value is beyond a limit value. The status flag to be activated can be set respectively.

10.1.4 Measured value mask

A measured value mask serves to “hide” measured values in a certain value range. The constant “hold value” is then output instead of a current measured value within the range set.

The measured value mask affects the analog output of the measured value, digital outputs and measured value displays.

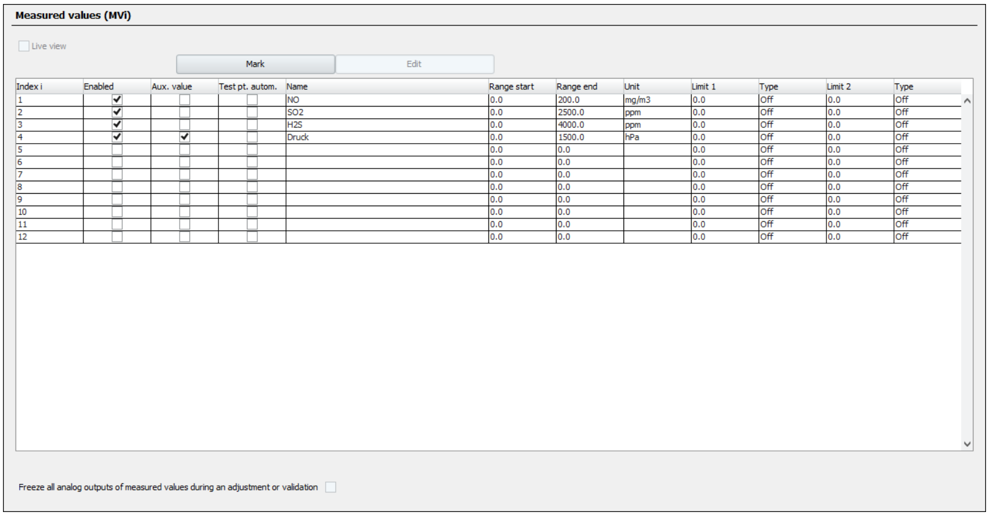
CAUTION
Risk of undesired effects with connected stations
The measured value displayed within the masked range does not normally correspond to the current measured value. All measuring screens display the current measured value again immediately as soon as the measured value leaves the masked range. This effect also occurs in the reverse direction.

▶ Consider the effect of the measured value mask when the measured value is used by external controllers.

10.2 Configuring measured values

Procedure

1. Call up menu: **BCU/Parameter/Measured values (MVi)**.



Measured values (MVi)

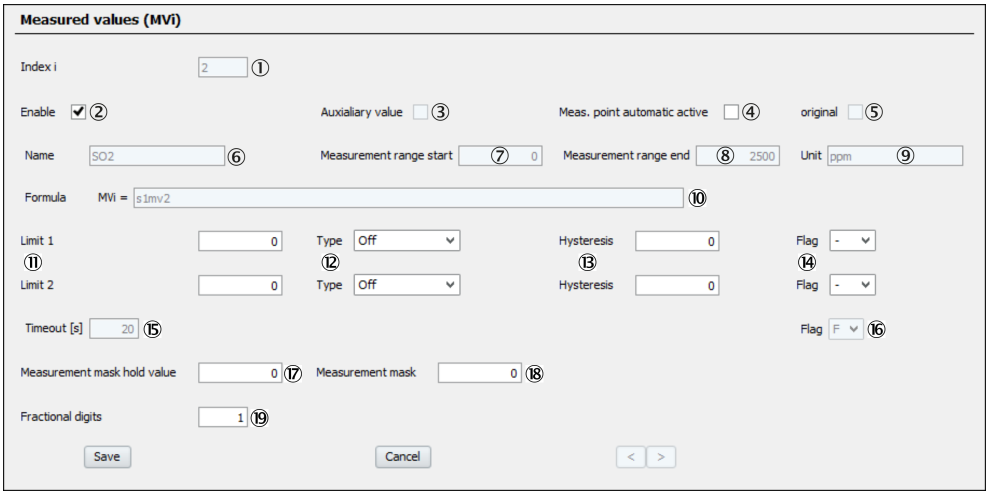
□ Live view

Index i	Enabled	Aux. value	Test pt. autom.	Name	Range start	Range end	Unit	Limit 1	Type	Limit 2	Type
1	☑	□	□	NO	0.0	200.0	mg/m3	0.0	Off	0.0	Off
2	☑	□	□	SO2	0.0	2500.0	ppm	0.0	Off	0.0	Off
3	☑	□	□	H2S	0.0	4000.0	ppm	0.0	Off	0.0	Off
4	☑	☑	□	Druck	0.0	1500.0	hPa	0.0	Off	0.0	Off
5	□	□	□		0.0	0.0		0.0	Off	0.0	Off
6	□	□	□		0.0	0.0		0.0	Off	0.0	Off
7	□	□	□		0.0	0.0		0.0	Off	0.0	Off
8	□	□	□		0.0	0.0		0.0	Off	0.0	Off
9	□	□	□		0.0	0.0		0.0	Off	0.0	Off
10	□	□	□		0.0	0.0		0.0	Off	0.0	Off
11	□	□	□		0.0	0.0		0.0	Off	0.0	Off
12	□	□	□		0.0	0.0		0.0	Off	0.0	Off

Freeze all analog outputs of measured values during an adjustment or validation □

Figure 42: Menu “Measured values (MVi)” – Table (example)

2. Deactivate **Live View**.
3. Mark desired Table rows.
4. Select **Edit**.



Measured values (MVi)

Index i: 2 ①

Enable ☑ ② Auxiliary value □ ③ Meas. point automatic active □ ④ original □ ⑤

Name: SO2 ⑥ Measurement range start: ⑦ 0 Measurement range end: ⑧ 2500 Unit: ppm ⑨

Formula MVi = s1mv2 ⑩

Limit 1: 0 Type: Off Hysteresis: 0 Flag: -

⑪ ⑫ ⑬ ⑭

Limit 2: 0 Type: Off Hysteresis: 0 Flag: -

Timeout [s]: 20 ⑮ Flag: F ⑯

Measurement mask hold value: 0 ⑰ Measurement mask: 0 ⑱

Fractional digits: 1 ⑲

Figure 43: Menu “Measured values (MVi)” – Edit (example)

①	Consecutive number (1 = BV01, 2 = BV02 etc.)
②	☑ = measured value displayed and output
③	☑ = measured value is not shown on the display (all other usage options remain available)
④	☑ = an own “measured value” is also created additionally for each measuring point[1]
⑤	The measured value configuration relative to name, unit and measured value follow the source (sensor measured value)
⑥	Programmed name of measured value
⑦	Start value of physical measuring range
⑧	End value of physical measuring range
⑨	Programmed physical unit for measured value

⑩ Formula for assigning or calculating measured value[2]

⑪ Limit value[3]

⑫
- Off: Limit value not active
- Overflow (+): Limit value message is active when the measured value is greater than the limit value
- Underflow (+): Limit value message is active when the measured value is less than the limit value

⑬ Hysteresis[3] [4]

⑭ This flag is activated for the measured value when the measured value is beyond the limit value ("–" in flag = no flag activation).

⑮ Programmed time limit value for an internal failure of the measurement signal or the source value for this measured value.

⑯ Activated flag when time limit value overflown.

⑰ Start value[3] for using the measured value mask. Also serves as output value in the measurement value mask range.[5]

⑱ Effective range[3] of the measured value mask. Valid as from the start value; positive or negative value possible.

⑲ Number of decimal places.

[1] Explanation, see "Function of the measuring point automatic", page 53

[2] Explanation, see "Functions for measured values", page 51

[3] In the physical unit.

[4] Determines the switching point at which the limit value message is deactivated again. The effective direction is automatically correct for "overflow" and "underflow".

[5] Explanation, see "Measured value mask", page 51

5. Enter desired data.
6. Select **Save**.

10.3 Measuring point automatic

10.3.1 Function of the measuring point automatic

Measuring point switching

Measuring points are extraction points for sample gas. Using the "measuring point automatic", the BCU can control up to eight measuring points automatically (prerequisites, see "Prerequisites for the measuring point automatic", page 54).

Hold functions for analog outputs

When the measuring point automatic is activated for a measured value (see "Configuring measured values", page 52), internal additional measured values of the measuring points (MViMPj) also exists apart from the measured value (MVi). These measured values of the measuring points can be output via analog outputs. During the measuring duration of the measuring point (see "Configuring the measuring point automatic", page 54), the current value measured by the GMS800 is output as measured value. The last measured value measured with this measuring point is output constantly during the remaining time.

NOTE
An identifier of the current measuring point is shown in the measuring screens and on the BCU display when measuring point automatic is active.

10.3.2 Prerequisites for the measuring point automatic

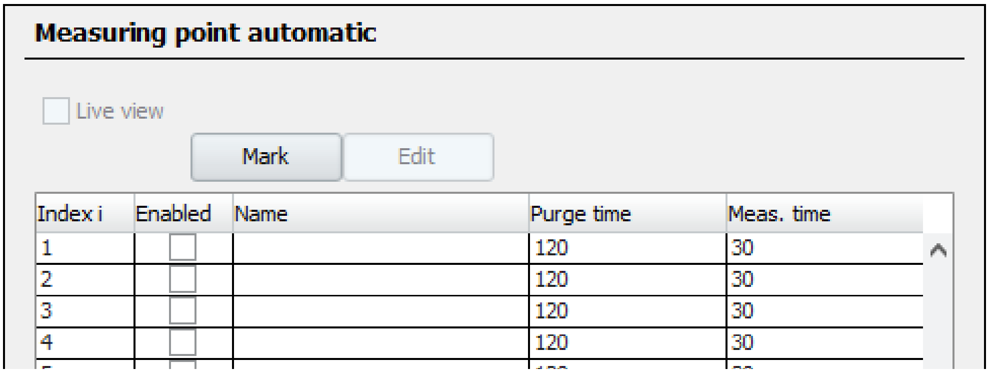
- The GMS800 has a digital output for each measuring point. The digital output is configured for the measuring point (see "Configuring digital outputs (DOi)", page 38).
- A mechanism is installed outside the GMS800 that switches the sample gas path for the measuring point (e.g. a solenoid valve). The associated digital output controls this mechanism.
- At least two measuring points are configured and switched to “active” (see "Configuring the measuring point automatic", page 54).

NOTE
Digital outputs for measuring point automatic are controlled with tag MPiS (see "Configuring digital outputs (DOi)", page 38).

10.3.3 Configuring the measuring point automatic

Procedure

1. Call up menu: **BCU/Parameter/Measuring point automatic**”.



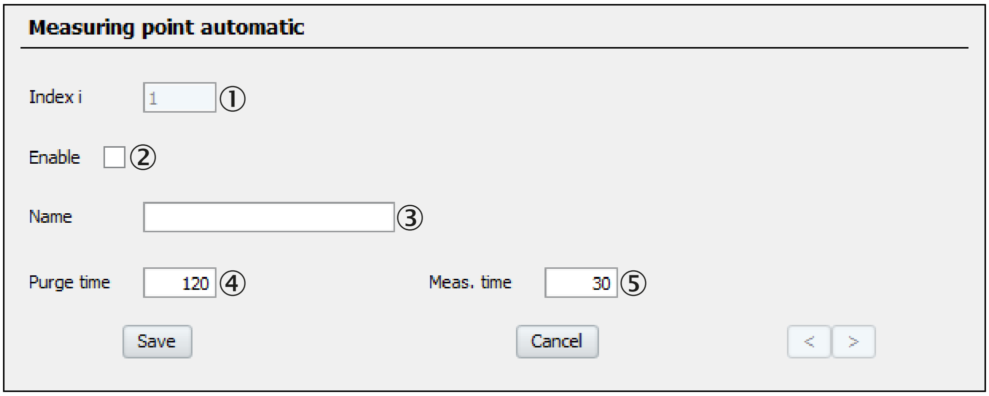
Measuring point automatic

□ Live view

Index i	Enabled	Name	Purge time	Meas. time
1	□		120	30
2	□		120	30
3	□		120	30
4	□		120	30

Figure 44: Menu “Measuring point automatic” – Table (example)

2. Deactivate **Live View**.
3. Mark desired Table rows.
4. Select **Edit**.



Measuring point automatic

Index i: 1 ①

Enable □ ②

Name: ③

Purge time: 120 ④ Meas. time: 30 ⑤

Figure 45: Menu “Measuring point automatic” – Edit (example)

① Consecutive number (1 = MP1, 2 = MP2 etc.)[1]

② ☑ = this measuring point is used by the measuring point automatic

③ Measuring point name

④ Wait time after switching to this measuring point[2]

⑤ Measuring time with sample gas from this measuring point[3] [4]

[1] Determines measuring point sequence during switching.
[2] Criterion: Response time + $T_{90\,\%}$.
[3] Select as required.
[4] Flush time + measuring time = activation time for the digital output = total time for this measuring point.

5. Enter desired data.
6. Select **Save**.

10.4 Configuring help values

Function

Some Sensor modules need measured values from other sources for their measuring function, e.g. the current pressure. Such measured values are “help values”.

The menu for a help value defines:

- Measured value source
- Target to which the help value is sent internally.

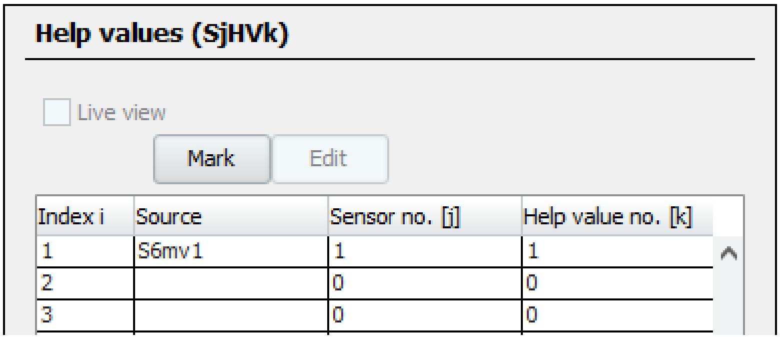
Requirements

The following must be known to configure the help values:

- Which fitted Sensor module needs which help values
- Tags of the respective measured values of the Sensor module
- The associated help value number in the Sensor module
- Help value source tag.

Procedure

1. Call up menu: **BCU/Parameter/Variables/Help values (SjHVk).**



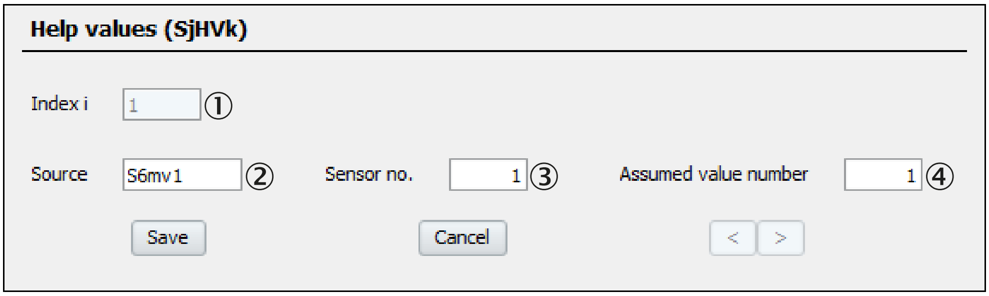
Help values (SjHVk)

☐ Live view

Index i	Source	Sensor no. [j]	Help value no. [k]
1	S6mv1	1	1
2		0	0
3		0	0

Figure 46: Menu “Help values (SjHVk)” – Table (example)

2. Deactivate **Live View.**
3. Mark desired Table rows.
4. Select **Edit.**



Help values (SjHVk)

Index i: 1 ①

Source: S6mv1 ② Sensor no.: 1 ③ Assumed value number: 1 ④

Figure 47: Menu “Help values (SjHVk)” – Edit (example)

① Consecutive number (1 = HV1, 2 = HV2 etc.)
② Internal identifier (tag) of the help value source[1]
③ Internal identifier of the Sensor module
④ Help value identifier in the Sensor module

[1] Explanation, see "Functions for measured values", page 51.

5. Enter desired data.
6. Select **Save**.

11 Modbus functions

11.1 Introduction to the Modbus protocol

Function

Modbus is a communication standard for digital controls to create a connection between a “Master” device and several “Slave” devices. The Modbus protocol only defines the communication commands, not their electronic transfer. It can therefore be used with differing digital interfaces.

Command structure

Device address (address)	Function code (function)	Function data (data)	Checksum (check sum)

- The device address is set individually for each connected device.
- Function codes are specified by the Modbus standard. For example, they order the slave to output device data (Read) or to change internal states (Force).
- The function data contain the additional information concerning the function code. This information is device-specific, i.e. it has to be defined by the manufacturer. Function code + function data form the command to be performed by the slave.
- The checksum is used to validate the transmitted data. It is automatically calculated by sender and receiver. If the results are identical, data transmission was correct.

Slave device response

Normally, the slave will respond to a command by sending an echo with the same function code and the function data containing the requested information. For error messages, the function code is modified and the function data contain the error code.

NOTE
Further information on the Modbus protocol: http://www.modbus.org.

11.2 Modbus specifications with the BCU Basic Control Unit

11.2.1 Installing the Modbus connection

Operating modes / interfaces supported

The Modbus can be operated

- in TCP mode via the Ethernet interface
- in RTU mode via the RS485 interface.

Interface	Parameter	Specification
LAN/Ethernet (TCP)	Port:	Adjustable[1]
RS485	Mode:	RTU
	Baud rate:	9600 or 19200
	Parity:	Even, uneven or none

[1] Via SOPAS ET or via the display (→ see “BCU Basic Control Unit” Operating Instructions).

Procedure

1. Check which Modbus mode is set (see "Checking Modbus activation", page 29).

2 When TCP is set:

▷ Connect the external Modbus system to the Ethernet interface[1] (see "Creating a network connection", page 9).

2 When RTU is set:

▷ Connect the external Modbus system to the RS485 interface[1].

▷ Match the interface parameters on all connected devices.

[1] Position → see Additional Operating Instructions of the enclosure.

11.2.2 Modbus parameters

- The GMS800 with the BCU runs as Slave.
- Maximum 5 Modbus commands per second are allowed (Modbus interval > 200 ms).
- 3 different addressing options are available:
  1. Standardized register assignment for Modbus outputs (already programmed) (see "Standardized register assignment for Modbus outputs", page 59)
  2. Modbus registers that can be set individually for specific information (see "Modbus outputs, configurable individually", page 61 as well as see "Modbus inputs, configurable individually", page 65)
  3. Interface in accordance with VDI 4201-3 (see "Assignment for communication in accordance with VDI 4201-3", page 69)
- The Modbus parameters can be configured in user level “Service” (see "Checking Modbus activation", page 29)

NOTE
The complete Address Table can be obtained from the product management.

11.2.3 Data formats and Modbus identification

Type	Description
String	Length: 32 bytes; Storage: In 16 registers. All registers receive a register address so that they can be accessed with the read and write functions for Holding registers.
Float	Length: 32 bits; Storage: In 2 registers (IEEE 754). Register n: SEEEEEEE EMMMMMMM, Register n+1: MMMMMMM MMMMMMM (S = Sign-Bit, E = Exponent, M = Mantissa)
USShort	Length: 16 bits. Storage: In 1 register.
USLong	Length: 32 bits. Storage: In 2 registers. Both registers receive a register address so that they can be accessed with the read and write functions for registers.
Boolean	Length: 1 bit. The group of 5 status bits belonging to the measured value are stored in a register. Each status bit receives a coil address to enable accessing via the read and write functions for coils. In addition, the common register receives a register address. This allows reading out and writing measured values together with status bits in one block with the read and write functions for Holding registers.

Modbus identification

Register address (holding and input)	Type	Function	Access[1]
0001 ... 0016	String	Sensor identification	R
0017 ... 0032	String	Software identification	R
0033 ... 0048	String	Device identification	R

[1] R = Read (value can be retrieved via Modbus).
W = Write (value can be changed via Modbus).

Modbus control word

Register address (holding and input)	Coil address	Type	Function	Access[1]
1369 ... 1370	16444 ... 16475	USLong / Boolean	Control word	R/W

[1] R = Read (value can be retrieved via Modbus).
W = Write (value can be changed via Modbus, only for Holding registers).

11.3 Function codes supported

Code	Designation	Function
0x01	Read Coils Status	Read binary values
0x03	Read Holding Registers	Read 16 bit values
0x04	Read Input Registers	Read 16 bit values
0x2B	Read Device Identification	Request for software version, serial number, device name and measuring components including display range end value from the Master to the Slave
0x08	Diagnostics[1]	0x00: Echo
		0x0A: Reset all counters
		0x0C: Number of CRC errors
		0x0D: Number of error replies
		0x0E: Number of inquiries
0x0F	Write multiple Coils	Write binary values
0x10	Write multiple Registers	Write 16 bit values

[1] Only when connected via an RS485 interface

11.4 Modbus addresses

11.4.1 Standardized register assignment for Modbus outputs

Information on individual measuring components

Register assignment in the Table page 60 has already been implemented in all GMS800 devices as from software version 4.0. The measured value sequence depends on the GMS800 configuration.

“Function code 0x04 - Read Input Registers” is used in the standardized register assignment.

NOTE

- The Modbus function must be activated (see "Checking Modbus activation", page 29).

Output	Register address (input)	Type	Function	Remark	Access[1]
Measured value 1	5000 … 5001	Float	Measured value	Measured values	R
	5002	USShort	Status	Status[2]	R
	5003	USShort	Free		R
	5004 … 5005	Float	Zero point value	Measured value of zero point adjustment	R
	5006 … 5007	Float	Reference point value	Measured value of reference point adjustment	R
	5008 … 5009	Float	Measuring range start		R
	5010 … 5011	Float	Measuring range end		R
Measured value 2	5050 … 5051	Float	Measured value		R
	5052	USShort	Status		R
	5053	USShort	Free		R
	5054 … 5055	Float	Zero point value		R
	5056 … 5057	Float	Reference point value		R
	5058 … 5059	Float	Measuring range start		R
	5060 … 5061	Float	Measuring range end		R

etc.. ↓

Measured value 12	5550 … 5551	Float	Measured value		R
	5552	USShort	Status		R
	5553	USShort	Free		R
	5554 … 5555	Float	Zero point value		R
	5556 … 5557	Float	Reference point value		R
	5558 … 5559	Float	Measuring range start		R
	5560 … 5561	Float	Measuring range end		R

[1] R = Read (value can be retrieved via Modbus).
W = Write (value can be modified via Modbus).

[2] Status: Bit field, see "Status information for individual measured values", page 60

Status information for individual measured values

The associated status information is made available for each measured value in register 5xx2, decoded in single bits.

Bit No.	Function	Remark
0	Failure	Bit = 1: active
1	Maintenance request	Bit = 1: active
2	Function check	Bit = 1: active
3	Uncertain	Bit = 1: active
4	Extended information	Bit = 1: active
5	Measuring range underflown (by more than 10%)	Bit = 1: active
6	Measuring range overflown (by more than 10%)	Bit = 1: active
7	Maintenance mode	Bit = 1: active
8	Free	Bit = 1: active

Bit No.	Function	Remark
9	Limit value	Bit = 1: active
10	Free	Bit = 1: active
11	UV lamp intensity alarm	Bit = 1: active, only for DEFOR
12	Adjustment	Bit = 1: active
13	Validation	Bit = 1: active
14	Free	Bit = 1: active
15	Measuring component activated	Bit = 1: active

Information on complete device

Information on the complete analyzer is shown in address are 6000 ... 6015.

Output	Register address (Input)	Type	Function	Remark	Access
Common OUT	6000	USShort	Year of current date	> 2000	R
	6001	USShort	Month of current date	1 - 12	R
	6002	USShort	Day of current month	1 - 31	R
	6003	USShort	Hour of current time	0 - 23	R
	6004	USShort	Minute of current time	0 - 59	R
	6005	USShort	Second of current time	0 - 59	R
	6006 ... 6007	USLong	Failure [complete]	Bit Field[1]	R
	6008 ... 6009	USLong	Maintenance necessary [complete]	Bit Field[1]	R
	6010 ... 6011	USLong	Function check [complete]	Bit Field[1]	R
	6012 ... 6013	USLong	Uncertain [complete]	Bit Field[1]	R
	6014 ... 6015	USLong	Extended information [complete]	Bit Field[1]	R

[1] These data fields/register addresses contain additional information. This allows capturing to which Measuring module the message is to be assigned (see following Table).

Bit No.	Function	Remark
0	BCU (System)	Bit = 1: active, collective signal
1	Sensor/Module 1	Bit = 1: active, collective signal
2	Sensor/Module 2	Bit = 1: active, collective signal
3	Sensor/Module 3	Bit = 1: active, collective signal
4	Sensor/Module 4	Bit = 1: active, collective signal
5	Sensor/Module 5	Bit = 1: active, collective signal
6	Sensor/Module 6	Bit = 1: active, collective signal

11.4.2 Modbus outputs, configurable individually

Individual information can be communicated via Modbus using a further address space.

NOTE
This function must be configured in SOPAS ET.

- Information on the Modbus inputs that can be configured individually (see "Modbus inputs, configurable individually", page 65)

“Function code 0x03 - Holding Registers” is used in the individually configurable register assignment.

11.4.2.1 Addresses of individual Modbus outputs

Output	Register address (holding and input)	Coil address	Type	Function	Access[1]
MB01	67 ... 68	–	USLong	Integer value	R
	69 ... 70	–	Float	Floating-point value	R
	71 (Bit 0)	262	Boolean	Failure (F)	R
	71 (Bit 1)	263	Boolean	Maintenance request (M)	R
	71 (Bit 2)	264	Boolean	Function check (C)	R
	71 (Bit 3)	265	Boolean	Uncertain (U)	R
	71 (Bit 4)	266	Boolean	Extended information (E)	R
MB02	88 ... 89	–	USLong	Integer value	R
	90 ... 91	–	Float	Floating-point value	R
	92 (Bit 0)	523	Boolean	Failure (F)	R
	92 (Bit 1)	524	Boolean	Maintenance request (M)	R
	92 (Bit 2)	525	Boolean	Function check (C)	R
	92 (Bit 3)	526	Boolean	Uncertain (U)	R
	92 (Bit 4)	527	Boolean	Extended information (E)	R

etc.

MB024	550 ... 551	–	USLong	Integer value	R
	552 ... 553	–	Float	Floating-point value	R
	554 (Bit 0)	6265	Boolean	Failure (F)	R
	554 (Bit 1)	6266	Boolean	Maintenance request (M)	R
	554 (Bit 2)	6267	Boolean	Function check (C)	R
	554 (Bit 3)	6268	Boolean	Uncertain (U)	R
	554 (Bit 4)	6269	Boolean	Extended information (E)	R

[1] R = Read (value can be retrieved via Modbus).
W = Write (value can be changed via Modbus).

11.4.2.2 Configuring Modbus outputs

Function

The data the GMS800 sends via the Modbus interface after an inquiry command can be programmed. Here, the outputs (Modbus outputs, MBO) are assigned the desired variables and values. Available are:

- The measured values
- All available variables, see "Defining variables", page 85

24 data outputs can be configured.

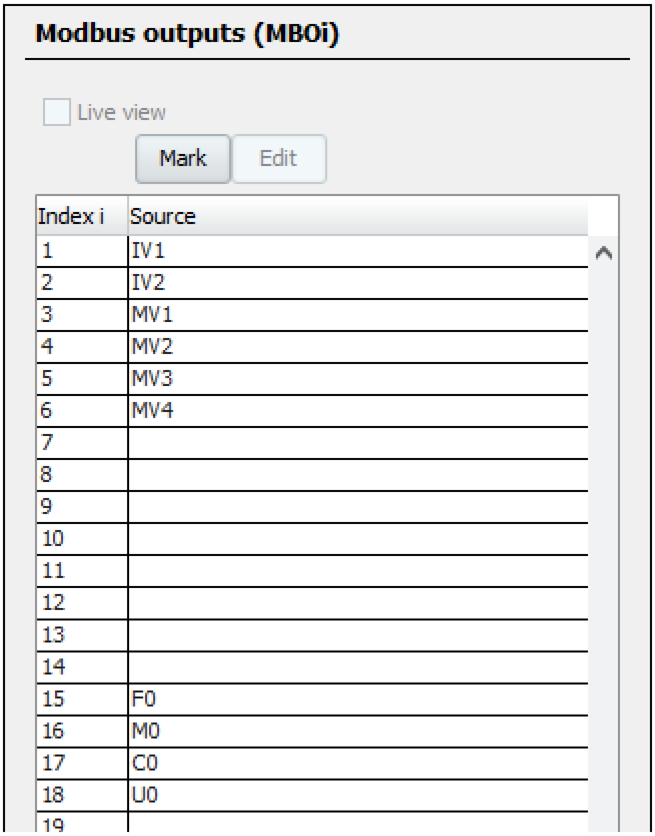
NOTE

- The Modbus function must be activated (see "Checking Modbus activation", page 29).
- Technical information on the Modbus interface, see "Modbus functions", page 57.

The following Figure (see figure 48, page 63) shows the assignment of the outputs preprogrammed at the factory. This configuration can be adapted to individual customer requirements.

Procedure

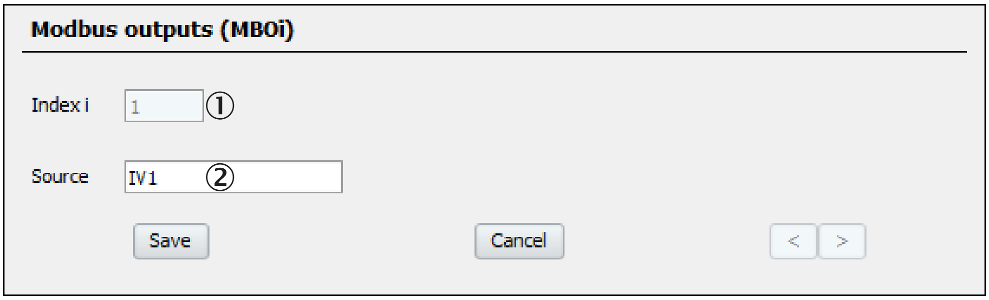
1. Call up menu: **BCU/Parameter/I/O/Modbus outputs** (MBOi).



Index i	Source
1	IV1
2	IV2
3	MV1
4	MV2
5	MV3
6	MV4
7	
8	
9	
10	
11	
12	
13	
14	
15	F0
16	M0
17	C0
18	U0
19	

Figure 48: Menu “Modbus outputs” – Table (example)

2. Deactivate **Live View.**
3. Mark desired Table rows.
4. Select **Edit.**



Index i: 1 ①

Source: IV1 ②

Figure 49: Menu “Modbus outputs” – Edit (example)

① Modbus output number
② Possible sources, see context help in SOPAS ET

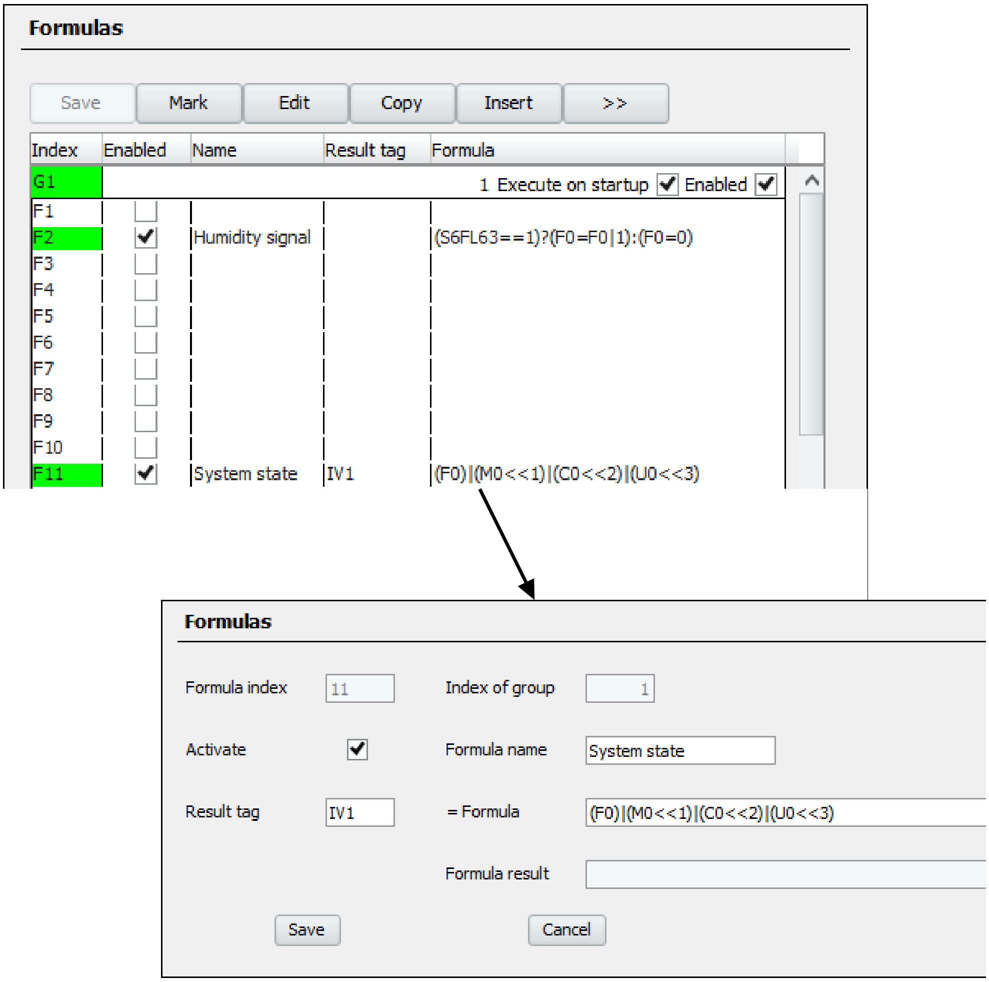
5. Enter desired source code.
6. Select **Save.**

The measured values (MVi) are assigned directly to the outputs. The status of a measured value can also be recorded using the Read Coils.

Example

The **status of the complete device** is to be shown in a variable.

1. Call up menu: **BCU/Parameter/Formulas/Formulas** (see "Menu functions for formulas", page 81)
2. Define variable: IV1= (F0)|(M0<<1)|(C0<<2)|(U0<<3)

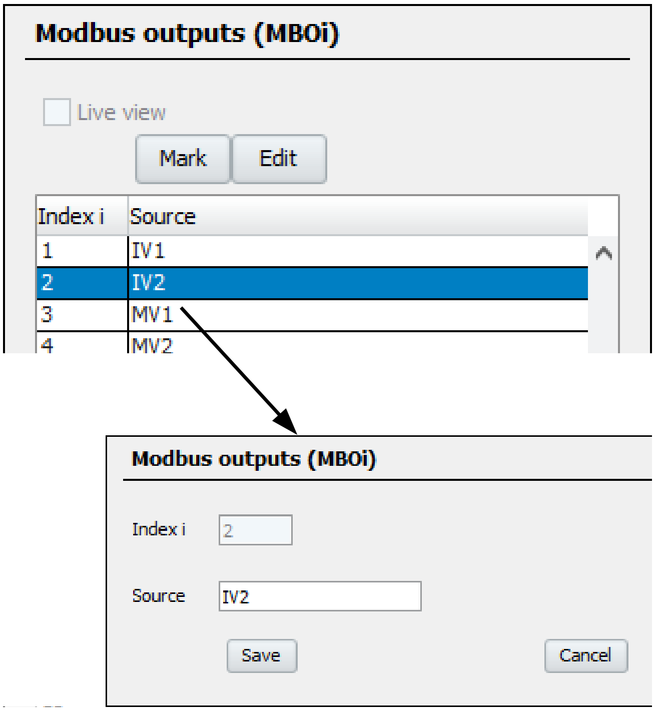


Index	Enabled	Name	Result tag	Formula
G1				1 Execute on startup ☑ Enabled ☑
F1	□			
F2	☑	Humidity signal		(S6FL63==1)?(F0=F0|1):(F0=0)
F3	□			
F4	□			
F5	□			
F6	□			
F7	□			
F8	□			
F9	□			
F10	□			
F11	☑	System state	IV1	(F0)|(M0<<1)|(C0<<2)|(U0<<3)

Formula index	11	Index of group	1
Activate	☑	Formula name	System state
Result tag	IV1	= Formula	(F0)|(M0<<1)|(C0<<2)|(U0<<3)
		Formula result	

Figure 50: Example: Configure Modbus output (Part 1)

3. Call up menu: **BCU/Parameter/I/O/Modbus outputs (MBOi)**
4. Assign defined variable to Modbus output: MBO1 = IV1



Index i	Source
1	IV1
2	IV2
3	MV1
4	MV2

Index i: 2

Source: IV2

Figure 51: Example: Configure Modbus output (Part 2)

11.4.3 Modbus inputs, configurable individually

Individual information can be communicated via Modbus using a further address space.

NOTE
This function must be configured in SOPAS ET.

- Information on the Modbus outputs that can be configured individually (see "Modbus outputs, configurable individually", page 61)

11.4.3.1 Addresses of the individual Modbus inputs

Function

Additional register addresses are available to enter external signals.

NOTE
After the information has been assigned to the register, an internal identifier should be defined for the documentation (see "Identifying Modbus inputs", page 66).

Input	Register address (holding and input)	Coil address	Type	Function	Access[1]
MBI1	1371 ... 1372	-	Float	Floating-point value	R/W
	1373 (Bit 0)	16476	Boolean	Failure (F)	R/W
	1373 (Bit 1)	16477	Boolean	Maintenance request (M)	R/W
	1373 (Bit 2)	16478	Boolean	Function check (C)	R/W
	1373 (Bit 3)	16479	Boolean	Uncertain (U)	R/W
	1373 (Bit 4)	16480	Boolean	Extended information (E)	R/W

Input	Register address (holding and input)	Coil address	Type	Function	Access[1]
MBI2	1374 ... 1375	-	Float	Floating-point value	R/W
	1376 (Bit 0)	16481	Boolean	Failure (F)	R/W
	1376 (Bit 1)	16482	Boolean	Maintenance request (M)	R/W
	1376 (Bit 2)	16483	Boolean	Function check (C)	R/W
	1376 (Bit 3)	16484	Boolean	Uncertain (U)	R/W
	1376 (Bit 4)	16485	Boolean	Extended information (E)	R/W

etc.

MBI24	1440 ... 1441	-	Float	Floating-point value	R/W
	1442 (Bit 0)	16591	Boolean	Failure (F)	R/W
	1442 (Bit 1)	16592	Boolean	Maintenance request (M)	R/W
	1442 (Bit 2)	16593	Boolean	Function check (C)	R/W
	1442 (Bit 3)	16594	Boolean	Uncertain (U)	R/W
	1442 (Bit 4)	16595	Boolean	Extended information (E)	R/W

[1] R = Read (value can be retrieved via Modbus) W = Write (value can be modified via Modbus), only for Holding registers.

11.4.3.2 Identifying Modbus inputs

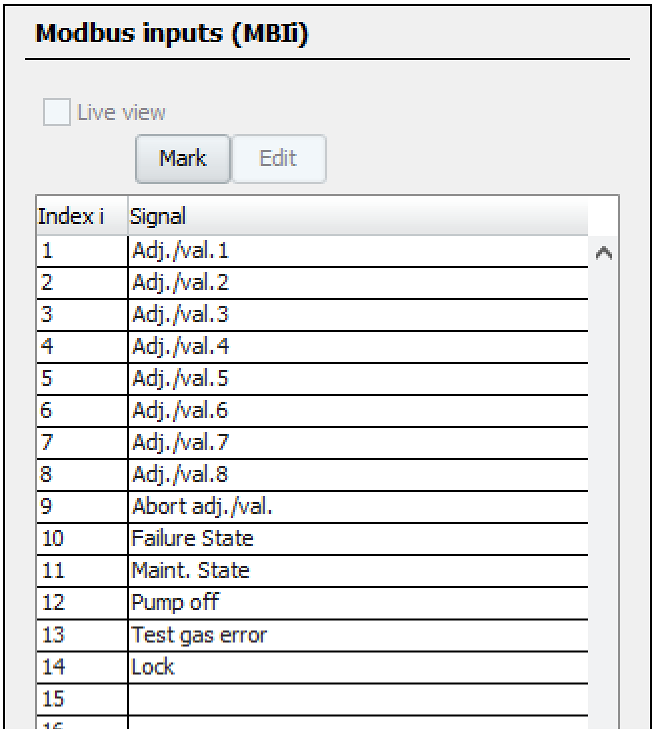
Function

An individual identifier can be entered for each Modbus input.

NOTE
Consider the send procedure with the function code according to chapter 11.3

Procedure

1. Call up menu: **BCU/Parameter/I/O/Modbus inputs** (MBIi).



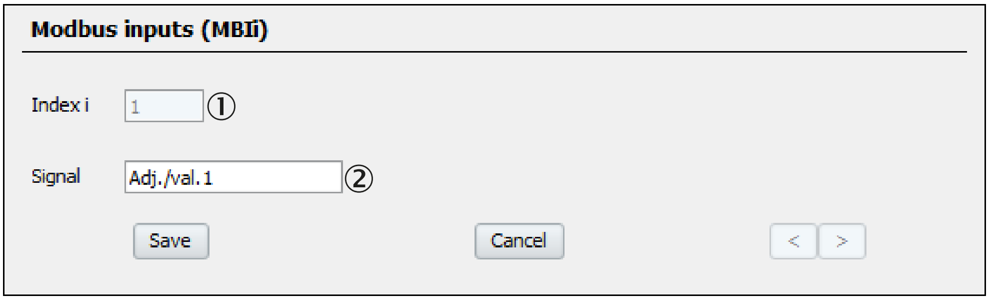
Modbus inputs (MBIi)

□ Live view

Index i	Signal
1	Adj./val.1
2	Adj./val.2
3	Adj./val.3
4	Adj./val.4
5	Adj./val.5
6	Adj./val.6
7	Adj./val.7
8	Adj./val.8
9	Abort adj./val.
10	Failure State
11	Maint. State
12	Pump off
13	Test gas error
14	Lock
15	
16	

Figure 52: Menu “Modbus inputs” – Table (example)

2. Deactivate **Live View**.
3. Mark desired Table rows.
4. Select **Edit**.
5. As required: Enter text for signal.



Modbus inputs (MBIi)

Index i: 1 ①

Signal: Adj./val.1 ②

Figure 53: Menu “Modbus inputs” – Edit (example)

①	Modbus input number
②	User identifier for this Modbus input[1]

[1] Any text. Not used internally.

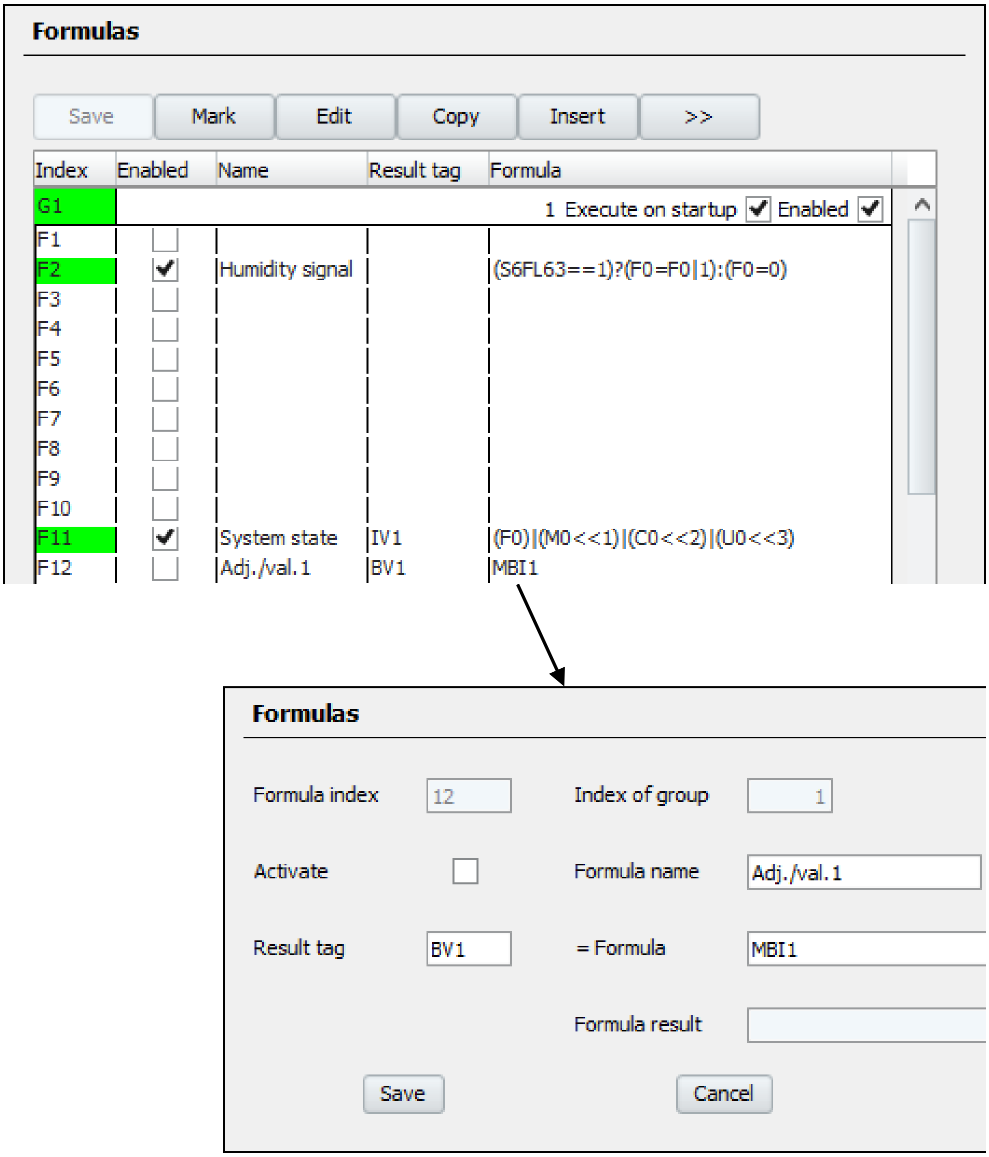
The signal processing of the Modbus inputs (MBIi) within the Analyzer must then be configured.

Example

The first adjustment function within the device is to be configured with the Modbus input (MBIi).

The input signals via the Modbus (Modbus inputs, MBIi) must be assigned the Boolean values (BVi) (see "Formulas", page 80). This is normally done by the Service technician during start-up.

1. Call up menu: **BCU/Parameter/Formulas/Formulas** (see "Menu functions for formulas", page 81)
2. Define variable and assign to the Modbus output: BV1 = MBI1

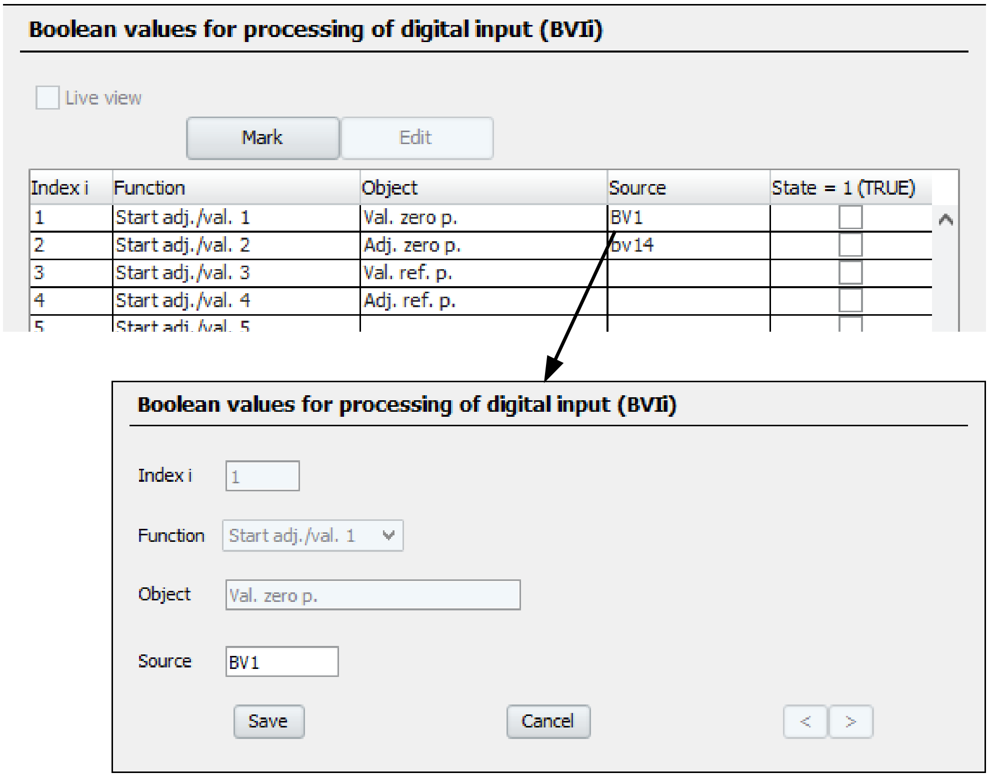


Index	Enabled	Name	Result tag	Formula
G1	1 Execute on startup ☑ Enabled ☑			
F1	□			
F2	☑	Humidity signal		(S6FL63==1)?(F0=F0|1):(F0=0)
F3	□			
F4	□			
F5	□			
F6	□			
F7	□			
F8	□			
F9	□			
F10	□			
F11	☑	System state	IV1	(F0)|(M0<<1)|(C0<<2)|(U0<<3)
F12	□	Adj./val.1	BV1	MBI1

Formula index: 12
Index of group: 1
Activate: □
Formula name: Adj./val.1
Result tag: BV1
= Formula: MBI1
Formula result:

Figure 54: Example: Configure Modbus inputs (Part 1)

3. Call up menu: **BCU/Parameter/Variables/Boolean values - input signals**
4. Select defined variable BVI as source and assign the Boolean values - Input signal: BVI1 = BV1



Boolean values for processing of digital input (BVIi)

Index i	Function	Object	Source	State = 1 (TRUE)
1	Start adj./val. 1	Val. zero p.	BV1	□
2	Start adj./val. 2	Adj. zero p.	bv14	□
3	Start adj./val. 3	Val. ref. p.		□
4	Start adj./val. 4	Adj. ref. p.		□
5	Start adj./val. 5			□

Boolean values for processing of digital input (BVIi)

Index i: 1

Function: Start adj./val. 1

Object: Val. zero p.

Source: BV1

Figure 55: Example: Configure Modbus input (Part 2)

figure 55 shows as an example how “adjustment/validation 1” is started by the Modbus input, the source for BVI1 is BV1 (MBI1).

5. To start adjustment function 1 via Modbus input 1 (MBI1), write the value 1.0 (MBI1 = 1) in the register of the Modbus input 1 (1371 ... 1372).

11.4.4 Assignment for communication in accordance with VDI 4201-3

VDI 4201-3 describes the minimum requirements on automatic measuring and electronic evaluate equipment for monitoring emissions within the framework of using a digital interface with the Modbus protocol.

The GMS800 supports the required function code 43 and provides the information in accordance with the Modbus specification (http://www.modbus-ida.org).

Function code	Function	Significance
0x2B (43)	Read Device Identification	Request for software version, serial number, device name and measuring components including display range end value from the Master to the Slave

Name	Register count	Object ID	Coding	Explanation	Contents example
Vendor Name	16	0x00	String32	Manufacturer name	EHS
ProductCode	16	0x01	String32	Manufacturer-specific device identifier	1234567

Table 7: Address assignment of device parameters for function code 43

Name	Register count	Object ID	Coding	Explanation	Contents example
MajorMinorRevision	16	0x02	String32	Software version	9342231_V3.007
ProductName	16	0x04	String32	Device identifier	GMS810
SerialNumber	16	0x80	String32	Serial number	1434005
ComponentNumber	1	0x81	16 Bit integer	Number of measured variables	8
BasisM	1	0x82	16 Bit integer	Base address of measuring block	4.000
BasisS	1	0x83	16 Bit integer	Base address of simulation data	4.200
BasisR	1	0x84	16 Bit integer	Base address of reference material data	4.400
Component1_Name	16	0x85	String32	Identifier of measuring component 1 1_Serial number_Sensor address (1-6)_Sensor type_Software version	CO2 12345678_1_9342231_3007
Component1_Range_Start	2	0x86	32 Bit float	Physical measuring range start 1[1]	-
Component1_Range_End	2	0x87	32 Bit float	Physical measuring range end 1[1]	500
Component1_Unit	16	0x88	String32	Unit of measuring component 1	mg/m3
Component2_Name	16	0x89	String32	Identifier of measuring component 2	SO2
Component2_Range_Start	2	0x90	32 Bit float	Physical measuring range start 2[1]	-
Component2_Range_End	2	0x91	32 Bit float	Physical measuring range end 2[1]	750
Component2_Unit	16	0x92	String32	Unit of measuring component 2	mg/m2
Component3_Name	16	0x93	String32	Identifier of measuring component 3	O2
etc.					

Table 7: Address assignment of device parameters for function code 43

[1] According to VDI 4201, the display range start or end of the measuring component should be output. The display range on the display of the Measuring module and the output range on the analog output can be freely configured and therefore the physical measuring range start or end is defined here.

Refer to the VDI 4201-3 standard for further information.

12 OPC

NOTICE
Observe the license conditions specified in the documentation provided.

- OPC (Openness Productivity Collaboration) is a standardized software interface that allows applications from differing manufacturers to exchange data.
- The SOPAS OPC server uses DCOM technology (Distributed Component Object Model) for communication between the applications. This allows the SOPAS OPC server to exchange data with a local process or also with a remote computer connected via Ethernet (TCP/IP).
- The OPC server collects the process data from the GMS800 and makes the data available as OPC objects.
- The OPC client accesses the data made available by the OPC Server and then processes the data further.

NOTE
Installing the OPC server and “first steps” → Online Help of the OPC servers

12.1 OPC interface

Item 2	Item 3	Item 4	Comment
Item 1: Device Login:			Login for a user level
Run			Login for this user level with the password for this user level
AuthorizedClient			Login for this user level with the password for this user level
Service			Login for this user level with the password for this user level
Item 1: Measuring_Device_1:			The physical analyzer system GMS800
General_Data			General data of the sensor / analyzer device. For complex systems it was defined by VDI4201 working group, that only the highest level of the system will be described.
	Device_Manufacturer		Manufacturer of the device
	Software_Revision		Software Version of the BCU, for example 9150883_4.000
	Device_Ser_Num		Serial number of the system
	Device_Name		
	Location		Entry parameter mounting position (configurable in SOPAS ET, page: parameter - device parameter)
		Failure_1	Status of the class Failure, for mapping see next item
		Check_1	Status of the class Check, for mapping see next item
		Out Of Spec_1	Status of the class Out Of Spec, for mapping see next item
		Maintenance Request_1	Status of the class Maintenance Request, for mapping see next item
		Extended_1	Status of the class Extended, for mapping see next item
	Maintenance	Initiate_Maintenance	
		Status_Maintenance	Source of / Reason for maintenance mode, for mapping see next item
		Description_Status_Maintenance	
	Initiate Restart		
	Time		
	Sample_Point_Description		Measuring point

Item 2	Item 3	Item 4	Comment
Measuring_Value	Component_1	Component_Name	Name of measuring component
		Range_Start	Start of measuring range; number refers to physical unit
		Range_End	End of measuring range; number refers to physical unit
		Unit	Unit of measuring component
		Qualifier	
		Sample_Point	Number of current measuring point [1..8], when this component is enabled, measuring point automatic is running and measuring point automatic enabled for this component, otherwise 0
		Value	Measuring signal scaled to [-10000; +10000], Zero at 0; refreshing rate : minimum once per second.
		Value_Unscaled_Live	Measuring value fitting to the physical unit (requested from Nexus). In usual measuring mode: same value as "Value_Unscaled_Hold"-item
		Status	Bit 00: Failure Bit 01: Check Bit 02: MaintenanceRequest Bit 03: OutOfSpec Bit 04: Simulation
			Status of the class Failure, for mapping see item Description_Failure_1
			Status of the class Check, for mapping see item Description_Check_1
			Status of the class Out Of Spec, for mapping see item Description_OutOfSpec_1
			Status of the class Maintenance Request, for mapping see item Description_MaintenanceRequest_1
			Status of the class Extended, for mapping see item Description_Extended_1
		Simulation_Value	Simulated measuring signal scaled to [-10000; +10000], Zero at 0; this value is only shown via OPC. On the OPC interface it only concerns the value item (not the value_unscaled)
		Simulation_Request	Allows switching between the submission of the actual measuring value and the simulated value:
		Control_Point_1	Validation zero point
			Absolute deviation in the given unit (s. "Unit" item below), Expected_Value_Out - Current_Value
			Actual value, measured value during adjustment/validation
			Format: yyyy-mm-dd hh:mm time stamp
			False = the values of the items of this control point are not valid True = the values of the items of this control point are valid
			Set value (expected value)
			Meaning of Control_Point
			Unit for "Expected_Value_In", "Expected_Value_Out", "Deviation" and "Current_Value"
			New set value (expected value), only written on changing of Input_Switch from False to True
			False = do nothing True = writes the new set value (expected value = Expected_Value_In) on changing of Input_Switch from False to True
		Control_Point_2	Adjustment zero point
		Control_Point_3	Validation reference point
		Control_Point_4	Adjustment reference point
		Cal_Type_Zero	False = zero point adjustment is generally not possible True = zero point adjustment is generally possible
		Cal_Type_Span	False = reference point adjustment is generally not possible True = reference point adjustment is generally possible
	Component 2-12 correspond to Component 1		
Reference			Test gases (only test gases, no cuvettes), only one test gas can be activated at same time
	Material_1		Test gas 1
		Status	Status of gas switching False = Measuring gas True = Reference material (test gas)
		Initiate	Demand for gas switching False = Measuring gas True = Reference material (test gas)
	Material 2-12 correspond to Material 1		Test gas 2

13 Specifications

13.1 Status flags

Flag	Significance		j[1]
F	Failure	Error (unsafe operating state)	1
M	Maintenance request	Maintenance request	2
C	Function check	Internal function sequence[2] (measuring function interrupted) / state “Maintenance“	3
U	Uncertain	Unsure state	4
E	Extended information	Extended message	
S	Group message	Group status for all associated F/M/C/U/E flags	

[1] In formula tags.
[2] e.g. adjustment procedure.

NOTE
Flag for limit value messages, see "Measured value configuration", page 51.

13.2 Task codes of adjustment and validation functions

Tasks (functions)

No.	Function	Reference variable	Point of the characteristic curve
1	Validation measurement	Test gas	Zero point
2	Adjustment		
(3)[1]	(Validation measurement)[1]	Adjustment unit[2]	
(4)[1]	(Adjustment)[1]		
5	Validation measurement	Test gas	Reference point
6	Adjustment		
7	Validation measurement	Adjustment unit[2]	
8	Adjustment		
9	Adjustment	Test gas	Linearity
0	None of the functions running[3]		
≥ 10	Abort/terminate the running function		

[1] These functions are not available for the present sensor.
[2] Only for Sensor modules with adjustment unit (option).
[3] Only relevant in a read tag.

Task States (sequence phases)

0	Idle
1	Flush
2	Measure
3	Calculate
4	Accept

13.3 Formula elements

13.3.1 Applicable values and states (read tags)

Values and states that can be used in formulas and measuring screens but can not be modified by formulas.

Exception: Variables identified with W = Write can also be written. (Remark: Variables only identified with R = Read can only be read out.)

Tag	Function	R/W[1]	Type[2]	Indices
Variables[3]				
BVi	Value of Boolean variable BVi (BVi=[term])	R/W	B	i = 1 ... 24
BVSi	Value of Boolean variable BVSi[7]	R	B	i = 1 ... 32
BVGi	Value of Boolean variable BVGi[4]	R	B	i = 1 ... 16
BVIi	Value of Boolean variable BVIi[5]	R	B	
BVOi	Value of Boolean variable BVOi[6]	R	B	
IVi	Integer value (IVi=[term])	R/W	I	i = 1 ... 24
RVi	Real value (RVi=[term])	R/W	R	
Process values (raw values) of sensor Si				

SiMVj	Measured value j	R	R	i = 1 ... 6
				j = 1 ... 10

SiMVjRL	Measuring range start of measured value j			
SiMVjRU	Measuring range end of measured value j			
BCU status				
XO	BCU group status for messages in category X	R/W	B	X =F/M/C/U[8]

[1] R = read (read/process) / W = write (write/modify, see "Programmable values and states (write tags)", page 75).
[2] R = floating point value (real), I = integer value (integer), B = Boolean value.
[3] Explanation, see "Defining variables", page 85.
[4] Corresponds to function status of relevant “operator settings” (see "Starting adjustment/validation (operator commands)", page 26).
[5] Explanation, see "Boolean variables (BVIi) – input signals", page 88.
[6] Explanation, see "Boolean variables (BVOi) – output signals", page 90.
[7] Explanation, see "Boolean variables (BVSi) - input signals (system)", page 86.
[8] Explanation, see "Status flags", page 73.

Tag	Function	R/W[1]	Type[2]	Indices
Input/Output				
AIi	Physical value [3] of analog input i	R	R	i = 1 ... 4
AIiI	Electronic value (0 ... 20 mA) of analog input i			
AOi	Physical value of analog output i[4]	R	R	i = 1 ... 8
AOiO	Electronic value (0 ... 20 mA) of analog output i			
AOiR	Current output range of analog output i[5]			
DIi	Logical status of digital input i (after inverse.[6])	R	B	i = 1 ... 16
DIiI	Electronic status of digital input i			
DOi	Logical state of digital output i (before inverse.[6])[4]			
DOiO	Electronic state of digital output i[4]			
MBIi	Value on Modbus input register	R	R	i = 1 ... 24
MBOi	Value on Modbus output register[4]	R		
Adjustments				

Tag	Function	R/W[1]	Type[2]	Indices

SiMVj-TAkTG	Test gas concentration (set point value)	R/W	R	i = 1 ...16[7]
				j = 1 ... 10[7]
				k = 1/2/5/6/9[7]

Status information				
MPiS	Activation state of measuring point i[9]	R	B	i = 1... 8
MPS	Operating state for measuring point automatic[8]	R	B	
MPiPPS	Activation state of measuring points advance extraction[10]	R	B	i = 1 ... 8
MPiM	Switch-on state/mode of measuring point[11]	R/W	B	i = 1 ... 8
Measuring screen				
MVi	Measured value i	R	R	i = 1 ... 12

MViMPj	Measured value i from measuring point j	R	R	i = 1 ... 12
				j = 1 ... 8

MViXO	(Overall) message status of measured value i	R	B	i = 1 ... 12
				X = F/M/C/U/E/S[12]

MViLIj	Limit value message for measured value i – limit value j	R	B	i = 1 ... 12
				j = 1 ... 2

MViRL	Measuring range start value of measured value i	R	R	i = 1 ... 12
MViRU	Measuring range end value of measured value i	R	R	i = 1 ... 12

[1] R = read (read/process) / W = write (write/modify, see "Programmable values and states (write tags)", page 75).
[2] R = floating point value (real), I = integer value (integer), B = Boolean value, T = (date &) time
[3] According to conversion programmed (see "Configuring analog inputs (AIi)", page 39).
[4] Only usable for measuring screens (see "Configuring measuring screen contents", page 31). Use the “source” instead in formulas.
[5] “0” = measuring range 1, “1” = measuring range 2.
[6] When logical inversion is activated.
[7] Description of indices according to this Table (see "Description of indices for adjustments/validations", page 75)
[8] “0” = flush, “1” = measure.
[9] “0” = not active, “1” = currently active.
[10] “0” = not active, “1” = currently active. (see "Measuring point automatic", page 53)
[11] “0” = switched off, “1” = switched on. (see "Measuring point automatic", page 53)
[12] Explanation, see "Status flags", page 73

Description of indices for adjustments/validations

i = index of automatic adjustment/validation

j = component (n-number) of automatic adjustment/validation i

k = (adjustment) function of component (n-number) of automatic adjustment/validation i (see "Tasks (functions)", page 73)

13.3.2 Programmable values and states (write tags)

Tag	Function	R/W[1]	Type[2]	Indices
Variables[3]				
BVi	Value of a Boolean variable (BVi = [term])	R/W	B	i = 1 ... 24
IVi	Integer value (IVi = [term])	R/W	I	
RVi	Real value (RVi=[term])	R/W	R	
Adjustment				

Tag	Function	R/W[1]	Type[2]	Indices

SiMVj-TAkTG=[Value]	Write test gas concentration (set point)
	[value] = concentration (floating point value)
		w	R	i = 1 ... 16[4]
				j = 1 ... 10[4]
				k = 1/2/5/6/9[4]

Measuring screen, sequence control programs, status information				
MPiM	Switch-on state/mode of measuring point[5]	R/W	B	i = 1 ... 8
BCU status				
XO	BCU group status for messages in category X	R/W	B	X =F/M/C/U[6]

[1] R = read (read/process) / W = write (write/modify).
[2] R = floating point value (real), I = integer value (integer), B = Boolean value.
[3] Explanation, see "Defining variables", page 85
[4] Description of indices, see "Description of indices for adjustments/validations", page 75
[5] “0” = switched off, “1” = switched on. (see "Measuring point automatic", page 53)
[6] Explanation, see "Status flags", page 73

13.3.3 Operators in formulas

Char-acters	Usage[1]	Value[2]	Function
Mathematical operators			
+	[op1]+[op2]	R, I	Addition
-	–[op]	R, I	• Value: Inverted sign • Operation: Subtraction
	[op1]–[op2]	R, I	
*	[op1]*[op2]	R, I	Multiplication
/	[op1]/[op2]	R, I	Division
**	[op1]**[op2]	R, I	Power
%	[op1]%[op2]	R, I	Modulo (rest from division of two integer values)
<	[op1]<[op2]	R, I, B	Less than
>	[op1]>[op2]	R, I, B	Greater than
<=	[op1]<=[op2]	R, I, B	Less than/equal to
>=	[op1]>=[op2]	R, I, B	Greater than/equal to
==	[op1]==[op2]	R, I, B	Equal to
!=	[op1]!=[op2]	R, I, B	Unequal
++	++[op]	R, I	[op] increment
––	––[op]	R, I	[op] decrement
( )		R, I	Combination of terms (see "Priority of formula operations", page 78)
Logical operators			
&&	[op1]&&[op2]	I, B	AND
||	[op1]||[op2]	I, B	OR
^^	[op1]^^[op2]	I, B	Exclusive OR (XOR)
!	![op]	B	NOT
Operators for Boolean values			
//	//[op]	B	[op] invert (toggle)
>+	>+[op]	B	• Boolean value: Activate [op] • Time variable:Start [op][3]

Char-acters	Usage[1]	Value[2]	Function
>–	>–[op]	B	• Boolean value: Activate [op] • Time variable: Stop [op] (value is retained)[3]
!+	!+[op]	B	Time variable: Restart [op] as from start value[3]
!–	!–[op]	B	Time variable: Reset [op] to start value[3]
~	~[op]	B	Build complement (invert single bits, e.g. 10 → 01)
Substitution operators (assignments)			
=	[op1]=[op2]	R, I, B	Assign [op1] the value or result from [op2]
	[op1]=[term1]		[op1] is the result from [term1]
+=	[op1]+=[op2]	R, I	Assign with addition ([op1] = [op1]+[op2])
-=	[op1]–=[op2]	R, I	Assign with subtraction ([op1] = [op1]–[op2])
*=	[op1]*=[op2]	R, I	Assign with multiplication ([op1] = [op1]*[op2])
/=	[op1]/=[op2]	R, I	Assign with division ([op1] = [op1]/[op2])
%=	[op1]%=[op2]	I	Assign with Modulo ([op1] = ([op1] % [op2])
&=	[op1]&=[op2]	I, B	Assign with AND ([op1] = [op1] AND [op2])
|=	[op1]|=[op2]	I, B	Assign with OR ([op1] = [op1] OR [op2])
^=	[op1]^=[op2]	I, B	Assign with exclusive OR ([op1] = [op1] XOR [op2])
Bit operators			
<<	[op1]<<[op2]	I	Left shift
>>	[op1]>>[op2]	I	Right shift
&	[op1]&[op2]	I	AND
^	[op1]^[op2]	I	Exclusive OR
|	[op1]||[op2]	I	Inclusive OR

[1] [op] = tag (e.g. "RV12", see "Programmable values and states (write tags)", page 75 and see "Applicable values and states (read tags)", page 74)
[2] R = floating point value (real), I = integer value (integer), B = Boolean value.
[3] Usable for countdown timer and cyclic trigger.

13.3.4 Conditions

Characters	Usage	Function
() ? () : ()	([term1]) ? ([term2]) : ([term3])	When [term1] is TRUE, [term2] is performed otherwise [term3] is performed → (if) ? (then) : (else)

13.3.5 Formula combinations

Several terms can be performed after each other in formulas.

Terminology	Function
[term1]; [term2]; ...; [termX]	Formula flag from several terms performed after each other. The result of the last term is the result of the formula.

13.3.6 Mathematical functions in formulas

Characters[1]	Function	Type[2]
sqrt([op])	[op]$^{-2}$ (square root of [op])	R
exp([op])	$e^{[op]}$	R

Characters[1]	Function	Type[2]
log10([op])	log$_{10}$([op]) (logarithm of base 10 of [op])	R
log([op])	log$_{e}$([op]) (logarithm of base e of [op])	R
frac([op])	Decimal places for [op]	R
abs([op])	Absolute value for [op]	R/I
sgn([op])	Sign for [op]	I
ceil([op])	Round up [op] (example: 1.1 → 2.0)	I
floor([op])	Round down [op] (example: 1.9 → 1.0)	I
bnd([op])	– When [op] < –1: → –1 – When [op] > +1: → +1 – When [op] = –1 … +1: No change	R/I

[1] [op] = operand.
[2] R = floating point value (real), I = integer value (integer).

13.3.7 Priority of formula operations

The sequence of operations depends on the priority of the operators. As in mathematics, operators can be combined in brackets to regulate the sequence.

1	( )	10	^
2	++ –– + – ~ ! >– >+ !– !+ // (type)	11	|
3	**	12	&&
4	* / %	13	^^
5	+ –	14	||
6	<< >>	15	? :
7	< <= > >=	16	= += –= *= /= %= &= ^= |=
8	== !=	17	,
9	&	18	;

13.4 Technical data

13.4.1 General data

Display:	LCD, monochrome, 63 mm x 63 mm
Operating temperature:	0 … +50 °C
Storage temperature:	–25 … +85°C
Degree of protection:	IP 20[1]
Weight:	0.53 kg[2]
Connection to interface module:	Ribbon cable, 26 poles, ERNI SMC plugs on both ends Max. length: 50 cm, Part No.: 6029960 (20 cm)

[1] Higher degree of protection depending on fitting type.
[2] Incl. front panel.

13.4.2 Ethernet parameters

Parameter	Specification	Example
Transfer type:	TCP/IP Peer-to-Peer	
Transfer parameter:	10 Mbit/s full duplex	
Address:	Freely selectable[1]	192.168.0.1

Parameter	Specification	Example
Subnet mask:	Selectable in value[1]	255.255.255.0
Port:	2112	

[1] Must match the allowable address range of the network used; set using the menu functions on the display of the BCU (→ “BCU Basic Control Unit” “Operating Instructions”).

NOTE

Modbus specifications, see "Modbus specifications with the BCU Basic Control Unit", page 57

13.4.3 Electronic system integration

Internal data traffic:	CAN bus
Network interface:	Ethernet connection on/in GMS800 enclosure

13.4.4 Auxiliary power supply

Voltage supply:	24 VDC
Power input:	4 W

14 Formulas

14.1 Introduction to formulas

14.1.1 Function of formulas

BCU “formulas” are programmable mathematical or logical functions.

- Mathematical functions serve to compute internal values or values from external sources.
- Logical functions serve, for example, to link internal messages or messages from external sources, or to control functions or processes.

Formulas can be used

- In the Formula Table (see "Formula Table", page 82)
- During other parameter settings where allowed (e.g. in sequence control programs)

14.1.2 Application options for formulas

- Assign measured values and other process values to analog signal connections
- Assign switching functions to digital signal connections
- Process and activate electronic states
- Start and control processes (e.g. adjustment procedures)
- Process conditions (if-then-else)
- Convert and calculate process values
- Activate and deactivate formulas

14.1.3 Formula elements (overview)

• Tags that can be used in formulas	see "Formula elements", page 74
Usable as required:	
• Operators to link tags	see "Operators in formulas", page 76
• Variables	see "Variables", page 80
• Activation variables	see "Activation variables", page 81
• Mathematical functions	see "Mathematical functions in formulas", page 77
• Conditions	see "Conditions", page 77
• Formula combinations	see "Formula combinations", page 77

14.1.4 Variables

- A variable can be assigned to a formula term. If the variable tag is used in a formula, the variable then substitutes the tag with the last result from the assigned formula term.
- Boolean variables represent a logical (binary) state or the result from a logical link or condition. Boolean variables can, for example, control digital outputs and can be used in formula conditions.
- 24 variables can be used (RVi/IVi/BVi, see "Programmable values and states (write tags)", page 75) for each value type (R/I/B, see "Value types", page 80). Variables can be given a name and a physical unit (see "Defining variables", page 85), and displayed as measured values (see "Measuring screens in SOPAS ET", page 30).

14.1.5 Value types

3 value types can be used in terms:

Value type		Rules	Example
Floating point value (real value)	R	Operations with a floating point value and an integer value have a floating point value as result.	Term: 8/5.0 Result: 1.6
Integer value	I	Operations with two integer values have an integer value as result.	Term: 8/5 Result: 1
		The decimal places are truncated when a floating point value is converted to an integer value. This is also valid for results from operations where integer variables (IVi) are assigned.	Term: IV08 = RV08 Value: RV08 =1.9 Result: IV08 = 1
Boolean value	B	The rules for Boolean algebra apply here (logical links).	-

! **NOTICE**
Use the decimal point (.) as decimal character. Example: 0.25

14.1.6 Activation variables

An “activation variable” is a Boolean variable that can activate or deactivate a particular formula or formula group. The associated formula or group is deactivated when the value “0” is assigned to the activation variable. Value “1” activates the respective formula or group.

This serves to activate and deactivate each formula or formula group with a different formula.

14.1.7 Formula examples

Formula	Function
frac(2.456)	Result: 0.456
floor(RV09=2.5764)	1 Value “2.5764” is assigned to variable RV09. 2 The value is rounded down. Result: 2
(S5S==3)?(BV7=1):(BV7=0)	Condition-dependent control of a logical parameter. Result: – When the operating state of sensor 5 is 3; logical variable BV7 = 1. – Otherwise: logical variable BV7 = 0.

RV02 = RV01*(RV20/273.15)/(RV21/1013.25)
Same meaning:
RV02 = RV01*RV20/273.15/(RV21/1013.25)
RV02 = RV01*RV20/273.15/RV21*1013.25
	Conversion of a measured gas concentration to normal conditions (T = 0 °C = 273.15 K, p = 1013.25 hPa).
	Values:
	– RV01= current measured value
	– RV20 = current temperature [K]
	– RV21 = current pressure [hPa]
	Result: RV02 = scaled measured value

! **NOTICE**
- Observe the sequence of operators (see "Priority of formula operations", page 78).
- Use brackets to determine the correct sequence of operations.

14.2 Menu functions for formulas

14.2.1 Formula Table

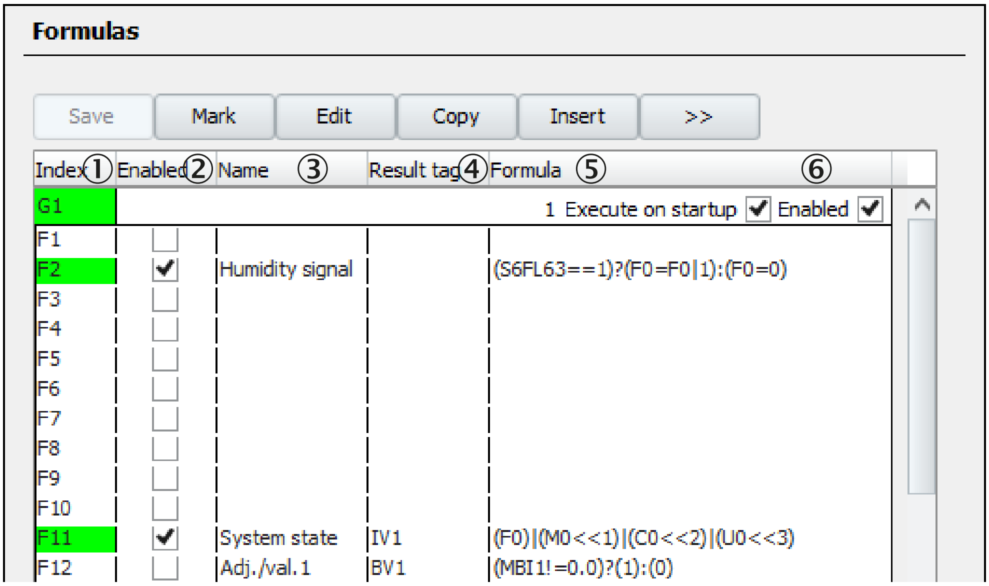
Function

40 formulas can be programmed. “Active” formulas are valid continuously and are recalculated every 1 second during operation (in “Index” number sequence). Results are immediately effective. Non-active formulas are not considered during operation, but remain stored.

Formulas can be combined in “groups”; formulas in a group can be activated or deactivated together. Formulas and formula groups can be activated and deactivated through other formulas.

Procedure

Menu: **BCU/Parameter/Formulas/Formulas**



Index ①	Enabled ②	Name ③	Result tag ④	Formula ⑤	⑥
G1					1 Execute on startup ☑ Enabled ☑
F1	☐				
F2	☑	Humidity signal		(S6FL63==1)?(F0=F0|1):(F0=0)	
F3	☐				
F4	☐				
F5	☐				
F6	☐				
F7	☐				
F8	☐				
F9	☐				
F10	☐				
F11	☑	System state	IV1	(F0)|(M0<<1)|(C0<<2)|(U0<<3)	
F12	☐	Adj./val.1	BV1	(MBI1!=0.0)?(1):(0)	

Figure 56: Menu “Formulas” – Table (example))

① Name of group (G) or formula (F)

- Green text = formula being performed (is active).

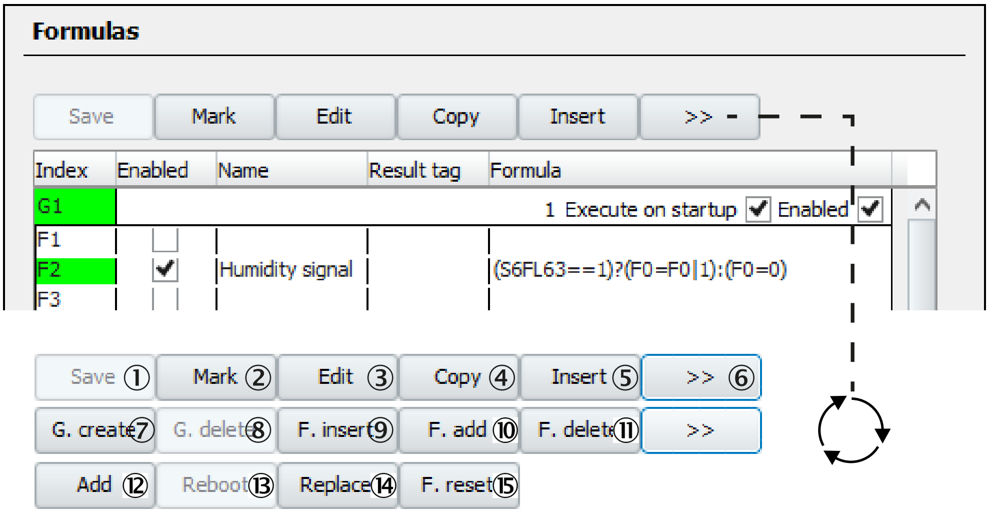
②/⑥ Status of group (see "Adding a formula group", page 83) or formula (see "Programming a formula", page 84)

③ Formula name (freely selectable text, max. 16 characters)

④ Variable assigned to the formula result

⑤ Programmed formula term

Functions in menu “Formulas”



Index	Enabled	Name	Result tag	Formula
G1	1 Execute on startup ☑ Enabled ☑			
F1	□			
F2	☑	Humidity signal		(S6FL63==1)?(F0=F0|1):(F0=0)
F3	□			

Figure 57: Functions in menu Formulas

① Save current state of formulas and groups
② Select several groups/formulas
③ Call up the settings menu for groups/formulas
④ Save a copy of the selected group/formula in intermediate storage
⑤ Insert a copy of the selected formula
⑥ Call up the next function group

⑦ Create a group
⑧ Delete a group
⑨ Insert a new or selected formula
⑩ Attach a new formula at the end of the Formula Table
⑪ Delete the selected formula

⑫ Insert the copied formula after the selected formula
⑬ Trigger a warm start
⑭ Insert the settings of the copied formula in the marked formula
⑮ Delete all individual settings of the selected formula

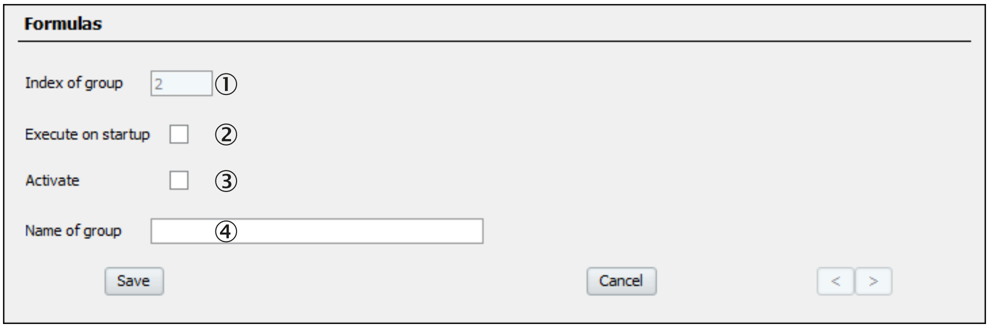
NOTE
Buttons highlighted gray can not be used.

14.2.2 Adding a formula group

1. Select (click) the formula in the Table to be the first formula in the new group.

NOTE
All formulas between the new group and the next group in the Table belong to the new group.

2. Select **G. create**.
3. Select the new group in the Table and select **Edit**.



Formulas

Index of group 2 ①

Execute on startup □ ②

Activate □ ③

Name of group ④

Save Cancel < >

Figure 58: Menu “Formulas” – Edit group (example)

① Group number

② ☑ = all formulas of this group are performed once and activated after start-up of the GMS800 and after a restart[1]

③ ☑ = formulas in this group are active during operation[1]

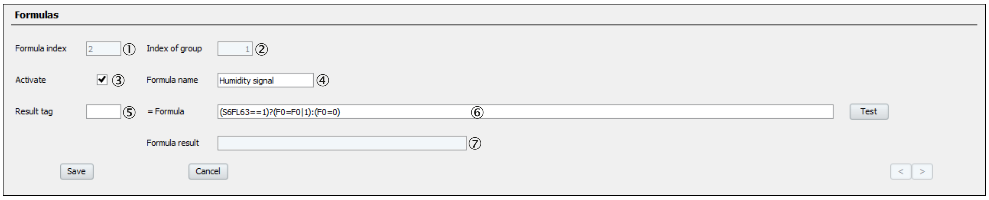
④ Name of the group (freely selectable text, max. 16 characters)

[1] The formulas in the group are only performed when **Execute on start** and **Activate** are activated.

4. Enter desired data.
5. To save the current settings: Select **Save**.

14.2.3 Programming a formula

1. Mark desired Table rows.
2. Select **Edit**.



Formulas

Formula index 2 ① Index of group 1 ②

Activate ☑ ③ Formula name Humidity signal ④

Result tag ⑤ = Formula (S6FL63==1)?(F0=F0|1):(F0=0) ⑥ Test

Formula result ⑦

Save Cancel < >

Figure 59: Menu “Formulas” – Edit formula (example)

① Formula number

② Number of the group to which the formula belongs

③ ☑ = this formula is performed once automatically after start-up of the GMS800 and after a restart[1]

④ Formula name (freely selectable text, max. 16 characters)

⑤ Variable (see "Variables", page 80) to which the result of the formula term is assigned[2]

⑥ Formula term (see "Introduction to formulas", page 80)

⑦ Current result of the formula term

[1] Only when the associated group is also activated through **Execute on startup**.

[2] Use as required; the complete formula can also be entered in the **Formula** field.

NOTE

Context help (available variables, operators, functions) is displayed in the SOPAS ET program window as long as the cursor is positioned to the **Formula** field.

3. Enter desired data.
4. To test a formula (when desired): Select **Test**.
5. To save the current settings: Select **Save**.

NOTICE
The numbers of all following groups are changed automatically when a new group is inserted.

- When groups are used in formulas: Check/adjust group numbers in formulas after inserting a new group.

14.3 Defining variables

14.3.1 Floating point variables (RVi)

Function

24 floating point variables (RVi, real values) can be calculated and used in formulas. Floating point variables have

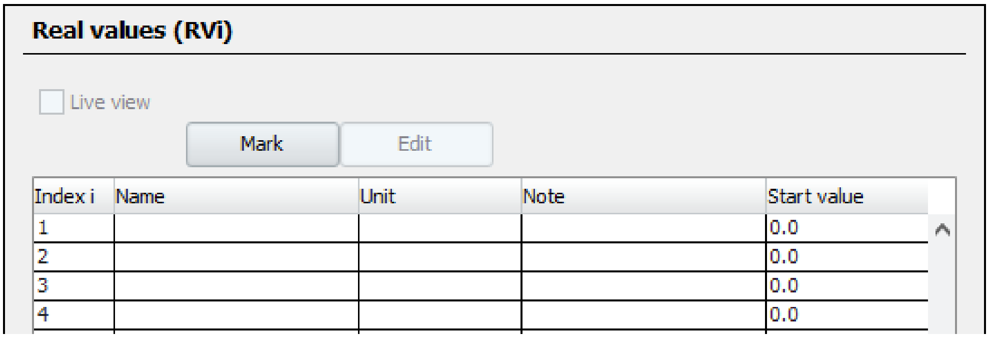
- three descriptive names
- an adjustable start value
- an internal status evaluation the same as a measured value

The current value of a floating point variable can be

- displayed with “name” and “unit” the same as a measured value
- output the same as a measured value.

Procedure

1. Call up menu: **BCU/Parameter/Variables/Real values** (RVi).



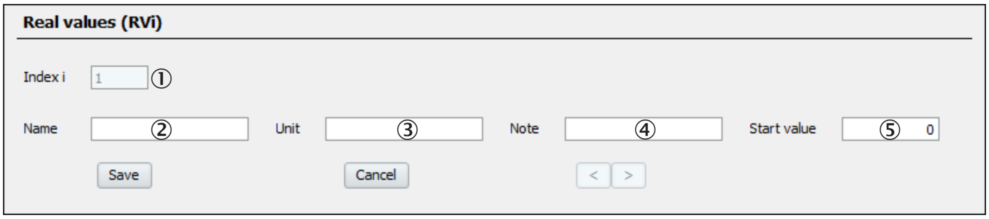
Real values (RVi)

□ Live view

Index i	Name	Unit	Note	Start value
1				0.0
2				0.0
3				0.0
4				0.0

Figure 60: Menu “Real values (RVi)” - Table (example)

2. Deactivate **Live View.**
3. Mark desired Table rows.
4. Select **Edit.**



Real values (RVi)

Index i 1 ①

Name ② Unit ③ Note ④ Start value ⑤ 0

Figure 61: Menu “Real values (RVi)” - Edit (example)

① Consecutive number (1 = RV01, 2 = RV02 etc.)
② Value name (freely selectable text, max. 16 characters)
③ Value unit (freely selectable text, max. 16 characters)
④ Comment (freely selectable text, max. 16 characters)

⑤ Variable value after start-up/restart

5. Enter desired data.
6. Select **Save.**

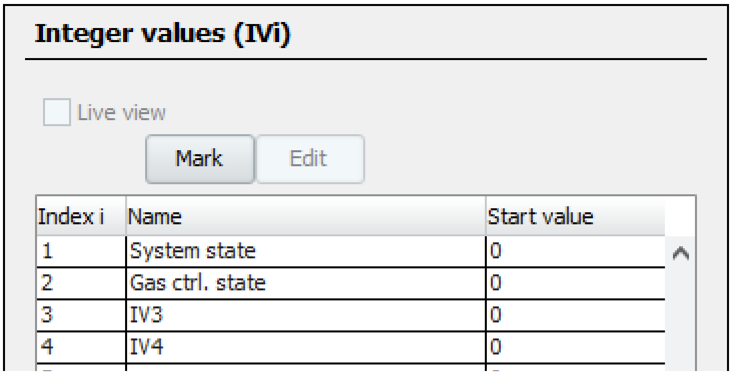
14.3.2 Integer variables (IVi)

Function

Integer variables (IVi, integer values) function the same as floating point variables (see "Floating point variables (RVi)", page 85) but however with integer values. The “name” can be programmed as a descriptive name.

Procedure

1. Call up menu: **BCU/Parameter/Variables/Integer values** (IVi).



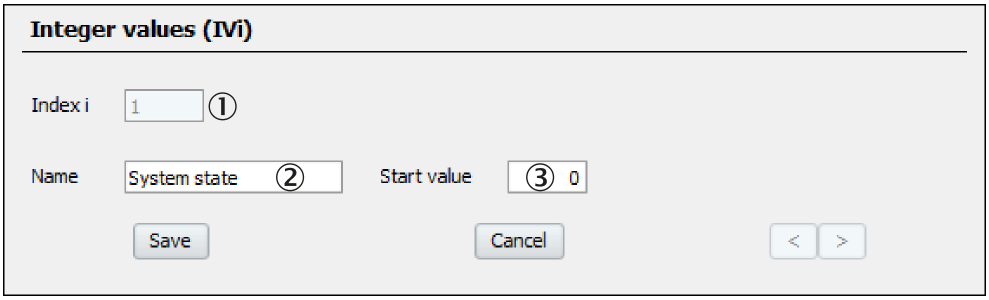
Integer values (IVi)

□ Live view

Index i	Name	Start value
1	System state	0
2	Gas ctrl. state	0
3	IV3	0
4	IV4	0

Figure 62: Menu “Integer values (IVi)” - Table (example)

2. Deactivate **Live View.**
3. Mark desired Table rows.
4. Select **Edit.**



Integer values (IVi)

Index i: 1 ①

Name: System state ②

Start value: ③ 0

Figure 63: Menu “Integer values (IVi)” - Edit (example)

① Consecutive number (1 = IV01, 2 = IV02 etc.)
② Value name (freely selectable text, max. 16 characters)
③ Variable value after start-up/restart for programmed calculations

5. Enter desired data.
6. Select **Save.**

14.3.3 Boolean variables (BVSi) - input signals (system)

Function

Boolean variables for input signals (system) serve to prepare and process internal and external signals, e.g.:

- Internal: Humidity (Gas module), flow rate (Gas module)
- External: Condensate container, temperature

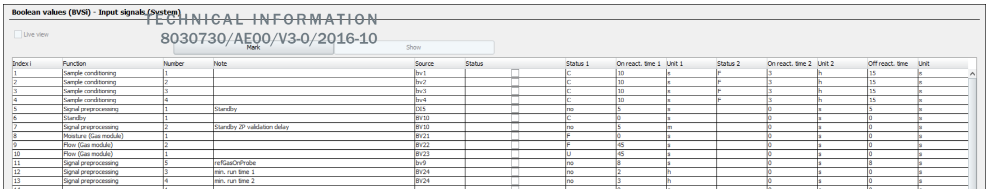
These variables, their function and status can be viewed by authorized operators. The state of the variables is saved when the device is switched off.

The function can be changed in two ways:

a) The value of the relevant Boolean variable (BVSi) is determined with a formula. The respective result from the formula controls or starts the function.
b) The function is assigned to a source. The source state controls or starts the function.

Procedure

1. Call up menu: **BCU/Parameter/Variables/Boolean values** (BVSi) **- Input signals (system).**



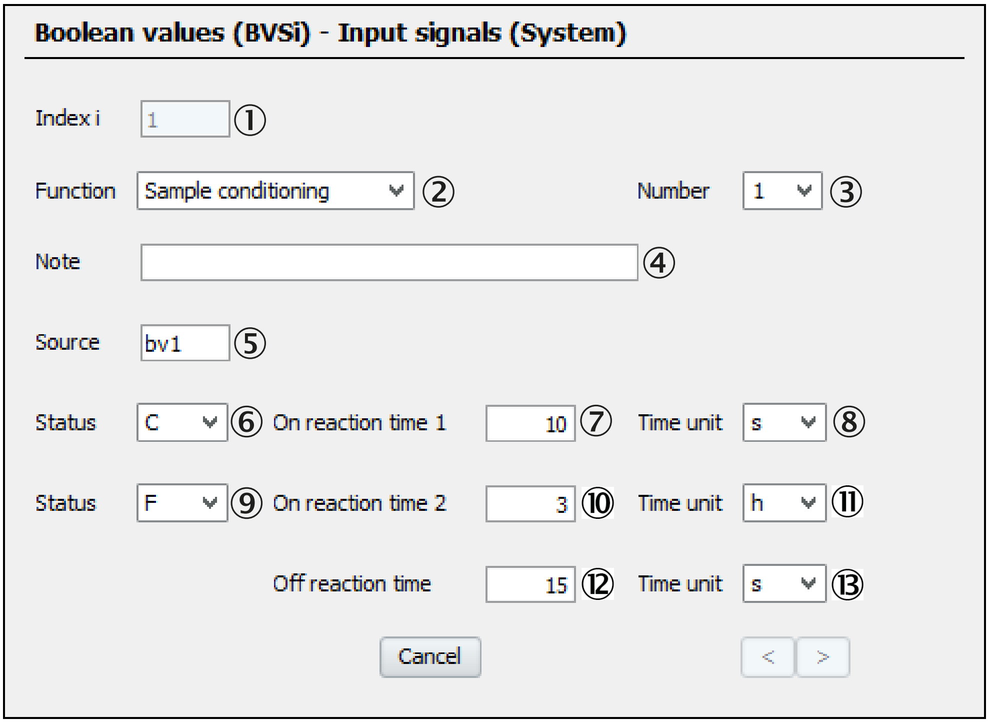
Boolean values (BVSi) - Input signals (System)

□ Live view

Index i	Function	Number	Note	Source	Status	Status 1	On react. time 1	Unit 1	Status 2	On react. time 2	Unit 2	Off react. time	Unit
1	Sample conditioning	1		bv1	□	C	10	s	F	3	h	15	s
2	Sample conditioning	2		bv2	□	C	10	s	F	3	h	15	s
3	Sample conditioning	3		bv3	□	C	10	s	F	3	h	15	s
4	Sample conditioning	4		bv4	□	C	10	s	F	3	h	15	s
5	Signal preprocessing	1	Standby	DI5	□	no	5	s		0	s	5	s
6	Standby	1		BV10	□	C	0	s		0	s	0	s
7	Signal preprocessing	2	Standby ZP validation delay	BV10	□	no	5	m		0	s	0	s
8	Moisture (Gas module)	1		BV21	□	F	0	s		0	s	0	s
9	Flow (Gas module)	2		BV22	□	F	45	s		0	s	0	s
10	Flow (Gas module)	1		BV23	□	U	45	s		0	s	0	s
11	Signal preprocessing	5	refGasOnProbe	bv9	□	no	8	s		0	s	8	s
12	Signal preprocessing	3	min. run time 1	BV24	□	no	2	h		0	s	0	s
13	Signal preprocessing	4	min. run time 2	BV24	□	no	3	h		0	s	0	s

Figure 64: Menu "Boolean values (BVSi) - Input signals (system)" - Table (example)

2. Deactivate **Live View.**
3. Mark desired Table rows.
4. Select **Show.**



Boolean values (BVSi) - Input signals (System)

Index i: 1 ①

Function: Sample conditioning ② Number: 1 ③

Note: ④

Source: bv1 ⑤

Status: C ⑥ On reaction time 1: 10 ⑦ Time unit: s ⑧

Status: F ⑨ On reaction time 2: 3 ⑩ Time unit: h ⑪

Off reaction time: 15 ⑫ Time unit: s ⑬

Figure 65: Menu "Boolean values (BVSi) - Input signals (system)" - Edit (example)

① Consecutive number (1 = BVSi1, 2 = BVSi2 etc.)
② Function name
③ Number
④ Note
⑤ Select source
⑥ Set status for 1

⑦ Reaction time on 1. The status of the variables is first changed after the signal is present for the selected time.
⑧ Time unit
⑨ Set status for 2
⑩ Reaction time on 2. The status of the variables is first changed after the signal is present for the selected time.
⑪ Time unit
⑫ Reaction time off. The signal first disappears when the signal is not present for the specified time.
⑬ Time unit

5. Enter desired data.
6. Select **Save**.

Example

The desired status for the sensor is defined which the sensor should assume after the specified reaction time has elapsed. After the first reaction time with the first desired status, a second reaction time with a further status can then follow. The reaction times imply a delay of the respective output signal.

The specifications made in the input screen (see figure 65, page 87) mean:

- Reaction time on 1 = 10 s: The status of the sensor is first set to C (Check) when the signal is present for 10 seconds.
- Reaction time on 2 = 3 h: The status change from C to F (Failure) is first made when the signal is present for a further 3 hours.
- Reaction time off = 15 s: The status is first reset when the signal is NO longer present for 15 seconds.

14.3.4 Boolean variables (BVIi) – input signals

Function

The Boolean variables for input signals control adjustment and validation procedures, and device functions. The tabular menu shows the respective function.

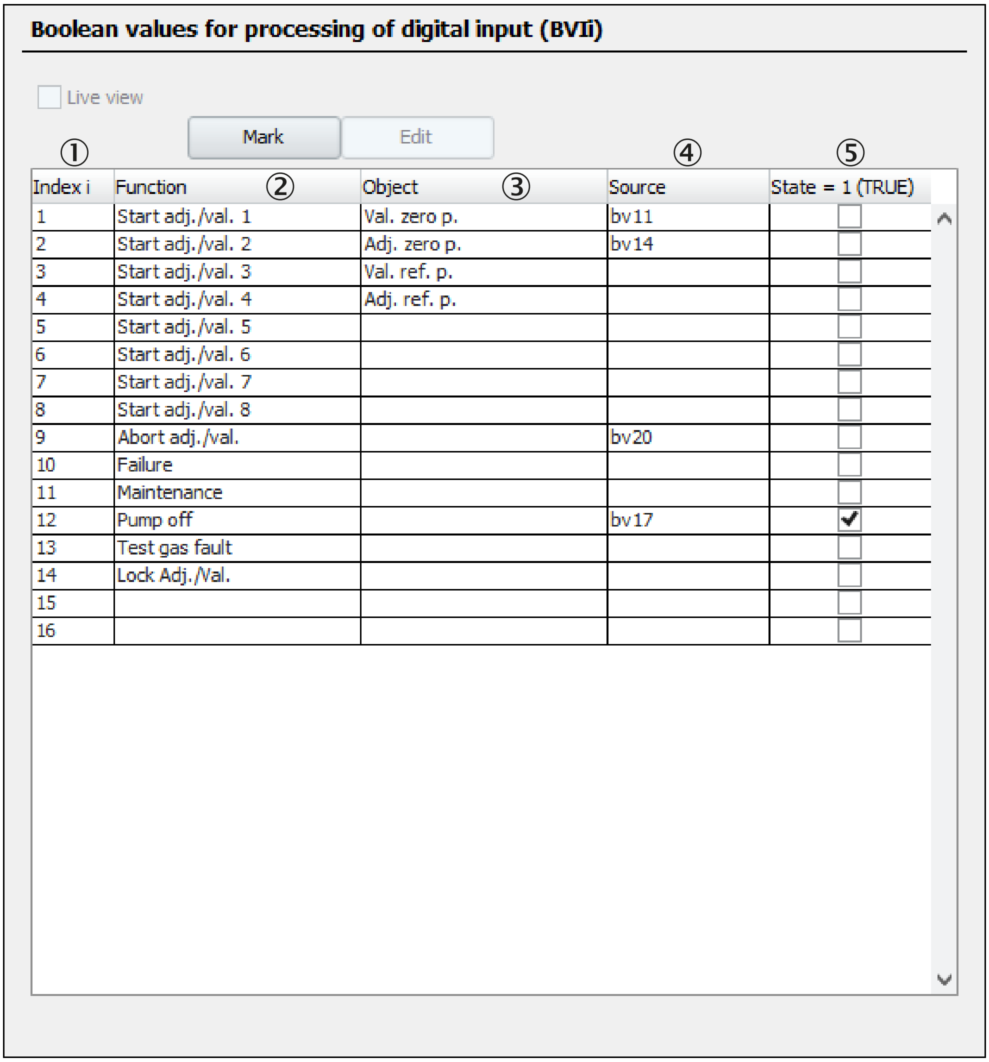
The function is assigned to a source. The source state controls or starts the function.

Tag	Internal function	Indices
DIi	Current logical state of digital input i	i = 1 ... 16
BVi	Current value of Boolean variable BVi	i = 1 ... 24

Table 8: Sources for Boolean values for signal processing of digital inputs

Procedure / Assigning a source

1. Call up menu: **BCU/Parameter/Variables/Boolean values** (BVIi) **- Input signals.**



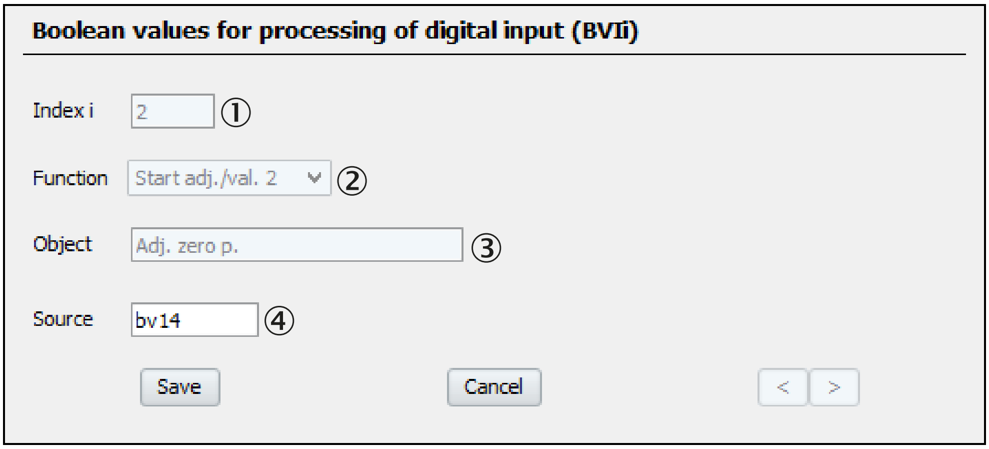
Index i	Function	Object	Source	State = 1 (TRUE)
1	Start adj./val. 1	Val. zero p.	bv11	□
2	Start adj./val. 2	Adj. zero p.	bv14	□
3	Start adj./val. 3	Val. ref. p.		□
4	Start adj./val. 4	Adj. ref. p.		□
5	Start adj./val. 5			□
6	Start adj./val. 6			□
7	Start adj./val. 7			□
8	Start adj./val. 8			□
9	Abort adj./val.		bv20	□
10	Failure			□
11	Maintenance			□
12	Pump off		bv17	☑
13	Test gas fault			□
14	Lock Adj./Val.			□
15				□
16				□

Figure 66: Menu "Boolean values for processing signals of digital inputs (BVIi)")– Table (example)

① Consecutive number (1 = BVI1, 2 = BVI2 etc.)[1]
② Internal name of function[1]
③ Programmed name of function
④ Programmed source for control (see table 8, page 88)
⑤ Current state (☑ = activated)

[1] 1 ...8 correspond to the programmed automatic adjustment/validation functions (see "Programming automatic adjustments/validations", page 49).

2. Deactivate **Live View**.
3. Mark desired Table rows.
4. Select **Edit**.



Boolean values for processing of digital input (BVIi)

Index i: 2

Function: Start adj./val. 2

Object: Adj. zero p.

Source: bv14

Figure 67: Menu “Boolean values for processing of digital input (BVIi)”) – Edit (example)

① Consecutive number (1 = BVI1, 2 = BVI2 etc.)
② Internal name of function
③ Programmed name of function
④ Internal source that determines the function state (see table 8, page 88)

5. Enter desired data.
6. Select **Save**.

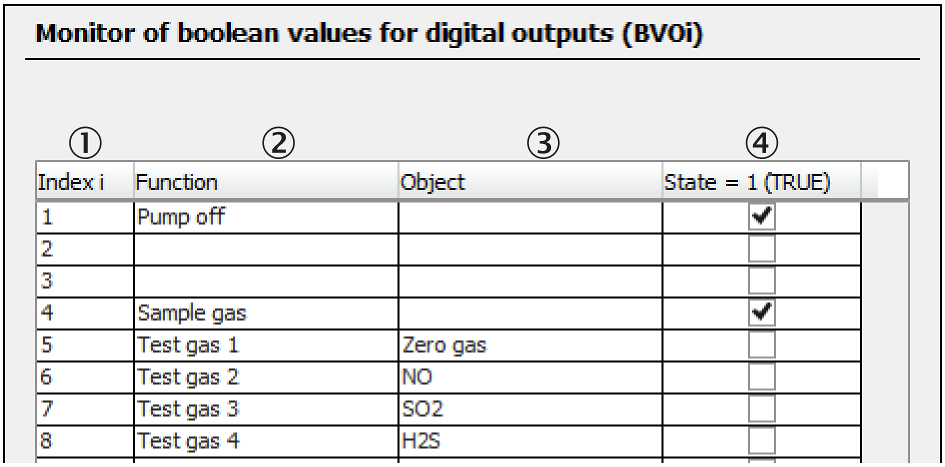
14.3.5 Boolean variables (BVOi) – output signals

Function

Boolean values for output signals are internal state variables for adjustment and validation procedures, and control functions. The tabular menu shows the current state of the functions (monitor function).

Procedure

1. Call up menu: **BCU/Parameter/Variables/Boolean values** (BVOi) - **Output signals.**



Monitor of boolean values for digital outputs (BVOi)

Index i	Function	Object	State = 1 (TRUE)
1	Pump off		☑
2			□
3			□
4	Sample gas		☑
5	Test gas 1	Zero gas	□
6	Test gas 2	NO	□
7	Test gas 3	SO2	□
8	Test gas 4	H2S	□

Figure 68: Menu “Boolean values (BVOi) – Output signals” - Table (example)

① Consecutive number (1 = BVO1, 2 = BVO2 etc.)[1]
② Internal name of function[1]
③ Programmed name of function[1]
④ Current state (☑ = activated)

[1] 5 ...16 correspond to test gas settings 1 ...12 (see "Configuring test gases (Test gas Table)", page 43).

14.3.6 Boolean variables (BVi)

Function

Boolean variables indicate a state. A Boolean variable has an integer value “0” or “1”:

Boolean value	Logical state	Electronic state
0	False	OFF/not activated
1	True	ON/activated

Table 9: Significance of Boolean values

Procedure

1. Call up menu: **BCU/Parameter/Variables/Boolean values** (BVi).
2. Deactivate **Live View**.
3. Mark desired Table rows.
4. Select **Edit**.
5. Enter desired data.
6. Select **Save**.

!

NOTICE
When the Boolean variable is used to control a digital output: Select the “start value” suitable for the switching logic (see "Configuring digital outputs (DOi)", page 38).

14.4 Programming the Function buttons

Function

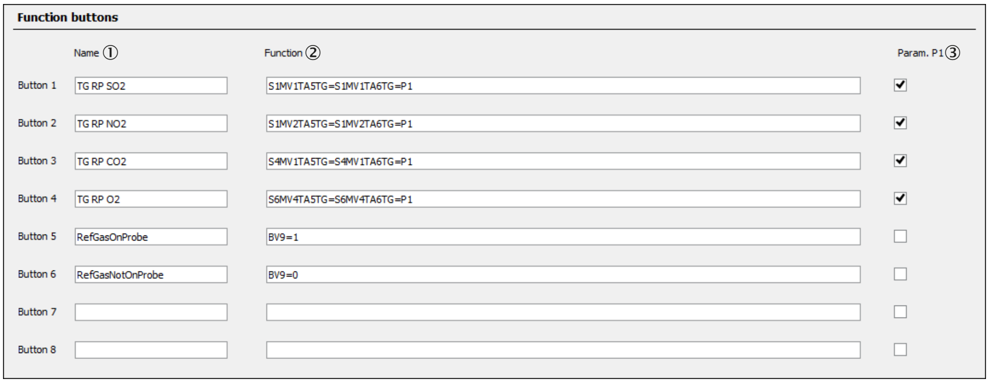
“Function buttons” are programmable formula terms to be performed when the respective “Function button” is clicked (see "Using the Function buttons", page 25). Name and function are freely programmable.

Many different data assignments and function calls are possible and should only be configured with appropriate technical knowledge. Additional documentation is necessary when Function buttons are configured on the device.

A maximum of 8 Function buttons are available. Each Function button executes a function that can be individually programmed (see "Defining variables", page 85).

Procedure

1. Call up menu: **BCU/Parameter/Formulas/Function buttons.**



Function buttons

	Name ①	Function ②	Param. P1③
Button 1	TG RP SO2	S1MV1TA5TG=S1MV1TA6TG=P1	☑
Button 2	TG RP NO2	S1MV2TA5TG=S1MV2TA6TG=P1	☑
Button 3	TG RP CO2	S4MV1TA5TG=S4MV1TA6TG=P1	☑
Button 4	TG RP O2	S6MV4TA5TG=S6MV4TA6TG=P1	☑
Button 5	RefGasOnProbe	BV9=1	□
Button 6	RefGasNotOnProbe	BV9=0	□
Button 7			□
Button 8			□

Figure 69: Menu “Function buttons” (example)

① Button name (max. 32 characters)
② Formula term (see "Introduction to formulas", page 80)

③ ☑ = a value can be assigned to a variable using the respective Function button. This creates an edit field to set a value.
□ = a function is started using the respective Function button.

2. Enter desired data.

✓ The programmed functions are then available (see "Basic settings", page 29).

15 Sequence control programs

15.1 Introduction to sequence control programs

15.1.1 How sequence control programs function

Sequence control programs support automatic conditional processes. 4 sequence control programs can be programmed in the BCU. 32 program sections can be programmed for each sequence control program. Each program section contains an “action” that can be linked to a “condition”. Program sections are chained into a program according to their “step” number, see "Program flow", page 93).

NOTE
Adjustment and validation sequences are already programmed in the BCU at the manufacturer's factory (see "Adjustment functions", page 43).

15.1.2 Starting sequence control programs

- A sequence control program starts automatically when its “start condition” and “enable condition” are met. These conditions are checked every second during operation when the sequence control program is idle. The sequence control program runs continuously when the start and enable conditions are set to “true”.
- Sequence control programs can be started directly with a tag in a formula term (see "Programmable values and states (write tags)", page 75).
- Sequence control programs can also be started directly using the **Start** button in the Edit function, e.g. for test purposes (see "Programming flow conditions", page 95).

15.1.3 Aborting sequence control programs

Running sequence control programs can be aborted with a Stop command in a formula term. This allows aborting a sequence control program automatically when a certain condition occurs, e.g. when an internal status or the status of a digital input changes.

The **step (state) after abort** specifies which program step automatically follows after a stop. Normally, a process that reestablishes the normal operating state is linked.

NOTE
Sequence control programs can also be aborted using the **Stop** button in the Edit function (see "Programming flow conditions", page 95).

15.1.4 Program flow

Step numbers

Each program section is assigned a step number. This step number is the address of the program section within the sequence control program. It does not have to be identical with the index number of the program section.

Step numbers are principally freely selectable within the range 1 ... 32. It is however advantageous from a technical viewpoint when the step numbers are in consecutive sequence.

Rules for step numbers

- Sequence control programs call up the program section with step number “1” automatically when starting.
- Each program section has a **next step** – this is the step number of the program section to be called next.
- Next step “0” means: The sequence control program stops after the **action.**
- Next step “–1” means: Abort the sequence control program and execute the **step (state) after abort** (see "Programming flow conditions", page 95).

!

NOTICE

▶ Assign step number “1” to the program section with which the sequence control program should start.

▶ Enter a **subsequent step** in each program section.

ℹ

NOTE

Next step “0” instructs the sensors to continue measuring operation (corresponds to term “SiMVjTAkS=0”, see "Programmable values and states (write tags)", page 75). This prevents the sensors being blocked by an ongoing action.

Program sections with the same step number

Several program sections can have the same step number. This results in alternative program steps that are valid in their “Index” number sequence. When a certain step number is called and several program sections with this step number exist, the sequence control program initially attempts to execute the first of these program sections (program section with the lowest “Index” number).

▶ This program section is then performed when the start and enable conditions are met; then the “next step” is called.

▶ If the start and enable conditions for the program section are not met, the next program section with the same step number is called (in “Index” number sequence).

This flow is valid for all program sections with the same step number.

Process flow control with a countdown timer

To create a delay time or time lag in a sequence control program:

1. Start one of the countdown timers (SCCDi) as **action** (see "Setting countdown timers", page 99). Enter the step number of the next program section as **next step.**
2. Define a **start condition** in the next program section that starts the **action** as soon as the countdown times has reached the desired state.

Regular start using a cyclic trigger

Starting a sequence control program in regular intervals:

1. Set a cyclic trigger accordingly (see "Setting a cyclic trigger", page 98).
2. Inquire the status of this cyclic trigger in a sequence control program as **start condition** (formula term: “CTi!= 0“).

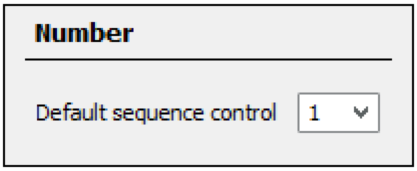
15.2 Determining the number of sequence control programs

Function

The number of sequence control programs available for selection is adjustable (maximum 4).

Procedure

1. Call up menu: **BCU/Parameter/Sequence control programs/Number.**



Number

Default sequence control: 1

Figure 70: Menu "Number"

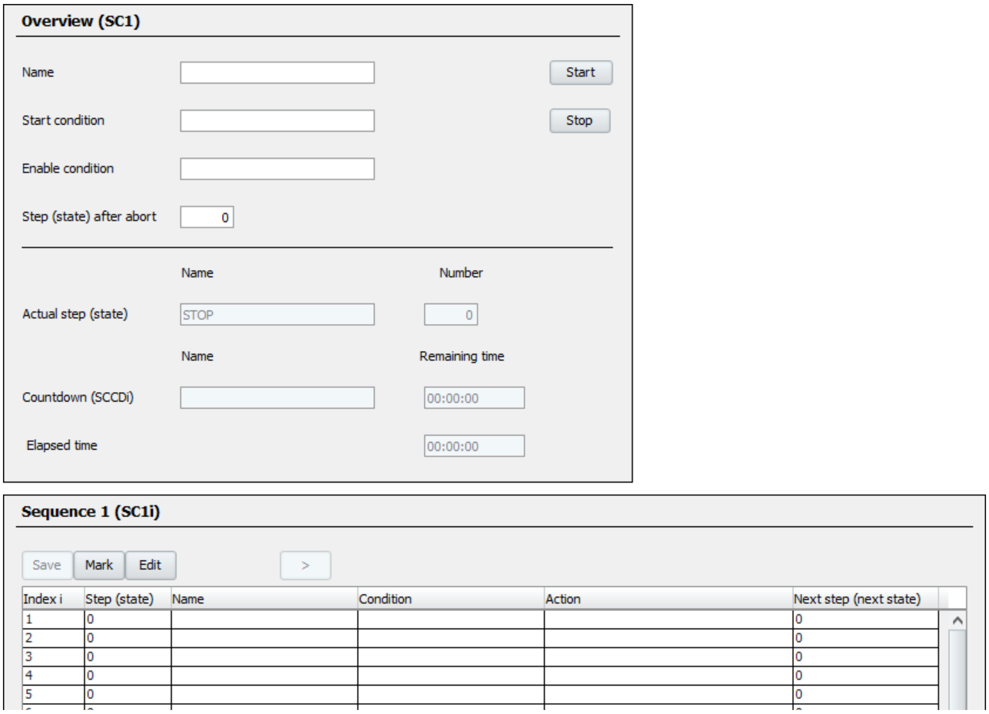
2. Select desired count.
3. Select **Save** to save the setting and exit the menu.

15.3 Programming sequence control programs

NOTE
Detailed explanation on sequence control programs, see "Introduction to sequence control programs", page 93

Procedure

1. Call up menu: "**BCU/Parameter/Sequence control programs/Sequence control programs/ Overview i (SCi)**
(i = number of the sequence control program, see "Determining the number of sequence control programs", page 94).



Overview (SC1)

Name:

Start condition:

Enable condition:

Step (state) after abort: 0

	Name	Number
Actual step (state)	STOP	0

	Name	Remaining time
Countdown (SCCDi)		00:00:00
Elapsed time		00:00:00

Sequence 1 (SC1i)

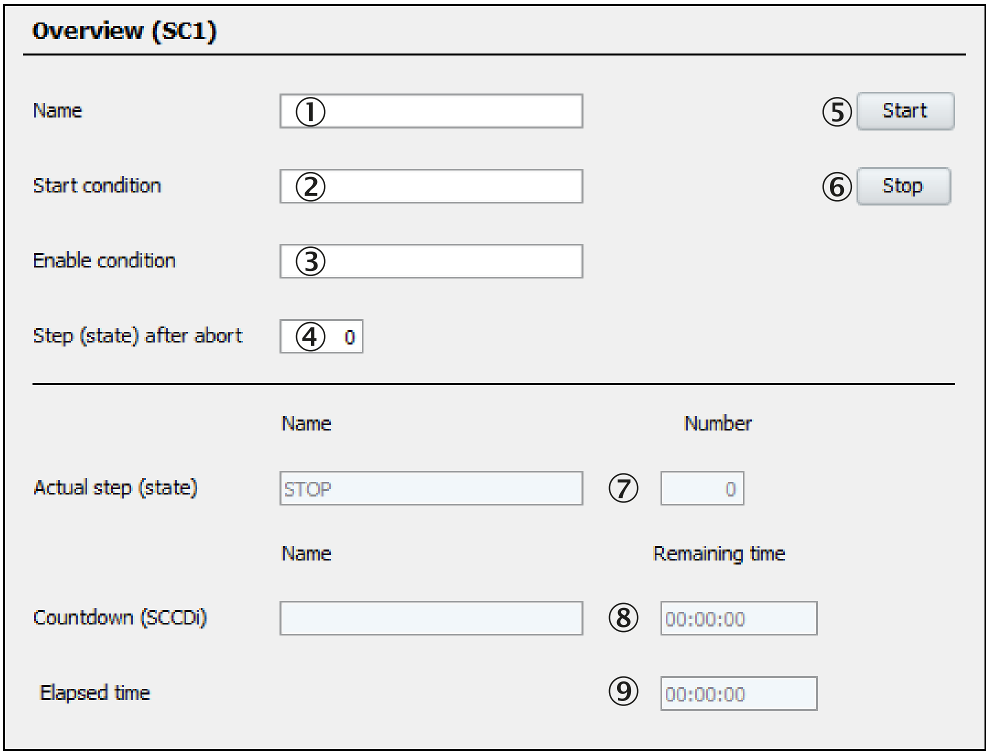
Index i	Step (state)	Name	Condition	Action	Next step (next state)
1	0				0
2	0				0
3	0				0
4	0				0
5	0				0

Figure 71: Menu "Sequence control program x (SCx)"

15.3.1 Programming flow conditions

Procedure

1. Select **Start condition i (SCi).**
2. Enter desired data.


Overview (SC1)

Name:

Start condition:

Enable condition:

Step (state) after abort: 0

Actual step (state): Name: STOP; Number: 0

Countdown (SCCDi): Name: ; Remaining time: 00:00:00

Elapsed time: 00:00:00

Figure 72: Menu “Sequence controls x (SCx)” – “Start conditions” (example)

① Sequence control program name (freely selectable text, max. 32 characters)
② Formula term defining a logical or mathematical relation. The “enable condition” is checked when this term is “true”.[1]
③ Formula term defining a logical or mathematical relation. The sequence control program starts when this term and the “start condition” are “true”.[1]
④ Step number of the program section to be called when the sequence control program has been aborted[2]
⑤ Starts the sequence control program (without considering start and enable conditions)
⑥ Terminates the sequence control program immediately

Information during process flow:

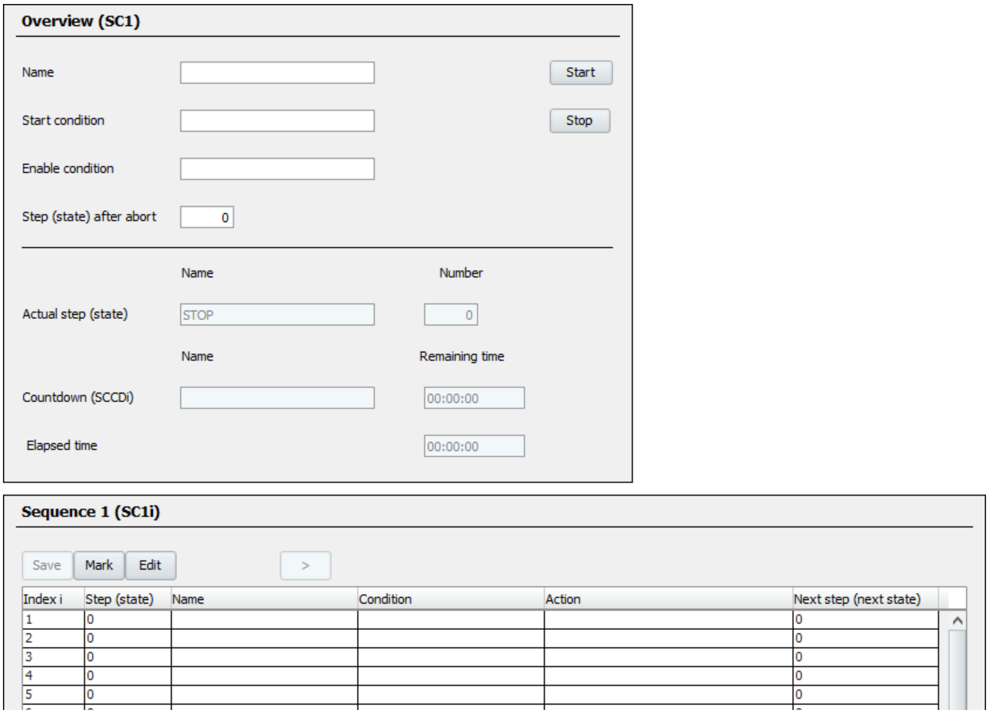
⑦ Name: Programmed identifier of the program step currently running
Number: Programmed step number of the program step currently running
⑧ Name: Name of the currently active countdown timer
Residual time: Current residual time[3] of the currently active countdown timer
⑨ Current running time for the sequence control program[4]

[1] Enter “True” or “1” when the condition is to be permanently TRUE.
[2] Recommendation: Start a program sequence that reestablishes the normal operating state reliably. Example: Reset valves and wait for a delay time to elapse before status “Maintenance” is deactivated again.
[3] When the countdown timer has run down: “ZERO”
[4] When all countdown timers have run down: “0:00”

15.3.2 Programming a program section

Procedure

1. Call up menu: **BCU/Parameter/Sequence control programs/Sequence control programs/Overview i (SCi)**.



Overview (SC1)

Name ___ Start

Start condition ___ Stop

Enable condition ___

Step (state) after abort 0

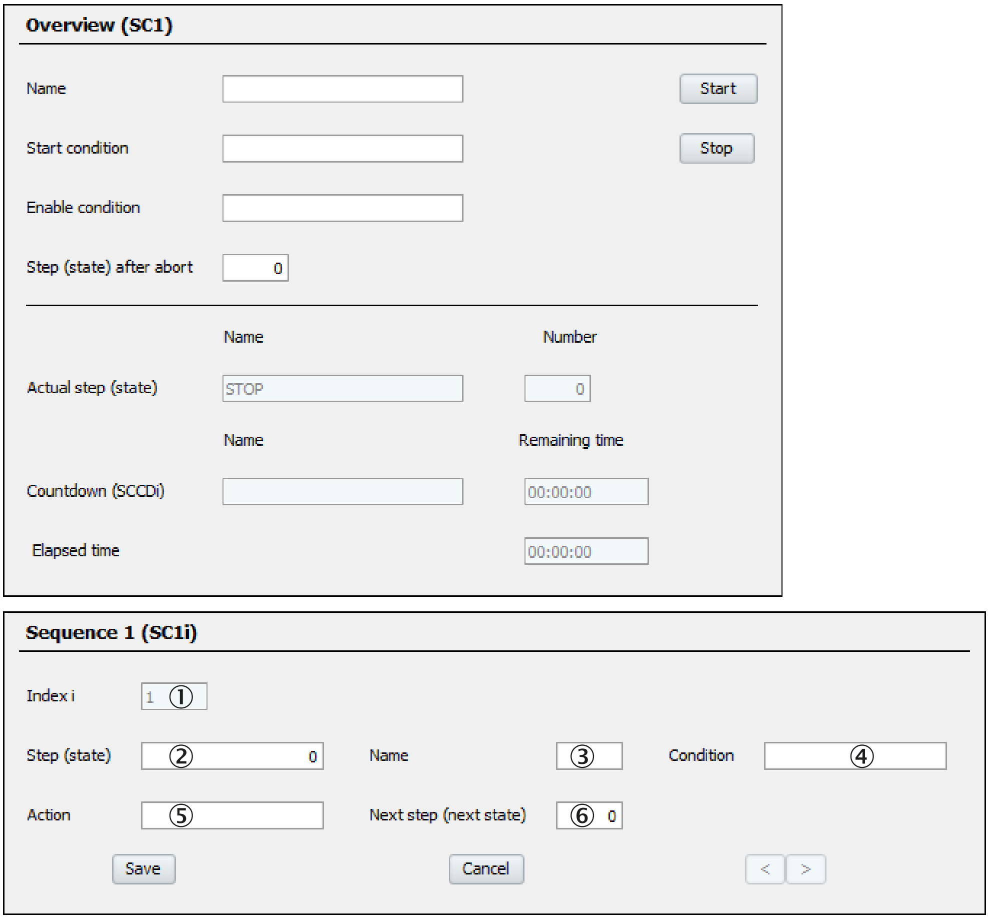
	Name	Number
Actual step (state)	STOP	0

	Name	Remaining time
Countdown (SCCDi)		00:00:00
Elapsed time		00:00:00

Sequence 1 (SC1i)

Index i	Step (state)	Name	Condition	Action	Next step (next state)
1	0				0
2	0				0
3	0				0
4	0				0
5	0				0

2. Select desired program section.
3. Select **Edit**.



Overview (SC1)

Name ___ Start

Start condition ___ Stop

Enable condition ___

Step (state) after abort 0

	Name	Number
Actual step (state)	STOP	0

	Name	Remaining time
Countdown (SCCDi)		00:00:00
Elapsed time		00:00:00

Sequence 1 (SC1i)

Index i 1 ①

Step (state) ② 0 Name ③ Condition ④

Action ⑤ Next step (next state) ⑥ 0

Figure 73: Menu “Program i (SCI)” – Table (example)

① Index number of this program section
② Step number of this program section (0 ... 32; explanation, see "Program flow", page 93)
③ Any text
④ Formula term defining a logical or mathematical condition. The **action** is executed when this condition is true.
⑤ Formula term (max. 64 characters) to be executed when the “condition” is true
⑥ Step number of the program section to be called after the **action** has been executed.

4. Enter desired data.
5. Select **Save**.

NOTE
Note on term **State**: In the theoretical model of the “automat”, a certain status of the parameters of the automat is regarded as a “state”. A change in the parameters leads to a **resulting state**. A **step** in a sequence control program represents a “state”.

15.4 Timers

15.4.1 Setting a cyclic trigger

Function

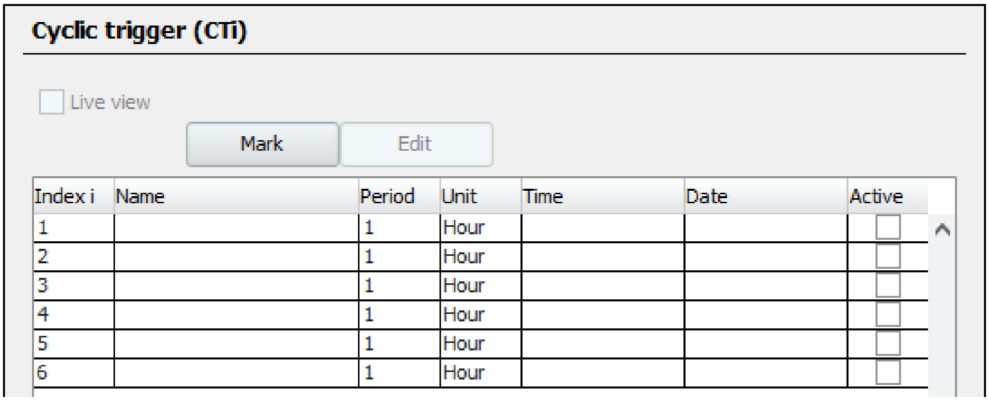
Cyclic triggers can be used in formulas to start a process in regular intervals or to change a state regularly. The next timepoint and time interval (cycle) are programmable.

The start of adjustment and validation functions has already been programmed at the manufacturer's factory and provided in the BCU (see "Adjustment functions", page 43).

NOTE
Application example, see "Regular start using a cyclic trigger", page 94.

Procedure

1. Call up menu: **BCU/Parameter/Timers/Cyclic trigger (CTi)**.



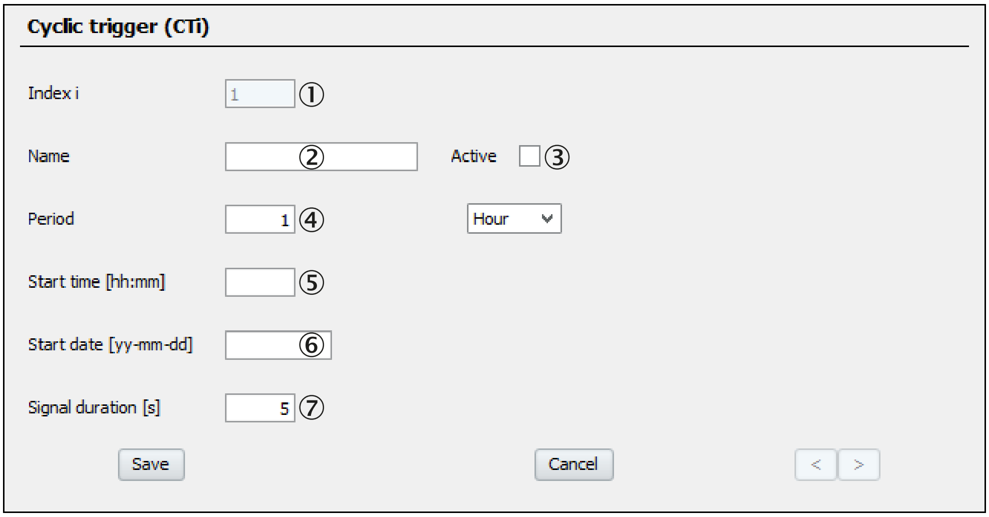
Cyclic trigger (CTi)

☐ Live view

Index i	Name	Period	Unit	Time	Date	Active
1		1	Hour			☐
2		1	Hour			☐
3		1	Hour			☐
4		1	Hour			☐
5		1	Hour			☐
6		1	Hour			☐

Figure 74: Menu “Cyclic trigger” – Table (example)

2. Deactivate **Live View**.
3. Mark desired Table rows.
4. Select **Edit**.



Cyclic trigger (CTi)

Index i: 1 ①

Name: ② Active □ ③

Period: 1 ④ Hour

Start time [hh:mm]: ⑤

Start date [yy-mm-dd]: ⑥

Signal duration [s]: 5 ⑦

Figure 75: Menu “Cyclic trigger” – Edit (example)

① Cyclic trigger number
② Cyclic trigger name (freely selectable text, max. 32 characters)
③ ☑ = this cyclic trigger is running
④ Interval for automatic starts (hours/days/weeks selectable)
⑤ Time when the next automatic start runs[1]
⑥ Calendar day for which the start time is applicable
⑦ Duration of the start status (seconds) after the start time has been reached[2]

[1] The further start points determine the “cycle”. Setting the internal clock, see "Configuring device (BCU) parameters", page 29.

[2] Standard: 2 ... 5 seconds

5. Enter desired data.
6. Select **Save**.

NOTE

The next start time can however be deferred by resetting the start time and start date.

15.4.2 Setting countdown timers

Function

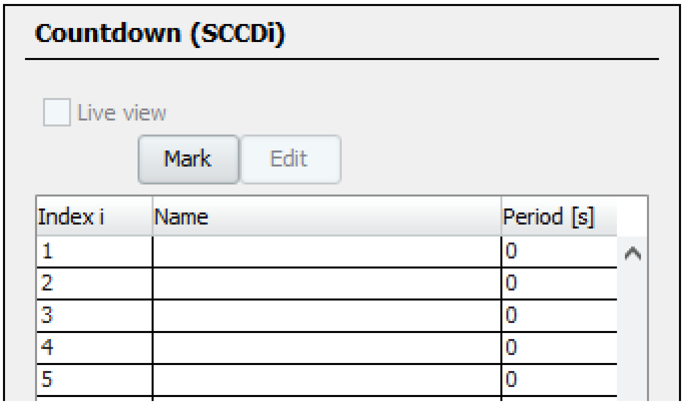
Countdown timers are needed for time controls in sequence control programs. There are 16 countdown timers.

NOTE

- Countdown timers can be paused and continued with formula tags (see "Programmable values and states (write tags)", page 75).
- The current remaining time of the countdown timers can be used in formulas.

Procedure

1. Call up menu: **BCU/Parameter/Timers /Countdown (SCCDi)**.



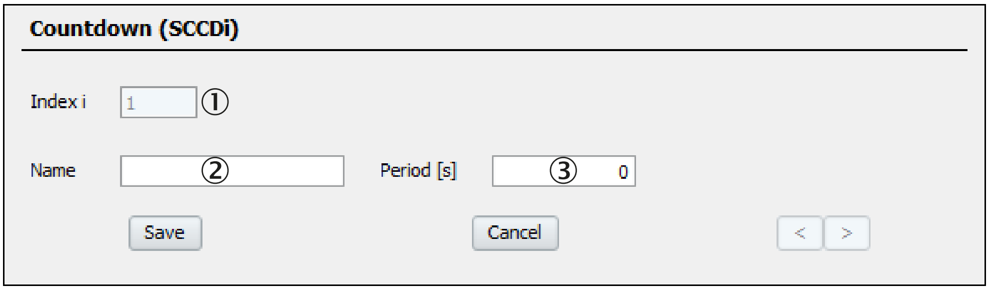
Countdown (SCCDi)

□ Live view

Index i	Name	Period [s]
1		0
2		0
3		0
4		0
5		0

Figure 76: Menu “Countdown timer (SCCDi)” - Table (example)

2. Deactivate **Live View**.
3. Mark desired Table rows.
4. Select **Edit**.



Countdown (SCCDi)

Index i: 1 ①

Name: ②

Period [s]: ③ 0

Figure 77: Menu “Countdown timer (SCCDi)” - Edit (example)

① Countdown timer number
② Name (freely selectable text, max. 16 characters)
③ Countdown runtime (seconds)

5. Enter desired data.
6. Select **Save**.

8030730/AE00/V3-0/2016-10

www.addresses.endress.com

Endress+Hauser
People for Process Automation