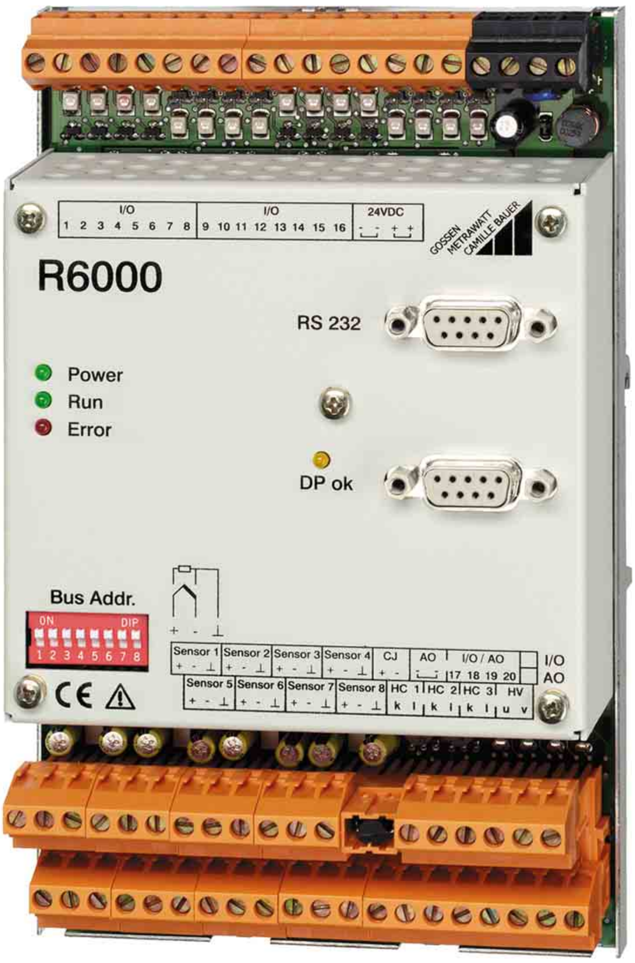
Operating Instructions

R6000

8-Channel-Controller

Z307B
14/5.06



Contents Page

Contents Page

1 Initial Start-Up

Read the operating instructions completely and carefully before using the device, and follow all instructions included therein.

Note
Parameter designations are printed in boldface, and *setting values* are printed in cursive typeface in these operating instructions.

The operating instructions should be made available to all users.

1.1 Safety Precautions

The instrument is manufactured and tested in accordance with safety regulations IEC 61010-1 / EN 61010-1 / VDE 0411 part 1.
If used for its intended purpose, safety of the user and of the device is assured.

Attention!
Check the specified nominal voltage at the front housing panel before placing the instrument into service.
When wiring the instrument, make sure the connector cables are not damaged, and that they are voltage-free.
If it can be assumed that safe operation is no longer possible, the instrument must be immediately removed from service (disconnect auxiliary voltage!). Safe operation can no longer be relied upon if the instrument demonstrates visible damage.
The device may not be placed back into operation until troubleshooting, repair and subsequent testing have been performed at our factory, or at one of our authorized service centers.
Work on live open instruments may only be carried out by trained personnel who are familiar with the dangers involved.
Capacitors inside the instrument may be dangerously charged, even if it has been disconnected from all power sources.

Requirements set forth in VDE 0100 must be observed during the performance of all work.

1.2 Installing the Controller

The instrument must be installed in accordance with separate installation instructions.

Make sure that all relevant criteria have been observed during assembly, preparation, installation, electrical connection and initial start-up by means of identification based upon article number and feature codes.

1.3 Operating the Controller via Interface

Bus interface

Data exchange with the controller can be accomplished via the bus interface.

Descriptions regarding functions, interfaces and data transmission are included in the subsequent chapters.

Service interface

Independent of the bus interface, the controller is equipped with an RS 232 service interface with protocol per EN 60870 (see chapter 3 on page 29), which allows for communication with each individual instrument.
R6KONFIG PC software is available for this purpose. It can be downloaded free of charge from the GMC-Instruments Deutschland GmbH website at: http://www.gossenmetrawatt.com

R6KONFIG PC software

All parameters can be conveniently accessed with R6KONFIG PC software, parameter sets can be saved to memory at the PC, and existing parameter sets can be uploaded to controller. Current measured values (cycle data) can be displayed.

Please read chapter 2 as from page 6 first, for a thorough understanding of R6KONFIG PC software and the controller.

System requirements:

IBM PC or compatible with Pentium 300 MHz processor or higher
Windows 95, 98, NT 4.0 or 2000
64 MB RAM for Windows 95 or 98, 128 MB RAM for Windows NT 4.0 / 2000 / XP
Approximately 5 MB available hard disk space

A separate operating manual for this software is available on the GMC-I Gossen-Metrawatt GmbH homepage.

2 Controller Settings

After installing the controller, its parameters must be configured for the desired task. Parameters can be configured with, for example, R6KONFIG configuration software. Upon delivery, the controller is configured as an 8-channel *3-step PDPI fixed setpoint controller* with *type J thermocouple* (default setting).

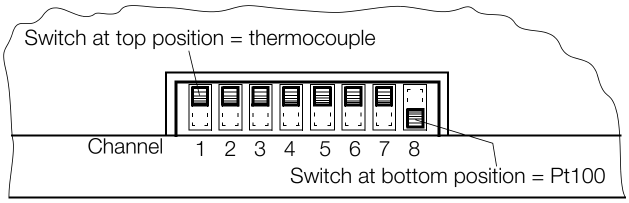
2.1 Basic Configuration as 2 or 3-Step Fixed Setpoint Controller

2.1.1 Configuring the Temperature Measurement Inputs

The 8 temperature measurement inputs are permanently linked to the 8 control channels.
The sensor type can be freely selected for each input.

- Selecting the sensor type with the DIP switches:

Sensor types are selected during installation of the instrument. For the temperature sensor version they are selected separately for each channel with the DIP switches at the left-hand side of the housing. Unused inputs must be set to thermocouple.



- Selecting a sensor with the **sensor type** parameter:

Sensor Type		Measuring Range Lower Limit		Measuring Range Upper Limit		DIP
No.	Type	°C	°F	°C	°F	Switch
0	J	0	32	900	1652	Top
1	L	0	32	900	1652	
2	K	0	32	1300	2372	
3	B	0	32	1800	3272	
4	S	0	32	1750	3182	
5	R	0	32	1750	3182	
6	N	0	32	1300	2372	
7	E	0	32	700	1292	
8	T	0	32	400	752	
9	U	0	32	600	1112	
10	linear [1)]	0 mV		50 mV		
11	Pt100	–200	–328	600	1112	Bottom
12	Ni100	–50	–58	250	482	

[1)] Scalable temperature, observe instructions in chapter 2.3.6 on page 11!

The factory default setting for all temperature measurement inputs is **sensor type:** *type J thermocouple*.

°C or °F can be selected for the transmission of temperature values via the (bus) interface with the parameter:
device control. All temperature quantities are saved in °C at the controller.

Control parameters which are related to manipulated variables (proportional zone heating and cooling, dead zone and switching hysteresis) are also saved in °C for improved clarity, and are thus independent of the selected sensor type.

2.1.2 Configuring the Control Channels

Upon shipment from the factory, the controller channels are configured with default settings including **controller class:** *fixed setpoint controller*, and **controller type:** *PDPI controller*. Whether the channels are configured as 2 or 3-step controllers, or as step-action or continuous action controllers, is determined by the **output configuration**.

The **controller type** should be set to *unused* for channels to which no sensor is connected, or for channels which are not required, in order to avoid unnecessary error messages.

Upon shipment from the factory none of the **controller functions** are enabled, which means that the actuating outputs are inactive.

The **controller on** bit must be set at each desired channel in order to enable controller functions.

2.1.3 Configuring the Actuating Outputs

All binary inputs and outputs and the continuous outputs can be freely assigned to actuating signals and other entry and display functions.

A controller channel can be set up as a 2-step controller by configuring one binary output as a heating output with the corresponding channel number.

A 3-step controller is created when, in addition to the heating output, another binary output is configured as a cooling output with the corresponding channel number.

The 8 bits included in the **output configuration** have the following significance in the case of a binary actuating output:

Bit Number	Value	Meaning
0	0	Configuration as output
1	1	Single channel
2 ... 4	0 ... 7	Channel number
5	0 / 1	Heating / cooling
6	0	Mode
7	0	Actuating signal

The **output configuration** for unused outputs should be set to 0.

Upon shipment from the factory, binary outputs 1 through 8 are set up as heating outputs for channels 1 through 8 in their **output configurations**, and binary outputs 9 through 16 are set up as cooling outputs, which means that all 8 channels are configured as discontinuous-action 3-step controllers.

2.2 Configuration of Controller Outputs and Actuators

2.2.1 2-Step, 3-Step Controllers, Continuous-Action Controllers, Step-Action Controllers

Various actuators for the heating and cooling functions can be freely combined per controller channel.

The controller's output function, i.e. 2-step, 3-step, continuous-action, step-action or combinations thereof, is defined by assigning an **output configuration** to the outputs.

Bit Number	Value	Meaning for Discontinuous-Action Output	Meaning for Continuous-Action Output
0	0	Configuration as output	
1	1	Single channel	
2 ... 4	0 ... 7	Channel number	
5	0 / 1	Heating / cooling	
6	0 / 1	More / less	Dead / live zero
7	0	Actuating signal	

Bits 5 and 6 define the actuator in the **output configuration**.

Heating Actuator	Configuration of 1st Heating Output		Configuration of 2nd Heating Output	
No heating actuator	—		—	
SSR, contactor for discontinuous control	Binary output	bit 5 = "heating" = 0 bit 6 = "more" = 0	—	
(continuous) Proportional actuator	Continuous output	bit 5 = "heating" = 0	—	
Motor actuator for step-action control	Binary output	bit 5 = "heating" = 0 bit 6 = "more" = 0	Binary output	bit 5 = "heating" = 0 bit 6 = "less" = 1

Cooling Actuator	Configuration of 1st Cooling Output		Configuration of 2nd Cooling Output	
No cooling actuator	—			
SSR, contactor for discontinuous control	Binary output	bit 5 = "cooling" = 1 bit 6 = "more" = 0	—	
(continuous) Proportional actuator	Continuous output	bit 5 = "cooling" = 1	—	
Motor actuator for step-action control	Binary output	bit 5 = "cooling" = 1 bit 6 = "more" = 0	Binary output	bit 5 = "cooling" = 1 bit 6 = "less" = 1

- Actuators for heating and cooling are selected independent of each other (this allows for the combination of, for example, step-action control for heating, as well as for cooling.)
- If 2-step control is required, heating and cooling outputs may not be configured simultaneously for the respective channel.
- Several outputs of the same type can be assigned to the same controller output for separate control of several actuators with a single controller output.
- If continuous and discontinuous outputs are configured simultaneously for heating (or cooling), the channel functions as a continuous-action controller, and the step-action outputs are disabled.
- If only a "less" output is inadvertently configured for heating (or cooling), it remains inactive.
- Settings for **controller class** and **controller type** can be freely combined.

2.2.2 Water Cooling

If the **water cooling** bit is set in the **controller configuration**, the cooling manipulated variable is read out in a modified fashion, in order to account for the disproportionately powerful cooling effect which prevails when water is evaporated.

2.2.3 Hot-Runner Controller

If the **hot-runner** bit is set in the **controller configuration**, the heating manipulated variable is read out as a rapidly pulsating signal. This assures that localized overheating is avoided at hygroscopic cartridge heaters during actuation, and prevents temperature fluctuation within the heaters. Further functions which are dependent upon this setting are described in a separate chapter, namely chapter 2.6 on page 15.

2.2.4 Controlling Contactors

If, during ascertainment of control parameters (manual optimization or self-tuning), a **cycle time** results which is significantly shorter than advisable for the service life of the contactor, **cycle time** can be increased all the way up to the limit of system controllability by setting the **contactor** bit in the **extended controller configuration** (PI = 23h). If the bit is set before self-tuning is started, cycle time is set to the highest possible value by the self-tuning function.

2.2.5 Power Limitation

If, for reasons of current loading, it is not permissible or reasonable to have the heaters of all eight control circuits activated simultaneously, the controller can be forced with the **power limitation** (PI = 3Ah) paramter to actuate only a predefined number of heating outputs per device at the same time.

For example, if a maximum of only 5 heaters may be activated at the same time, the power limitation is set to 62% (approx. 5/8). This function is cancelled by entering 0%.

The manipulated variables of the channels with a configured heating output are automatically limited by the controller in accordance with power limitation. The positioning output of the individual channels is synchronized and the heaters are activated with in a staggered pattern.

The actual currents applied (if they were known from heating current monitoring) are not taken into account in this context.

This function is also active when power limitation has been set to 100% so that all eight channels produce full heat during actuation. However, current loading is more evenly spread at the operating point, thus avoiding power peaks.

If self-tuning is launched during active power limitation (see chapter 2.7.1 on page 17), the **actuation cycle time** is not established by the self-tuning function.
It is therefore necessary to adjust a reasonable actuation cycle time for those control loops involved in power limitation or to effect the self-tuning without power limitation.

2.3 Processing Setpoints and Actual Values

2.3.1 Setpoint Ramps, Proxy Setpoint, Setpoint Limiting

- The setpoint ramp is activated when:
  – Auxiliary voltage is switched on / after reset
  – When the setpoint is changed / the proxy setpoint is activated
  – Upon switching from the off state or manual operation to automatic operation
- Setpoint ramps are inactive during self-optimization.
- Relative limit values make reference to the targeted setpoint, not the ramp.
- Corresponding bits are set in **controller status** when setpoint ramps are active.

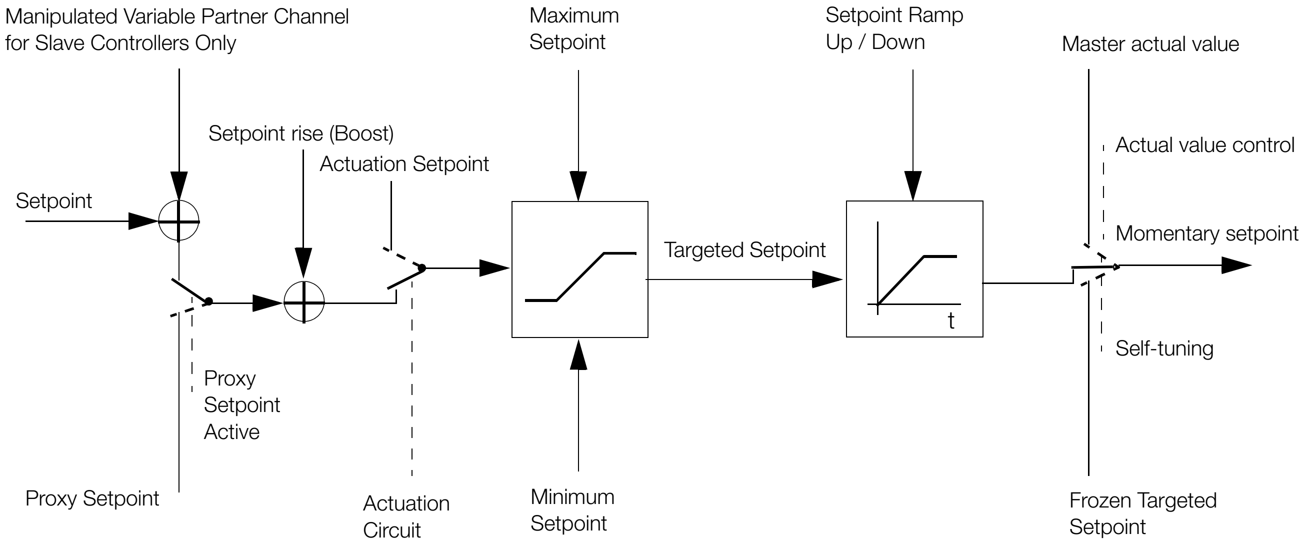


Figure 1 Setpoint Processing Schematic

2.3.2 External Actual Value

If the **external actual value** bit is set in the **extended controller configuration** (PI = 23h), the **external actual value** (PI = 27h) read in via the interface is used instead of the actual value measured by the instrument. Scaling or correction by means of **actual value factor** and **actual value correction** is not possible in this case.

2.3.3 Adaptive Measured Value Correction for Ascertaining the Actual Value

If a control loop is impaired by a periodic interference signal effecting the actual value, control can be improved by activating adaptive measured value correction. Periodic interference is thus suppressed without reducing the controller's ability to respond to system deviation. This is accomplished by adaptively adjusting correction to the oscillation amplitude of the interference signal, and by transmitting only the mean value to the controller.

Bit 14 in the **controller configuration** activates adaptive measured value correction.

Adaptation of the correction to the interference is executed in accordance with controller dynamics, and does not required any other parameters.

Control can only be **improved** if:

- The oscillation amplitude of the interference signal is constant, or changes slowly
- The oscillation period is shorter than half of system delay (compare: PI = 14h)

Due to the fact that this correction greatly influences actual value determination, control may also be **worsened**, for example if:

- Measured value deviation is irregular
- Isolated measured value outliers occur
- Fluctuation is not periodic
- The interference signal is random

2.3.4 Suppression of Periodic Disturbances

If the measured value is superimposed by a heavy periodic oscillation, which is caused, for example, by cyclical withdrawal of energy from the control circuit, the manipulated variable may fluctuate between its extremes and the control result may be unsatisfactory.

When the period is constant, the oscillation can be filtered out by adjusting the period in the **oscillation hold-off** (PI = 25h) parameter. This is done by filtering out the signal component within a narrow band with the adjusted period and by deducting it from the measuring signal for the control action. This leaves the actual display values unaffected.

In contrast to the adaptive measured value correction (see chapter 2.3.3) you can also suppress oscillations with periods longer than half of system delay.

Periods from 0.3 to 25 sec. can be adjusted. The filter remains inactive when other values are adjusted (0 sec. to 0.2 sec. or more than 25 sec.).

Since the hold-off filter influences controller dynamics, it is necessary to determine the control parameters by manual or self-optimization with the oscillation hold-off activated.

2.3.5 Actual Value Correction for Temperature Sensors

If a temperature sensor has been directly connected (i.e. if **sensor type** has not been set to *linear*), both the **actual value correction** and the **actual value factor** parameters can be used to compensate for deviations between measured temperature and the temperature value to be displayed.

The **actual value factor** changes temperature in proportion to the measured value. No change takes place with an **actual value factor** of 100.0% (default setting).

The value selected for the **actual value correction** parameter is added to the measured temperature value (and may also be changed by means of the actual value factor). Excessively large measured values obtained from resistance thermometers and with 2-wire connections are thus corrected.

Two measuring points are required for calculating the parameter setting (the measured value corresponds to temperature prior to correction, and the display value corresponds to temperature after correction):

$$\textbf{Actual value factor} = \frac{\text{display value 1} - \text{display value 2}}{\text{meas. value 1} - \text{meas. value 2}} \bullet 100\%$$

$$\textbf{Actual value correction} = \text{display value} - \frac{\text{meas. value} \bullet \text{actual value factor}}{100\%} \qquad \text{where } \textbf{unit of measure} = {}^\circ C$$

$$\textbf{Actual value correction} = (\text{display value} - 32.0^\circ\ \text{F}) - \frac{(\text{meas. value} - 32\ {}^\circ\text{F}) \bullet \text{actual val. corr.}}{100\%} \qquad \text{where } \textbf{unit of measure} = {}^\circ F$$

Example:

A temperature drop occurs between a tool heater and the surface of the tool. The measured temperature value (at the heater) is 375° C (measured value 1), and the temperature at the surface of the tool (temperature to be displayed) is 245° C (display value 1). The measured value should not be changed at room temperature (i.e. with tool heater switched off).
(Measured value 2 = display value 2 = 23.0° C.)

Solution:

$$\textbf{Actual value factor} = \frac{245^\circ\ \text{C} - 23^\circ\ \text{C}}{375^\circ\ \text{C} - 23^\circ\ \text{C}} \bullet 100\% = 63.1\%$$

$$\textbf{Actual value correction} = 23^\circ\ \text{C} - \frac{23^\circ\ \text{C} \bullet 63.1\%}{100\%} = 8.5^\circ\ \text{C}$$

2.3.6 Using the Thermocouple Input As Linear Input

When the linear input has been selected (**sensor type** = *linear*), the thermocouple input is used without taking the reference junction into consideration.

In the case of high impedance sources, the measured value is influenced as a result of broken sensor monitoring:

Shift:	approx. + 1.2 mV / kΩ
Attenuation:	approx. 0.5% / kΩ

The **actual value correction** and **actual value factor** parameters are used to scale the measured value.

The scaled measured value is treated by the controller as a temperature value because the units of measure for the various controller parameters (e.g. setpoint or proportional band) are specified in °C or °F. Where control or monitoring of quantities other than temperature are involved, the unit of measure for the controlled variable should thus not be changed after scaling, because scaling is converted for °C / °F.

The **actual value factor** is the display range which corresponds to an input range of 0 to 50 mV.

The 0 mV measuring point is displayed as 0.0° C or 32.0° F, as long as **actual value correction** is set to 0.

The value assigned to the **actual value correction** parameter is added to the display value.

Two measuring points are required for calculating the parameter setting (measured values in mV):

$$\textbf{Actual value factor} = \frac{\text{display value 1} - \text{display value 2}}{\text{measured value 1} - \text{measured value 2}} \bullet 50\text{ mV}$$

$$\textbf{Actual value correction} = \text{display value} - \frac{\text{measured value} \bullet \text{actual value factor}}{50\text{ mV}} \qquad \text{where } \textbf{unit of measure} = °C$$

$$\textbf{Actual value correction} = (\text{display value} - 32.0°\text{ F}) - \frac{\text{measured value} \bullet \text{actual value factor}}{50\text{ mV}} \qquad \text{where } \textbf{unit of measure} = °F$$

Example:

Pressure needs to be monitored in addition to temperature control in °F. 44 mV are applied to the input at a pressure of 100 bar, and 0 bar corresponds to 0 mV. The measured value is to be transmitted via the interface with a resolution of 0.01 bar.

Solution:

The resolution of 0.1° F is replaced with a resolution of 0.01 bar for the interpretation of all temperature values.

$$\textbf{Actual value factor} = \frac{100.00\text{ bar} - 0.00\text{ bar}}{44\text{ mV} - 0\text{ mV}} \bullet 50\text{ mV} = 113.64\text{ bar} \qquad \text{corresponds to 1136.4° F}$$

$$\textbf{Actual value correction} = (0.00\text{ bar} - 3.20\text{ bar}) - \frac{113.64\text{ bar} \bullet 0\text{ mV}}{50\text{ mV}} = -3.20\text{ bar} \qquad \text{corresponds to 32.0° F}$$

2.4 Configuring Control Response

2.4.1 Controller Type

The **controller type** determines how system deviation is processed.

The type of manipulated variable output, i.e. the utilized actuators, depend upon the controller type.

This setting can be combined with all other configurations.

Controller Type	Processing
Unused (controller type 0)	This configuration is intended for unused channels. The actual value is only measured, without monitoring, error messages etc.
Measuring (controller type 1)	This configuration is intended for temperature monitoring. Limit value monitoring can be configured. System deviation is not used for any other purpose.
Actuator (controller type 2)	Same as **controller type 1** (*measuring*) In addition, the actuator manipulating factor is read out with the actuating cycle.
Limit transducer (controller type 3)	The maximum manipulating factor is read out, if the actual value is less than the momentary setpoint. The minimum manipulating factor is read out, if the actual value is greater than the momentary setpoint plus the dead zone. Switching hysteresis is adjustable, and status changes are possible after each actuating cycle. Actuation cycle time is used as a time constant for an additional input filter.
PDPI controller (controller type 4, 5)	The PDPI control algorithm assures short settling time without overshooting. The actuating cycle is at least as long as the selected value. The dead zone suppresses switching back and forth between heating and cooling if no lasting deviation occurs. The controller selects **controller type** *4* or *5* itself, the user can enter either. *Type 5* is a pure *PDPI step-action controller*, and *type 4* may include any other combination of actuators.
Proportional actuator (controller type 6)	The manipulated variable is proportional to system deviation, and a static dead zone can be adjusted at the cooling side. Actuation cycle time is used as a time constant for an additional input filter. The controller type is not intended for temperature regulation, because it does not demonstrate the dynamics required for control without overshooting.

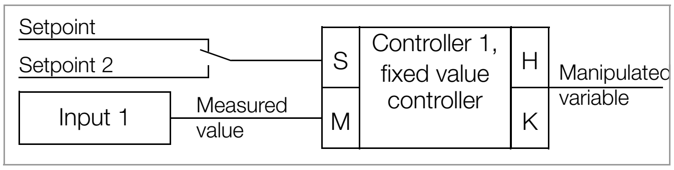
2.4.2 Controller Classes

The **controller class** determines how the controller processes input quantity actual and setpoint values.

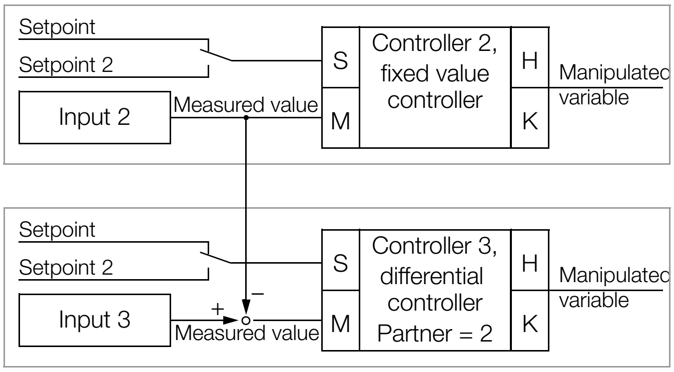
This setting can be combined with all other configurations.

Controller Class	Processing
Fixed setpoint controller (controller class 0)	System deviation equals setpoint value minus actual value
Differential controller (controller class 1)	Actual value difference is controlled, i.e. actual value of the differential controller channel minus the actual value of the partner channel Due to the sampling sequence, it is advisable to position the partner channel upstream from the differential controller channel in high speed circuits. Limit value monitoring is relative to actual value difference, and not the two actual values.
Master controller (controller class 2)	Due to the fact that inputs are not normally assigned to the master controller, it must be configured as such to assure calculation of a suitable manipulated variable for the slave controller. Control dynamics are attenuated to assure that the manipulated variable used as a delta actual value remains steady. Actuation cycle time is used as a time constant for an additional input filter. The manipulated variable is added by the slave controller as a delta setpoint value. 1% manipulated variable is always delta setpoint value 1° C (independent of unit of measurement selection °C or °F).
Slave controller (controller class 3)	The manipulated variable of the partner channel is added to the setpoint value, but only if the partner channel is a master controller. 1% manipulated variable is always delta setpoint value 1° C. Any possible setpoint shifting depends upon manipulated variable limiting at the master controller, and thus has a maximum value of ± 100° C. If switching occurs to the proxy setpoint, the channel becomes a *fixed setpoint controller*, in which case nothing is added to the proxy setpoint value. All functions which effect setpoint values, as well as setpoints ramps, setpoint limiting or actuation, are applied to the setpoint sum.
Switching Controller (controller class 4)	If a control loop has only one actuator and two sensors, and if the sensor to be used depends upon the operating state, switching can be executed by a switching controller in combination with a fixed setpoint controller used as a partner channel. **Configuration:** The channel to which the first sensor and the actuator are connected is configured as a fixed setpoint controller (controller class = 0). The channel to which the second sensor (and no actuator) is connected is configured as a switching controller (controller class = 4), and the channel to which the first sensor is connected is set up as a partner channel. If switching is to be triggered via a binary input, the corresponding input is assigned to the fixed setpoint controller with function selection = 4 (switching controller active). **Function:** As long as the “switching controller active” bit has not been set in the controller function of the fixed setpoint controller, the fixed setpoint controller with the first sensor is active, and the switching controller with the second sensor is inactive. If the “switching controller active” bit at the fixed setpoint controller is set, the fixed setpoint controller is inactive. The switching controller is active in this case and utilizes the setpoint of the fixed setpoint controller (including setpoint limits and the proxy setpoint), as well as its actuator outputs. The internal statuses of the respectively inactive controller are frozen in order to assure bumpless switching in both directions. The **controller on** bit for the controller function in the fixed setpoint controller is also used for the respective switching controller. The two related channels are thus always switched on and off together. The switching controller’s **controller on** bit cannot be changed. Limit value 1 is only monitored at the respectively active controller, and limit values 2 is always monitored at both.

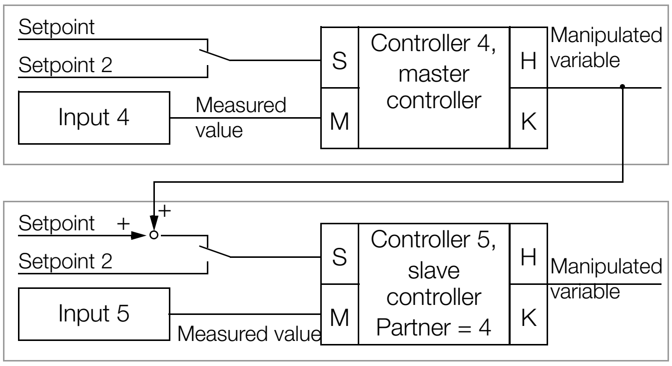
Fixed Value Control



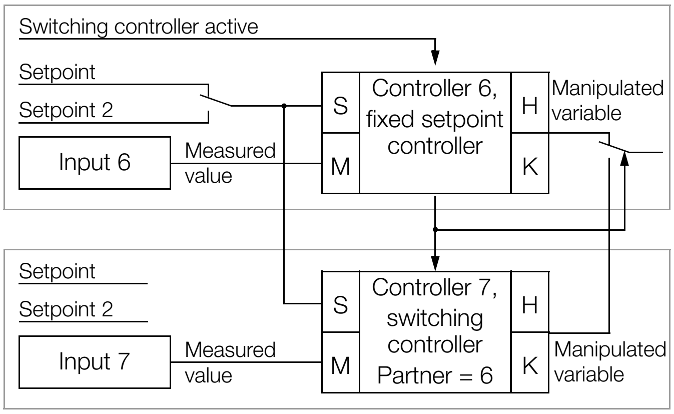
Differential Control



Cascade Control



Switching Control



2.5 Regulating the Control Functions

With the controller function byte, eight functions can be regulated via the interface or else via binary input. Assignment to groups is possible, so that several channels can be controlled simultaneously.

2.5.1 Assignment to Groups

Individual controller channels can be assigned to one group by setting **group** in the **controller configuration** to a valid *group number (from 0 to 3)*. In this way, channels assigned to a group can participate mutually in **actual value control**, selective changes to the **controller function** via binary input (see chapter 2.5.2 on page 14) or combining of channel-specific alarms into **group alarms** (see chapter 2.8.12 on page 24).

2.5.2 Setting Controller Functions via Binary Input

The bits included in **controller function** which are set via (bus) interface in order to activate individual functions, can also be set with the binary inputs. In this case, the binary input takes precedence over the interface. One input is required per function, and control can be executed per channel, for one group (1 to 3) or for all eight channels.

In the case of control per individual channel, the **output configuration** of the output is as follows:

Bit Number	Value	Meaning
0	1	Configuration as input
1	1	Control per individual channel
2 ... 4	0 ... 7	Channel number
5 ... 7	0 ... 7	Function selection

In the case of control per group, the **output configuration** of the output is as follows:

Bit Number	Value	Meaning
0	1	Configuration as input
1	0	Control per group
2, 3	0 / 1 ... 3	All 8 channels / group number
4 ... 6	0 ... 7	Function selection
7	0	—

Function selection:

Value	Meaning	Comment
0	Proxy setpoint active	see chapter 2.3.1
1	Actuation circuit	see chapter 2.6.1
2	Feed-forward control	see chapter 2.5.4
3	Temporary setpoint rise (Boost)	see chapter 2.6.2
4	Switching controller active	see chapter 2.4.2
5	Clear error	see chapter 2.7.1
6	Controller on	see chapter 2.8.1 and 7.4.3
7	Start self-tuning	see chapter 2.7.1

2.5.3 Manual Operation / Controller Off

The **controller on** bit in the **controller function** activates the controller channel (automatic operating mode). The controller outputs can then be driven in accordance with the controller's configuration.

If the controller channel has not been activated (**controller on** = *0*), output performance is determined by the **manual instead of off** bit in the **controller configuration**:

"**Manual instead of off**" not set:	Outputs are deactivated (off state). The integral-action component is cleared for PDPI controllers, i.e. the temperature must settle in once again when switched back on.
"**Manual instead of off**" set:	The last active manipulated variable continues to be read out and can be changed with the **manual manipulating factor** (manual operating mode). The integral-action component is not cleared for PDPI controllers, instead it is preset to the last (possibly changed) manipulated variable so that no jump occurs when switched back on. In this way, for example, the manipulated variable can be temporarily frozen, or another operating point can be approached in a bumpless fashion.

In the event that both states, **controller off** and **manual operation**, are required independent of each other, the **manual instead of boost** bit is set in the **extended controller configuration**; **manual instead of off** is not set.

The **controller on** and **boost** bits in the **controller function** control performance:

Controller on not set:	Outputs off
Controller on set and **boost** not set:	Automatic operation
Controller on and **boost** set:	Manual operation

2.5.4 Feed-Forward Control

Control quality can be significantly improved by means of feed-forward control where abrupt load fluctuations prevail when configured as a *PDPI controller*:

When the **feed-forward control** bit is set in **controller function**, the manipulating factor (integral-action component) of the controller is increased by a value equal to the **influencing quantity manipulating factor**, and when the **feed-forward control** bit is cleared, it is reduced by the same value.

Feed-forward control is inactive during self-optimization.

The **feed-forward control** bit is not (no longer) set after a device reset.

The feed-forward control is also active during manual operation or in the event of a sensor error.

Example:
If a machine requires an average of 70% heating power during production operation, but only 10% during idle time, the difference of the influencing quantity manipulating factor is set to 60%, and the **feed-forward control** bit is only activated during production.

2.6 Hot-runner Control

The manipulated value is read out in rapid cycles after setting the **hot-runner** bit in the **controller configuration**, i.e. actuation cycle time is 0.1 s regardless of the setting used for the **actuation cycle time** parameter.

The actuation circuit and temporary setpoint rise described below are even functional when the **hot-runner** bit is *not* set.

2.6.1 Actuating Circuit

The actuation circuit is enabled by setting the **actuation circuit** bit in the **controller function**.
The actuation circuit is only enabled for **controller type** *PDPI*. No actuation occurs for other controller types.

If the actuation bit is cleared, any currently active actuation operation is stopped immediately.

The actuation operation is started	if the actual value is more than 2° less than the **actuation setpoint** after auxiliary voltage is turned on (reset), or after the off state has been ended, or if the actual value drops to more than 40° less than the **actuation setpoint** after an actuation operation has been completed or during dwell time.
Actuation continues	until the actual value exceeds the **actuation value** minus 2° C. The manipulated variable is limited to the **actuation manipulating factor**. If the manipulated variable also needs to be read out as a rapidly pulsating signal, the channel must be configured as a **hot-runner** (**controller configuration**).
Dwell time then begins,	which is adjusted with **dwell time**. The controller regulates temperature to the actuation setpoint.
The actuation operation is ended	as soon as dwell time has expired. The controller then regulates temperature to the valid setpoint.

If the currently valid setpoint is still so far beneath the actuation setpoint that the condition for ending actuation cannot be fulfilled, the actuation operation continues indefinitely. In this case, manipulated variable limiting by means of **maximum manipulating factor** is advisable.

The corresponding bits in **controller status** indicate when actuation and dwell time are active.

2.6.2 Temporary Setpoint Rise (Boost)

Temporarily increasing the setpoint, for example in the hot-runner control mode, can be used to free clogged mold nozzles of “frozen” material remnants.

This procedure is triggered by bit 3 of the controller function, which is set via the interface or the binary input. The process is ended by clearing the bit, or automatically after maximum boost duration has elapsed.

The relative increase is stored per channel in the setpoint rise parameter (PI = 08), and maximum boost duration is stored in the boost duration parameter (PI = 09).

Boosting effects only the setpoint or the proxy setpoint, and not the actuation setpoint or the ramp function.

2.6.3 Actual Value Control, Synchronous Heat-Up

The objective is to reduce thermoelectromotive forces within the group by minimizing dynamic actual value differences.

The slowest control system within the group dictates setpoint rise for all other control systems within the group to this end. This is also possible for several devices. Selected setpoint ramps and the actuation circuit are taken into consideration.

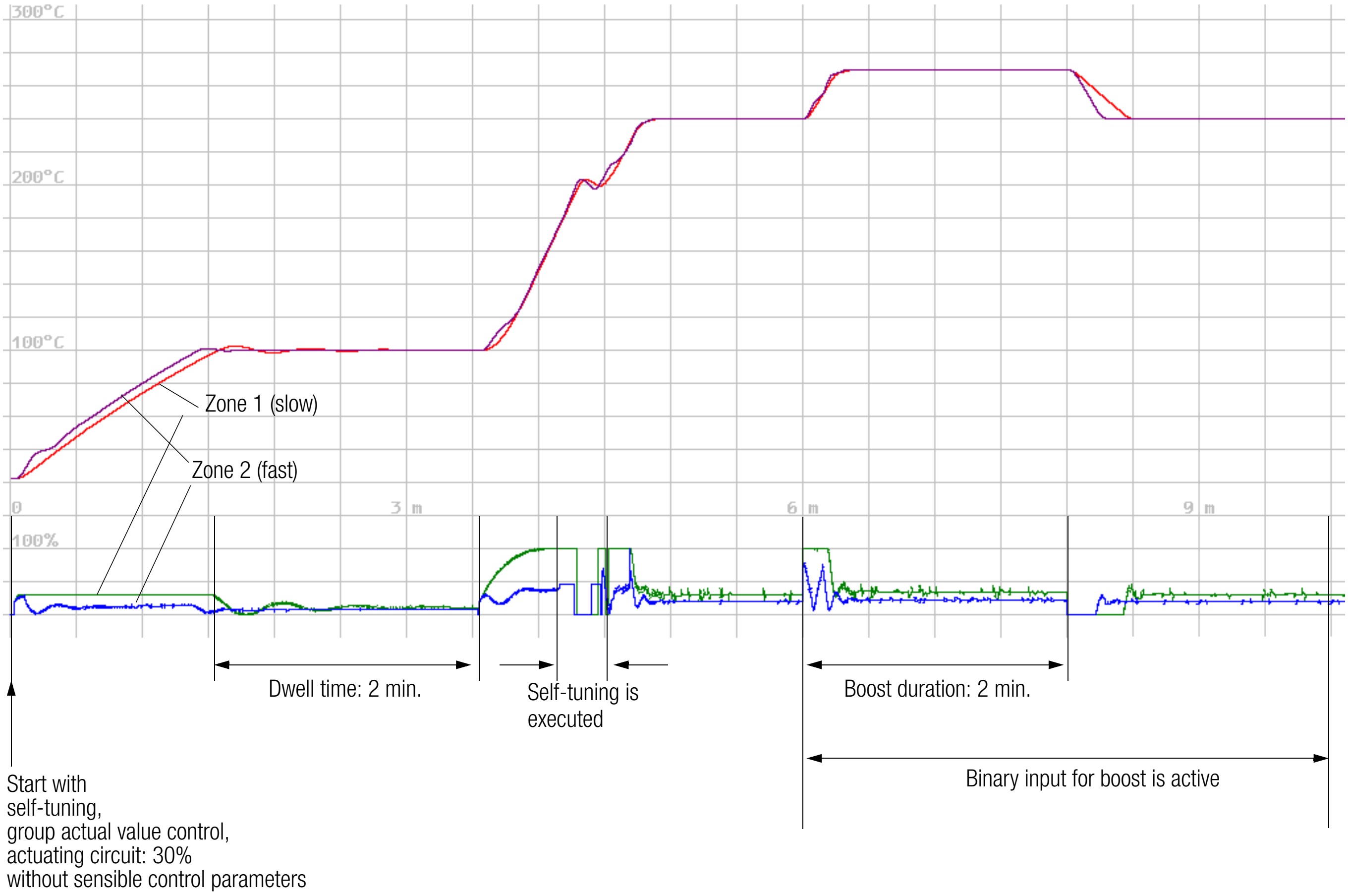
If the **actual value control** bit is set and assigned to a **group** (0 to 3) in the **controller configuration**, the channels which belong to the respective group participate in actual value control. **Controller type** must be set to **PDPI controller** for the participating channels to this end, and control must be activated, i.e. *controller on* or *self-tuning started* must be set in the controller function. In the **controller status** the corresponding bits indicate whether the actual value control is active and which channel is the slowest (compare chapter 7.4.6 on page 64).

The lowest actual value within the group, which can be made available to other devices as a **master actual value** via the bus, is determined. If the master actual value of another device in the same group is sent to the device, this is taken into consideration. In this way, many more than just eight channels can be heated up synchronously. All involved devices pass on their master actual values within a closed loop to this end, i.e. device 1 → device 2, device 2 → device 3, ... , last device → device 1. If CANopen is used, this process can be run automatically by means of PDOs. The bus master has to manage the process with all other bus types.

After all channels within the group have reached their setpoints, the master actual value is set to 1800° C in order to indicate this fact.

Control response relative to the master actual value varies depending upon whether or not the **hot-runner** bit is set in the **controller configuration**:

In the **hot-runner control** mode, the master actual value determines the setpoints of all channels within the group, such that temperature difference remains minimal. If self-tuning is started at the beginning of the actuation process, for example because another tool with yet unknown control parameters is started up, the zones with default parameters are used and the self-tuning sequence is influenced such that no large temperature differences occur during self-tuning.



If the hot-runner bit has not been set, the master actual value is not used for **two-step, three-step or continuous action control**. Instead, and ideal ramp gradient is determined for all channels within the group, so that all temperatures increase at the same rate. In this case, self-optimization does not take actual value management into consideration.

2.7 Determining Controller Parameters

Proportional zone heating and cooling (XpI / XpII) parameters, **system delay (Tu)** and **actuation cycle time** must be determined in order to obtain optimized controller dynamics.
Appropriate values for controller amplification, derivative-action time, integral-action time and the measured quantity sampling rate are generated based upon this data internally by the controller.

2.7.1 Self-Optimization (self-tuning)

Self-optimization is used to optimize controller dynamics, i.e. the parameters **proportional zone heating and cooling (XpI / XpII), delay (Tu)** and **actuation cycle time** are determined.

Preparation

- Complete configuration must be performed **before** self-optimization is started.
- The setpoint value is adjusted to the value which is required **after** optimization.
- If the self-tuning error bit for the channel error status is set, it must first be cleared.

Start

- Self-optimization is started by setting the **self-tuning on** bit in the **controller function,** provided that the **controller on** bit has also been set.
- The start command is accepted if **controller type** is set to *PDPI controller*,
  outputs are assigned to the channel and
  manipulated variable limiting is no less than 10%.

  If the start command is rejected, the **start error** bit is set for the **channel error status** of the corresponding channel (see also **events data**).
- Self-optimization remains activated even if the **self-tuning on** bit is cleared again.

Sequence

- The setpoint value which was active at the time self-optimization is started remains valid – changes are not effective at first (slave controllers: changing delta setpoints have no effect).
- Activation or deactivation of the proxy setpoint is not effective.
- Selected setpoint ramps are not taken into consideration.
- If started at the operating point (actual value approximates setpoint value), overshooting cannot be avoided.
- In the case of 3-step controllers, cooling is activated if the upper limit value is exceeded in order to prevent overheating. Self-optimization then performs a oscillation test around the setpoint.
- The bottom 4 bits in **controller status** indicate the optimization phase.
- The **self-tuning on** bit is reset after optimization has been completed.
- If self-optimization is started via the binary input, the binary input must be deactivated before self-optimization has been completed, because it would otherwise be restarted upon its completion. Self-optimization cannot be aborted via the binary input.

Abort

- Self-optimization can be aborted at any time by clearing the **controller on** bit.
- If an error occurs during self-optimization, the controller no longer reads out an actuating signal and the **self-tuning error** bit is set for the **channel error status** of the corresponding channel (in **events data**). This is the case in the event of a sensor error, or if the parameters configuration for the channel has been changed such that self-optimization is no longer sensible.
- In the event of an error, the **self-tuning error** bit of the channel error status must be cleared before closed loop control mode operation can be restarted.

2.7.2 Manual Optimization

The parameters **proportional zone heating and cooling, delay** and **cycle time** are determined by means of manual optimization. An actuation test or an oscillation test is performed to this end.

Preparing for the Actuation Test or the Oscillation Test

- Complete configuration must first be performed for use of the controller.
- The actuators are deactivated by setting **controller on** to *0* in **controller function**.
- A recorder must be connected to the sensor and adjusted appropriately for prevailing circuit dynamics and the setpoint. If the case of differential controllers, the actual value difference must be recorded.
- On and off-time of the heating output must be recorded for 3-step controllers (e.g. with an additional recorder channel or a stopwatch).
- Set **controller type** to *limit transducer.*
- Set **cycle time** to its minimum setting (*0.1 s*).
- If possible, deactivate any manipulating factor limiting.
- Reduce (or increase) the setpoint so that overshooting and undershooting do not cause any impermissible values.

Performing the Actuation Test

- Set **dead zone** to *MRS (measuring range span)* for 3-step controllers (cooling may not be triggered).
  Set **dead zone** to *0* for step-action controllers ("less output" must be triggered)
- Start the recorder.
- Activate the actuators by setting **controller on** to *1*.
- Record two overshoots and two undershoots. The actuation test is now complete for 2-step controllers.
  Continue as follows for 3-step controllers:
- Set **dead zone** to *0* in order to cause further overshooting with active cooling output. Record two overshoots and two undershoots.
- Record heating output on-time T_I and off-time T_{II} for the last oscillation.

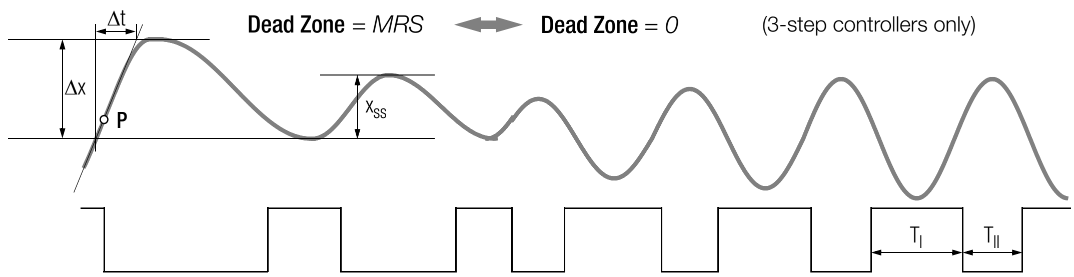


Figure 2 Characteristic Curve during Actuation Test

Evaluating the Actuation Test

- Apply a tangent to the curve at the intersection of the actual value and the setpoint, or the cut-off point of the output.
- Measure time Δt.
- Measure oscillation amplitude x_{SS}, or for step-action controllers overshooting Δx.

Parameter	Parameter Values			
	2-Step Controller	3-Step Controller	Continuous-Action Controller	Step-Action Controller [1)]
Delay (Tu)	1.5 • Δt			Δt – (**Ty** / 4)
Cycle time	**Tu** / 12			**Ty** / 100
Proportional zone heating (XpI)	x_{SS}		2 • x_{SS}	0.5 • Δx
Proportional zone cooling (XpII)	–	**XpI** • (T_I / T_{II})	–	–

[1)] **Ty = motor actuation time**

If manipulating factor limiting was active, the proportional zone must be corrected:

XpI multiply by 100% / **maximum manipulating factor**
XpII multiply by –100% / **minimum manipulating factor**

Performing the Oscillation Test

If an actuation test is not possible, for example if neighboring control loops influence the actual value too greatly, if cooling must be active in order to maintain the actual value (cooling operating point), or if optimization is required directly to the setpoint for any given reason, control parameters can be determined by means of sustained oscillation. However, calculated values for delay may be too large in this case under certain circumstances.

The test can be performed without a recorder if the actual value is observed and times are measured with a stopwatch.

- Set **dead zone** to *0*.
- Activate the actuators by setting **controller on** to *1*, and start the recorder if one is used. Record several oscillations until they become uniform in size.
- Measure oscillation amplitude x_{SS}.
- Record on-time T_I and off-time T_{II} of the heating output for the oscillations.

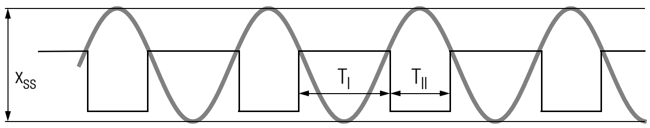


Figure 3 Oscillatory Characteristics

Evaluating the Oscillation Test

Parameter	Parameter Values			
	2-Step Controller	3-Step Controller	Continuous-Action Controller	Step-Action Controller [1]
Delay (Tu) [2]	$0.3 \bullet (T_I + T_{II})$			$0.2 \bullet (T_I + T_{II} - 2$**Ty**$)$
Cycle time	**Tu** / 12			**Ty** / 100
Proportional zone heating (XpI)	x_{SS}	$\frac{x_{SS} \bullet T_{II}}{(T_I + T_{II})}$	$2 \bullet x_{SS}$	$0.5 \bullet x_{SS}$
Proportional zone cooling (XpII)	—	**XpI** $\bullet (T_I / T_{II})$	—	—

[1] **Ty = motor actuation time**

[2] If either T_I or T_{II} is significantly greater than the other, value Tu is too large.

Correction for manipulating factor limiting:

XpI multiply by 100% / **maximum manipulating factor**

XpII multiply by –100% / **minimum manipulating factor**

Correction for step-action controllers in the event that T_I or T_{II} is smaller than **Ty**:

XpI multiply by $\frac{Ty \bullet Ty}{T_I \bullet T_I}$ if T_I is smallest, or by $\frac{Ty \bullet Ty}{T_{II} \bullet T_{II}}$, if T_{II} is smallest.

The value for **Tu** is very inaccurate in this case. It should be optimized in closed loop control mode.

Closed Loop Control Mode

The closed loop control mode is started after manual optimization has been completed:

- Set **controller type** to *PDPI*.
  – Adjust the setpoint to the required value.
  – The dead zone can be increased from **dead zone** = *0* for 3-step and step-action controllers, if control of the heating and cooling outputs, or more and less outputs, changes too rapidly due to an unsteady actual value.

2.8 Monitoring Functions

The results of individual monitoring functions are written to the **events data** bits, which can be queried via the (bus) interface, or read out selectively at the binary outputs.

2.8.1 Overview of Channel-Specific Alarms

These alarms are summarized for each channel in the channel error status word.

Bit no.	Meaning	Causes	Remedy	Channel Performance	Comment
0	Broken sensor	Interrupted cable	Inspect wiring and sensor	Depends upon configuration, e.g. read-out of sensor error manipulating factor	See chapter 2.8.7.
1	Polarity reversal	Polarity reversed at thermocouple or incorrectly connected Pt100			
2	2nd upper limit value exceeded	Temperature too high	Inspect the actuators Acknowledge alarm in event of alarm memory	No influence on control, except when configured as a limiter (see 2.8.4)	See chapter 2.8.3.
3	1st upper limit value exceeded				
4	1st lower limit value fallen short of	Temperature too low			
5	2nd lower limit value fallen short of				
6	Impermissible parameter	Transmitted parameter value out of limits Value has been rejected	Transmit plausible parameter value	No influence on control	Acknowledge alarm
7	Heating current not off with deactivated actuating signal	Short-circuited actuator	Inspect actuator and heating current circuit	No influence on control	See chapter 2.8.6.
8	Too little heating current with active actuating signal	Actuator interrupted / fuse blown			
9	Heating circuit error	Sensor does not measure correctly Heating current circuit interrupted	Inspect sensor, actuator and heating current circuit	No manipulated variable until error is acknowledged	See chapter 2.8.5. Acknowledge alarm
10	Self-tuning start-up error	Controller not activated Controller is configured incorrectly Controller cannot be self-tuned	Configure controller correctly	No influence on control	See chapter 2.7.1. Acknowledge alarm
11	Self-tuning error and abort	Sensor error has occurred Configuration has been changed during self-tuning		The channel is deactivated. Forced cooling until the error is acknowledged if an upper limit value has been exceeded	
12	Too big heating current with active actuating signal	Shunt circuit at actuator Current nominal value too little	Inspect actuator / heating current circuit Adjust current nominal value correctly	No influence on control	See chapter 2.8.6

2.8.2 Overview of Device-Specific Alarms

These alarms are summarized in the device error status word.

Bit no.	Meaning	Causes	Remedy	Device Performance	Comment
0	Analog error	Device is defective	Repair	All channels are deactivated	Error LED lights up
1	Overload, heating current 1	Secondary heating current greater than 1.2 A Interference voltage	Use a different transformer Transformer secondary must be potential-free	No influence on control	
2	Overload, heating current 2				
3	Overload, heating current 3				
4	Heating voltage overload	Secondary heating voltage greater than 60 V Interference voltage	Use a different transformer Transformer secondary must be potential-free		
6	Reference junction error	Wiring to the remote cold junction is interrupted or short-circuited	Inspect wiring	Control is continued with an assumed reference junction temperature of 30° C.	
		Defective reference junction	Replace reference junction		
7	EEPROM error	Implausible parameter values in memory	Restore default settings and reenter parameter values	All outputs are low	Error LED lights up See chapter 2.10. Acknowledge alarm
		Defective parameters memory	Repair		
8	Group output error	Inactive output has high level signal (> 14 V), or active output has low level signal (< 7 V)	Correct wiring error or short-circuit	Control is continued	Error LED lights up
		Output defective	Repair		
9	Mapping error	Sensor and heater assigned to different channels	Correct wiring or configuration	All manipulated variables off until the error is acknowledged	See chapter 2.9.4 Acknowledge alarm
10	Parameter error	Program sequence error	EMC measures	Parameter value is corrected from parameter value memory	
13	CRC error	Faulty parameter set DB (DB100) transmitted from CPU to controller	Download parameter set DB from controller or config tool into CPU	Parameter set has not been accepted by the controller	Acknowledge error

2.8.3 Limit Value Monitoring

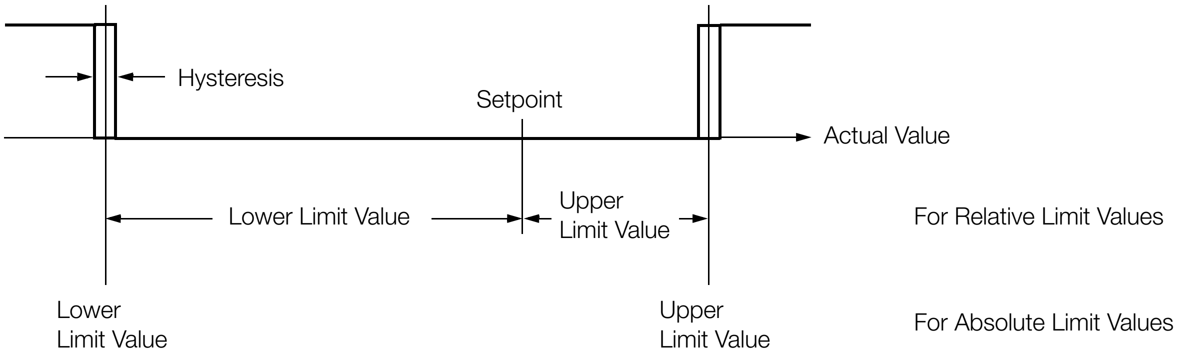


Figure 4 Schematic Representation of Limit Value Monitoring

Actuation Suppression

Alarm suppression is active during actuation (**actuation suppression** bit set in **limit value configuration**) until temperature has exceeded the lower limit value for the first time. During cooling, suppression is active until temperature has fallen below the upper limit value for the first time.
Suppression is active when auxiliary power is activated, if the momentary setpoint is changed or the proxy setpoint is activated, or if switching takes place from controller off to controller on.

Alarm Memory

If alarm memory is active (**alarm memory** bit set in **limit value configuration**), any bit which has been set in the **channel error status** remains set until it is cleared.

2.8.4 Limiter

If a controller needs to be deactivated in the event of a limit value violation within the control loop, the channel must be configured as a limiter. In this case, the controller responds just as it would if the "**controller on**" bit were not set in the **controller function** (PI = 20h). (Refer to chapter on manual operation 2.5.3)

The limiter can be combined with all **controller types** and **controller classes**.

- The limiter bit is set in the **limit value function** parameter (PI = 36h) in order to activate the **limiter** function.
- The limiter reacts to the **second limit values** (PI = 04h and 05h), which must be accordingly adjusted and configured. (See also chapter 2.8.3)
- As soon as a second limit value is violated, i.e. when either bit 2 or 5 is set in the **channel error status**, the controller is deactivated. If neither of these bits is set, the controller is reactivated.
- If the controller is to remain continuously deactivated after limit value monitoring has been triggered, the "**save alarm 2 active**" bit must be set in the **limit value function** parameter (PI = 36h).
- **Channel error status** bits 2 and 5 must then be cleared in order to reactivate the controller.
- This is also possible with a binary input by means of the **clear error** function (see also chapter 2.5.2).

2.8.5 Heating Circuit Monitoring

- Heating circuit monitoring is activated with the **heating circuit monitoring** bit in the **limit value configuration**.
- The controller must be configured as **controller type** *PDPI*, discontinuous or continuous heating with a **maximum manipulating range of** greater than or equal to *20%*.
- No monitoring takes place during self-optimization.
- The monitoring function utilizes the **delay Tu** and **proportional zone heating Xpl** control parameters, which must be correctly optimized for this reason. In the event of manual optimization or subsequent adaptation of control parameters, a lower limit for **Tu** must be maintained. The lower limit is:
  min. Tu = 2 • Xpl / (Δx / Δt) Δx / Δt = maximum temperature rise during actuation with 100% on-time.
  The limit is cut in half with continuous heating.
- An error message occurs at approximately 2 times Tu, if heating is discontinuous and the measured temperature increase is too small, or immediately if temperature plunges rapidly, as would not normally be possible.
  This may be caused by:
  – Polarity is reversed at the sensor, or the sensor is short-circuited.
  – No sensor is installed, the sensor has slipped out of place or has been installed at an incorrect position.
  – The heating current circuit is interrupted or has not been switched on.
  – The actuator is defective.
- In the event of error, the outputs are deactivated and the **heating circuit error** bit is set for **channel error status** (see also **events data**).
- The controller channel remains off until the **heating circuit error** bit is cleared.

2.8.6 Heating Current Monitoring

Connection

- 1 to 3 identical external summation current transformers can be connected (for all 8 channels simultaneously). The controller's current inputs are dimensioned for 1 A / 50 / 60 Hz.
  The current value which results in a secondary value of 1 A must be entered to the **summation current transformation ratio** parameter.
- A voltage transformer can be connected in order to compensate for heating voltage fluctuation.
- All channels are monitored whose current is fed through the transformer.

Parameters Configuration

- The current values (sum of phases 1 through 3) to be monitored must be entered to the **heating current nominal value** parameter for each monitored channel. Channels which are not monitored must be set to 0.0 A.
- The open-circuit voltage value which prevails at nominal primary heating voltage must be entered to the **heating voltage transformer secondary voltage** parameter in order to activate compensation. Compensation is deactivated if a value of less than 10.0 V is entered.
- Automatic adjustment of **nominal heating current values** and **secondary heating voltage** can be activated by setting the parameter **device control** (PI=32h) to 55h.

Write		Read		Meaning	
Bit Number	Code	Bit Number	Value		
0 ... 7	55h	4 ... 7	5h	Determination of heating current nominal values	start / running
	—		0h		finished

  Nominal heating current values are thus determined for all channels with a discontinuous-action heating output, and monitoring is thereby activated.
  If a value of 10.0 V or less is measured for secondary heating voltage, the value remains at 0.0 V and compensation is inactive. If a value of at least 10.0 V has already been selected for secondary heating voltage, no new value is determined for compensation.
  Normal control operation is interrupted by this measurement for approximately 1 second. Due to the fact that a currently running self-tuning process would be rendered useless in this case, the measurement is not performed as long as self-tuning is still active at any given channel or channels.

Function

- If heating current monitoring has been activated for at least one channel, the controller runs through a cycle of operating states (depending upon the **delay** Tu parameter) such that heating is only activated at one of the channels to be monitored (all other heaters are off), and all heaters are off. In this way, heating current can be measured at the individual channels with the summation current transformers. The measuring cycle is ideally adapted to paths when the **heating current sampling cycle** parameter is set to 0 = Auto.
- The measuring cycle can also be specified by setting the **heating current sampling cycle** parameter accordingly.
- If a secondary heating voltage value within a range of 10.0 V and 50.0 V has been selected, measured current values are compensated:

$$\text{monitored current} = \frac{\text{measured current} \bullet \text{secondary heating voltage}}{\text{measured voltage}}$$

  This allows for more accurate monitoring, for example in the case of parallel connected heaters.
- Monitoring and possible error messages take place with reference to the states:
  Heat off and current is present → Error: **Heating current not off**
  Heat on and too little current → Error: **Too little heating current**
  Heat on and too big current → Error: **Too big heating current**
- **Too little heating current** is indicated if the nominal heating current value is fallen short of by more than 20% with inactive heating voltage compensation, or if the nominal heating current value is fallen short of by more than 5% with active heating voltage compensation.
  The same limits apply when **heating current** is **too big**.

Monitoring 16/24 Channels

- Up to 3 devices can be connected via binary inputs and outputs such that all heating current for these 3 devices is monitored via heating current acquisition at the first device. This is advisable, for example, if only a small amount of heating current per device is being monitored.
- The monitoring function is limited as regards measuring technology, if the smallest heating current does not male up an appreciable portion (approx. 2%) of primary transformer current.
- The devices are connected via accordingly configured binary inputs and outputs in order to synchronize measurement:

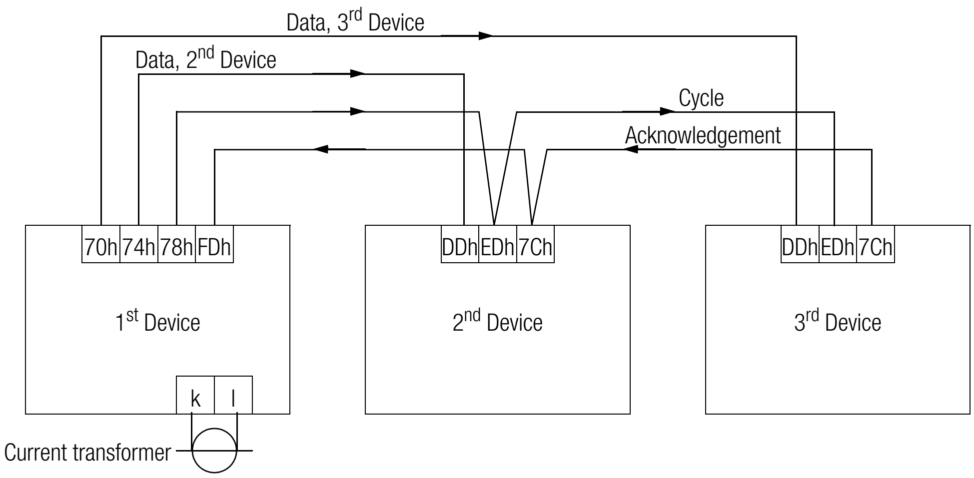


Figure 5 Wiring Diagram with Output Configuration Values

- Parameters for **heating current nominal values** at all 3 devices are configured at the 1st device. Automatic determination (see above) takes place at all 3 devices, if the binary inputs and outputs are correctly configured and connected. Any heating current parameter settings at the 2nd and 3rd devices are ignored.
- The **heating current sampling cycle** *must* be additionally set at the 1st device. The ideal value for rapid error detection is roughly half of delay time Tu, which means that the **heating current sampling cycle** should be set to the smallest value for 50% delay time of all monitored channels.
- The error message appears in the error status of the respective channel of the respective device.
- The error **too big heating current** is **not** registered for the 2nd and 3rd device.

2.8.7 Performance in the Event of Sensor Error

In the event of a broken sensor, thermocouple polarity reversal or short-circuiting of the Pt100, the **broken sensor** bit or the **polarity reversal** bit is set for **channel error status**.

The controller outputs respond as follows:

- No response occurs if **controller type** is set to *off*, *measure* or *actuator*.
- If **controller type** is set to *limit transducer*, *PDPI step-action controller* (controller type 5) or *proportional element*, the **sensor error manipulating factor** is read out in the automatic mode.
- If **controller type** is set to *PDPI controller* (=4) performance depends upon the selected **sensor error manipulating factor**:
  – Where **sensor error manipulating factor** = 0%, or minimum (–100%) or maximum (100%) manipulating factor:
  The sensor error manipulating factor is read out.
  – Where **sensor error manipulating factor** = any other value:
  After the control system settles in, a “plausible” manipulating factor is read out
  which keeps the temperature as close as possible to the setpoint.
  If the control system has not yet settled in (during start-up, after a reset), the sensor error manipulating factor is read out.
  If the controller is configured as a hot-runner controller, the “plausible” manipulating factor is averaged so that fluctuation throughout the injection cycle is averaged as well.

2.8.8 Monitoring the Binary Outputs

All binary outputs which have not been configured as inputs are monitored for short-circuiting and incorrect triggering.
2 times 24 bits are included in **output error**, which are set if the output is active although no signal is present at the terminal (short-circuit), or if the output is inactive and a signal is present at the terminal, i.e. the output is triggered due to a wiring error etc. Only the **operating current group error outputs** are not subject to output monitoring, so that they can be parallel connected amongst several devices.

2.8.9 Device Errors

Appropriate bits are set in **device error status** and the **error LED** at the front of the housing lights up if:

- Measured value acquirement is defective
- An error has been detected in the digital hardware
- An error in the parameters memory has been discovered
- An output monitoring error has occurred

Appropriate bits are also set if:

- Overload occurs at the heating current monitoring inputs
- The reference junction is interrupted or short-circuited

2.8.10 Clearing Error Bits

Several of the error bits in the **channel error status** and the **device error status** must be acknowledged because they are not cleared by the controller (except after a reset). This can be accomplished by overwriting the error status words via the interface as described in chapter 7.4.3.

The following bits in the **channel error status** can also be cleared via a binary input by adjusting the controller function selection setting to *clear errors* (see also chapter 2.5.2):

- Limit value error for alarm memory
- Heating circuit error
- Self-tuning start-up error
- Self-tuning error

Newly occurring errors are not suppressed.

The signal at the binary input must be applied for at least 100 ms.

2.8.11 Read-Out of Channel-Specific Alarms

Each channel has its own **channel error mask**, by means of which the errors to be read out via a binary output are selected from the **channel error status** (see chapter 7.4.7 on page 65 for details regarding error bits).

The **output configuration** of the selected output is set as follows for read-out:

Bit Number	Value	Meaning
0	0	Configuration as output
1	1	Single channel
2 ... 4	0 ... 7	Channel number
5	0	—
6	0 / 1	Operating current / closed-circuit current
7	1	Configuration as alarm output

2.8.12 Read-Out of Group Alarms or Self-Optimization Active Status

Eight **group error masks** can be programmed, by means of which the group errors are selected which are to be read out via a binary output (see chapter 7.4.8 on page 65 for details regarding error bits).

The **group alarms** are comprised of the channel-specific alarms by linking the alarms of all channels which belong to the same group by means of OR functions (see also chapter 2.5.1 on page 14).

The **output configuration** of the selected output is set as follows for the read-out of group alarms or the status indicating that self-optimization is still active or defective at some channel:

Bit Number	Value	Meaning
0	0	Configuration as output
1	0	Group error
2 ... 6	1 ... 8 9 10 ... 13	Group error 0 ... 7, Self-tuning in progress or self-tuning error Group error 0 ... 3
7	0 / 1	Operating current / closed-circuit current

2.9 Special Functions

2.9.1 Control of Binary Inputs and Outputs

The momentary **state of the binary inputs and outputs** can be read in at any time with PI = E0h (see chapter 7.10 on page 69).
The state may change every 10 ms, depening on the input and output configuration.

If the outputs are not required for a controller function, they can be configured as free inputs or outputs and are thus available for independent control functions.

For a **free input** the output configuration (PI = 37h) must be set to the value 81h so that no I/O error is indicated.

For a **free output** the output configuration (PI = 37h) must be set to the value 40h so that the output can be set to PI = E0h by writing (see chapter 7.10 on page 69). In this process, only those statuses are adopted that are associated with free outputs.

If only a maximum of eight free binary inputs are required, they may also be configured as message inputs (compare chapter 7.5.5 on page 67). The statuses can subsequently be queried by channel 9 in the **message word** as **controller status** (PI = 24h).

2.9.2 Control of Continuous Outputs

The **momentary state of the continuous outputs** can be read in at any time with PI = E1h (see chapter 7.10 on page 69). The value range from 0 ... 1000 corresponds to 0 ... 20 mA and/or 0 ... 10 V.

If individual continuous outputs are not required for a controller function, they can be configured as **free outputs** and are thus available for independent read-out.

The output configuration (PI = 37h) must be set to the value 40h for this purpose so that the output can be set to PI = E1h by writing (see chapter 7.10 on page 69). In this process, only those statuses are adopted that are associated with free outputs.

2.9.3 Data Logger

The data logger has enough capacity for 3600 sampled value pairs including actual values and manipulated variables for all 8 channels.

Recording is started over each time the device is reset, and data are lost if auxiliary power fails.

After memory has been filled to capacity with 3600 entries, the oldest values are deleted as new ones are recorded.

The **logger sampling cycle** (PI = 92h) can be configured within a range of 0.1 to 600.0 seconds. This results in recording times of 0.1 to 600 hours (6 minutes to 25 days).

In order to avoid overwriting any existing data, the recording can be stopped either via binary input (output configuration = CDh) or via interface (logger control (PI=93h) = 1).

The **number of samples** which can be read out can be queried with PI = 98h.

Actual values and manipulated variables are read out separately, and read-out is controlled by the **read-out starting point sampled values** (for actual values PI = 94h, for manipulated variables PI = 95h).

Read-out starting points can be envisioned as flags for a sampled actual value or manipulated variable, as of which the sampled values are read out during the next read operation. The very first sampling is flagged after a reset.

The respective read-out starting point indicates how many samplings are read from the recent past up to the current point in time. The read-out starting point is increased each time sampled values are saved.

The value cannot be greater than the number of samplings (PI = 98h).

Sampled values are read out with PI = 96h for actual values and PI = 97h for manipulated variables. Memory contents are not changed by read-out.

Each time an entry is read out via service interface or RS 485 bus (EN60870 or Modbus protocol), the read-out starting point value is automatically reduced so that the next read request accesses the next entry.

If read-out starting points are not manipulated via the interface, all sampled values can be picked up continuously and uninterruptedly by means of downloading at regular intervals (before old values are overwritten).

If sampled values are read out, the respective read-out starting point is automatically reduced such that the next read-out of sampled values occurs uninterruptedly, and without overlapping. Up to 120 values (15 samplings x 8 channels), or 8 x “read-out starting point” values can be requested.

With Profibus-DP and CANopen, a maximum of 8 words are read at once. The read-in starting point is not reduced automatically, but rather as a result of writing the value -1 to the read-in starting point.

The **time of the last sample** can be queried with PI = 99h.

Example:

- The logger sampling cycle is set to 10 seconds (PI = 92h: 100). This corresponds to a total recording time of 10 hours.
- Auxiliary voltage for the instrument was switched on about 3 hours ago, and no sampled values have yet been queried. Querying of the quantities “read-out starting point sampled actual values” (PI = 94h), “read-out starting point sampled manipulated variables” (PI = 95h) and “number of samplings” (PI = 98h) results in approximately 1080 = 3 x 60 x 60 / 10.
- Samplings for all 8 actual values over the last 15 minutes are now to be read out. The “read-out starting point sampled actual value” (PI = 94h) must be set to 90 = 15 x 60 / 10 to this end.
- The 90 x 8 sampled actual values can now be picked up.
- The “read-out starting point sampled actual value” (PI = 94h) is then reset to zero.
- The “read-out starting point sampled manipulated variable” (PI = 95h) remains unchanged.

2.9.4 Checking Sensor and Heater Assignments (mapping)

This function is used to check for correct wiring of the heater and the sensors.

Any included cooling system is not taken into consideration because the function is typically activated prior to initial heat-up, at which time the zones are cold.

Please note: This function is intended to provide assistance in testing, but it is not capable of preventing damage resulting from overheating due to incorrect wiring.
Independent monitoring of actual temperature is required under certain circumstances.

Preparation:

- **Controller type** must be set to *PDPI controller* for all control loops under test. Channels set to other options are not tested.
- The duration of testing for each individual channel depends upon the **delay** parameter. If the control parameters have already been optimized, the delay value need not be changed because it is already ideal. Otherwise, the delay should be set to approximately the time it takes for temperature in the respective zone to climb several degrees after the heat has been switched on.
- **Test time** is calculated for each channel based upon delay. Test time is twice as long as **delay**, or at least 10 seconds and not more than 5 minutes.

Attention!
If test time is too long, overheating may occur if a sensor cannot be assigned.
For example if the sensor is short-circuited, or if it is connected to another device.

Sequence:

- Checking sensor and heater assignments can be initialized from any state by transmitting AAh code for the **device control** parameter (PI = 32h).

Write		Read		Meaning
0 ... 7	AAh	4 ... 7	Ah	Monitoring of sensor/heating assignment start / running
	AAh		0h	stop / finished

- Testing is conducted during the initial phase (stabilization phase) to make sure that temperatures do not rise when all of the outputs of the channels under test are inactive. The duration of the stabilization phase is equal to the maximum test time value.
- Assignments are checked during the second phase for each individual channel, one after the other. Heat is switched on for the channel currently under test to this end, and all measured temperature values are observed which demonstrate a change of more than 5°. Any manipulating factor limiting, or the actuation circuit, is taken into consideration.
- Heat is switched off no later than after test time has elapsed, and the process is continued with the next channel.
- If no errors are detected, the controller returns to the selected operating mode after testing has been completed.
- If an error is detected, the **mapping error** bit is set for the **device error status**, and all heating and cooling outputs for all channels remain inactive until the mapping error bit is acknowledged.

Abort:

- Testing can be aborted at any time by transmitting the AAh code for the **device control** parameter.
- Testing is ended prematurely and the **mapping error** bit for the device error status is set, if the measured temperature value of any given channel rises excessively. The respective thresholds amount to 20° during the stabilizing phase and 50° during the second phase. Subsequent channels are then no longer tested.
- The same applies if temperature falls below the measuring range due to a sensor with polarity reversal.

Evaluation:

The results of the test appear in the **controller status** and in the **channel error status**:

- The **mapping address** for the **controller status** indicates the address of the sensor which has responded to the heater. The mapping address is only valid if the **mapping completed** bit for the **controller status** is set (see also chapter 7.4.6 on page 64).

The **mapping error** bit for the **device error status** is set in the event that any of the following errors occurs:

- The **mapping address** does not coincide with the channel number.
  Cause: Sensor or heater swapped, or excessive thermal coupling.
- If the **mapping completed** bit for the **controller status** is not set, even though the channel has been tested, no temperature change has been detected before test time elapsed.
  Cause: Test time was too short, i.e. a delay time has been selected which is too short, the heater is not active, the sensor is short-circuited or the sensor or the heater has been connected to another device.
- If a negative temperature change has been detected, the **polarity reversal** bit for the **channel error status** of the channel with the negative temperature change is set.
  Cause: sensor polarity reversed.
- If testing is ended prematurely due to an unexpected temperature rise, the **broken sensor** bit for the **channel error status** of the channel which demonstrated the temperature rise is set.
  Cause: The sensor is assigned to another device, the heater is being controlled by another device or excessive thermal coupling to the heater of another device prevails.
- The **broken sensor** and **polarity reversal** bits remain set until the mapping error is acknowledged.

2.9.5 Alarm History

Alarm history includes 100 error status entries with their respective time stamps.

Whenever at least one entire bit of the overall error status changes (compare PI = 21h or event data), the complete error status is saved with the current time stamp.

Recording is started over each time the device is reset, and data are lost if auxiliary power fails.

After memory has been filled to capacity with 100 entries, the oldest entry is deleted each time a new one is recorded.

The **number of entries** in alarm history can be queried with PI = 2Fh.

Read-out of alarm history entries is controlled with the **alarm history read-out starting point** value (PI = 2Dh).

This value specifies how many entries, from the past up to the current point in time, can be read out. The value cannot be greater than the number of entries (PI = 2Fh).

The read-out starting point can be envisioned as a flag for the entry which will be read out in response to the next read request.

The first entry is flagged after a reset, and the read-out starting point value is increased each time a new entry is saved to memory.

The **time stamp** is generated by a simple elapsed time meter, and not a real-time clock, i.e. the elapsed time meter starts again at zero after a device reset (1 January 00, 0:00 o'clock). In order to establish a relationship to real time, the current status of the elapsed time meter can be set to momentary time and date with PI = 90h.

Alarm history entries are read out with PI = 2Eh. Memory contents are not changed by read-out. The format of alarm history entries is described in chapter 7.4.9 on page 65.

Each time an entry is read out via service interface or RS 485 bus (EN60870 or Modbus protocol), the read-out starting point value is automatically reduced so that the next read request accesses the next entry.

Note: This is also the case even if not all 15 words are requested at once.

If the read-out starting point is not manipulated via the interface, all entries can be picked up continuously and uninterruptedly by means of read-out at regular intervals (before old values are overwritten).

Due to the fact that not all 15 words can be read at once with Profibus-DP and CANopen, the read-in starting point is not reduced automatically, but rather by writing the value -1 to the read-in starting point.

2.10 Parameter Sets

There are three parameter sets stored to non-volatile memory.
The device works with the momentary parameter sets, and only this set is effected by changes to individual parameters.

The two background parameter sets can be overwritten with the momentary parameter set, or loaded as the momentary parameter set. This allows for easy switching back and forth between two applications, and intermediate statuses can be saved during testing.

The default parameter set is stored to the firmware, and the controller can thus be reset to its default parameters at any time by overwriting the momentary parameter set.

Copying is controlled by means of the **device control** parameter (PI = 32h).

Bit Number	Value	Meaning	Comment
0 ... 7	0Fh 1Eh 1Fh 2Eh 2Fh	Load default settings to momentary parameter set Save momentary parameters to parameter set 1 Load parameter set 1 as momentary parameters Save momentary parameters to parameter set 2 Load parameter set 2 as momentary parameters	Cannot be read back

The copying procedure effects all parameters and configurations listed in the table on page 28,
except for the interface configurations (PI = A0h and A1h).

Overview of all Parameters and Configurations

The parameters listed below are saved to an EEPROM, and are not lost even in the event of mains power failure. Other quantities are stored to volatile RAM, or are permanently programmed. A complete list of all parameter indices (PI) is included in chapter 7.1 on page 59.

PI	Parameter Designation	U/M	Format	Setting Range	Default	Comment
Temperature Parameters						
00h	Setpoint	0.1°	± 15 bit	Minimum ... maximum setpoint	0.0° C	
01h	First upper limit value	0.1°	± 15 bit	0.0 ° = off, –MRS ... +MRS *)	0.0°	For Relative Limit Value
				0.0 ° = off, –MRS ... +MRS		For absolute LV and diff. controller
				0.0 °C / 32.0 °F = off, MRL ... MRU		For abs. LV and abs. value controller
02h	First lower limit value	0.1°	± 15 bit	Same as first upper limit value	0.0°	
03h	Proxy setpoint	0.1°	± 15 bit	Same as setpoint	0.0° C	
04h	Second upper limit value	0.1°	± 15 bit	Same as first upper limit value	0.0°	
05h	Second lower limit value	0.1°	± 15 bit	Same as first upper limit value	0.0°	
06h	Minimum setpoint	0.1°	± 15 bit	MRL ... maximum setpoint *)	0.0° C	For absolute value controller
				–MRS ... maximum setpoint		For differential controller
07h	Maximum setpoint	0.1°	± 15 bit	minimum setpoint ... MRU *)	600.0° C	For absolute value controller
				minimum setpoint ... MRS		For differential controller
08h	Setpoint rise (Boost)	0.1°	± 15 bit	–MRS ... +MRS	0.0 °	See chapter 2.5.3 on page 14
09h	Boost duration	0.1 s	± 15 bit	0.0 ... 3000.0 s	0.0 s	
0Ah	Actuation setpoint	0.1°	± 15 bit	Same as setpoint	0.0° C	See chapter 2.6.1 on page 15
0Bh	Dwell time	0.1 s	± 15 bit	0.0 ... 3000.0 s	0.0 s	
0Ch	Actual value correction	0.1°	± 15 bit	–MRS ... +MRS *)	0.0°	See chapter 2.6.3 on page 16
0Dh	Actual value factor	‰ / 0.1°	± 15 bits	10.0 ... 1800.0 ‰ / °C	100.0 %	See chapter 2.3.5 on page 10
0Eh	Setpoint ramp, up	0.1° / min.	± 15 bit	0.0 ° = off, 0.1 ° ... MRS *)	0.0	See chapter 2.3.1 on page 9
0Fh	Setpoint ramp, down	0.1° / min.	± 15 bit	0.0 ° = off, 0.1 ° ... MRS *)	0.0	
Control Parameters						
10h	Proportional zone heating	0.1°	± 15 bit	0.0 ° ... MRS *)	50.0 °	See chapter 2.7 on page 17
11h	Proportional zone cooling	0.1°	± 15 bit	0.0 ° ... MRS *)	50.0 °	
12h	Dead zone	0.1°	± 15 bit	0.0 ° ... MRS *)	0.0 °	Not for 2-step controllers
14h	System delay	0.1 s	± 15 bit	0.0 ... 3000.0 s	50.0 s	See chapter 2.7 on page 17
15h	Actuation cycle time	0.1 s	± 15 bit	0.1 ... 300.0 s	1.0 s	
16h	Actuator manipulating factor	%	± 7 bit	Min. ... max. manipulating factor	0%	
17h	Actuation manipulating factor	%	± 7 bit	Min. ... max. manipulating factor	100%	See chapter 2.6.1 on page 15
18h	Motor actuation time	0.1 s	± 15 bit	1.0 ...600.0 s	60.0 s	With step-action controllers
19h	Influencing quantity manipulating factor	%	± 7 bit	Min. ... max. manipulating factor	0%	See chapter 2.5.4 on page 15
1Ch	Minimum manipulating factor	%	± 7 bit	–100 ... 0%	–100%	Not with step-action controllers
1Dh	Maximum manipulating factor	%	± 7 bit	0 ... +100%	100%	Not with step-action controllers
1Eh	Sensor error manipulating factor	%	± 7 bit	Min. ... max. manipulating factor	0%	See chapter 2.8.7 on page 23
1Fh	Switching hysteresis	0.1°	± 15 bit	0.0 ° ... MRS *)	4.0 °	For limit value monitoring and limit transducers
Control Commands (further PIs are given in chapter 7.4 on page 62)						
20h	Controller function	bit	8 bit	See chapter 7.4.2 on page 62	0 = off	
22h	Controller configuration	bit	16 bit	See chapter 7.4.4 on page 64	1 = PDPI	
23h	Extended controller configuration	Bit	8 bit	See chapter 7.4.5 on page 64	0	
25h	Oscillation hold-off	0.1 s	8 bit	0.0 = off, 0.3 ... 25.0 s	0.0 s	See chapter 2.3.4
29h	Channel error mask	bit	16 bit	See chapter 7.4.7 on page 65	0 = none	See chapter 2.8.11 on page 24
2Ah	Group error mask	bit	16 bit	See chapter 7.4.8 on page 65	0 = none	See chapter 2.8.12 on page 24
Device Specification (further PIs are given in chapter 7.5 on page 66)						
32h	Device control	bit	8 bit	See chapter 7.5.3 on page 66	0 =° C	See chapter 2.10 on page 27
33h	Sensor type	–	8 bit	See chapter 7.5.2 on page 66	0 = type J	See chapter 2.1.1 on page 6
36h	Limit value configuration	bit	8 bit	See chapter 7.5.4 on page 66	0 = none	See chapter 2.8.3 on page 21
37h	Output configuration I/O 1 ... 16 continuous 1 ... 4	bit	8 bit	See chapter 7.5.5 on page 67	8-chan. 3-step	
3Ah	Power limitation	%	± 7 bit	0 ... +100 %	0 = off	See chapter 2.2.5
Heating Current Monitoring						
60h	Nominal heating current	0.1 A	± 15 bit	0.0 = off, 0.1 ... 1000.0 A	0 = off	See chapter 2.8.6 on page 22
61h	Heating current nominal value 2nd controller	0.1 A	± 15 bit	0.0 = off, 0.1 ... 250.0 A	0 = off	
62h	Heating current nominal value 3rd controller	0.1 A	± 15 bit	0.0 = off, 0.1 ... 250.0 A	0 = off	
64h	Summation current transformation ratio	0.1 A	± 15 bit	0.0 ... 1000.0 A	100.0 A	
67h	Heating current sampling cycle	0,1 s	± 15 bit	0.0 = auto, 0.1 ... 3000.0 s	0 = Auto	
69h	Secondary heating voltage	0.1 V	± 15 bit	0.0 = off, 10.0 ... 50.0 V	0 = off	
Data logger (further PIs are given in chapter 7.7 on page 68)						
92h	Logger sampling cycle	0.1 s	± 15 bit	0.1 ... 600.0 s	1.0 s	
Interface (not via Profibus)						
A0h	Interface configuration	bit	8 bit	See chapter 7.8.2 on page 69	2 = 19.2 kB	
A1h	CAN baud rate	bit	8 bit	See	4 = 125 kB	even P.

*) MRL = measuring range lower limit, MRU = measuring range upper limit, MRS = measuring range span

3 RS 232 Service Interface with Protocol per EN 60870

3.1 General

Interface connection is described in a separate set of installation instructions.

3.1.1 Interface Configuration

The controller is equipped with a serial interface with the following configuration:

- Modes RS 232 and RS 485 (2-wire)
- Baud rates 4800, 9600 and 19,200 (adjustable via interface)
- Format 8 data bit, 1 parity bit, 1 stop bit
- Parity even, odd, space or none (adjustable via interface)

Selection of a user address (0 ... 254) for RS 485 bus operation is accomplished with a DIP switch at the front panel. User address changes do not become effective until the device has been switched off, and then back on again.

3.1.2 Communication Protocol

The data transmission protocol per EN 60870 is used for communication between the field control level and the device level.
Only a sub-group of the functions defined by this protocol is utilized by the controller.

3.1.3 Primary Function

A master-slave protocol is used with a permanently assigned master (master computer) and up to 255 slaves (devices).
Communication takes place in the half-duplex operating mode, i.e. a device connected to the master computer only becomes active (responds):

- If it receives a valid frame addressed to itself
- If the specified maximum response delay time (t rd) has expired, allowing the master computer enough time to become ready to receive

The master computer may not become active again until:

- It receives a valid response frame from the addressed device and the specified waiting period after completion of the response frame (t rw) has expired
- The specified maximum response delay time (t rd) has expired
- The specified character delay time has expired (t cdt = pause between 2 character transmissions). This waiting time also applies for the receipt of invalid and incomplete responses!

3.1.4 Time Response

Ready to transmit/receive after power-up	t rdy	approx. 5 s
Character delay time (instrument)	t cdt	< 3 ms
Character delay time (master)	t cdm	< 100 ms
Response delay time (instrument)	t rd	< 10 ... 100 ms
Query waiting time after response (master)	t rw	> 10 ms

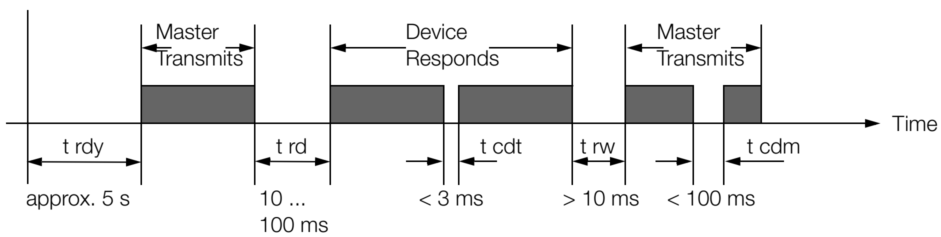


Figure 6 Basic Time Response

3.2 Frame Types and Layout

In both the query and the response direction, all frames consist of one of three string types which differ in their basic structure. Their use is required for all interface functions made available by the controller, and they are described as follows.

3.2.1 Short Strings

Short strings are used

by the querying device:

- For the transmission of short commands to devices (e.g. “reset” etc.)
- For abbreviated querying of important data from the devices (e.g. events data etc.)

by the responding device:

- To acknowledge queries which do not require response

Basic Short String Layout

Character No.	Content	Meaning		Comment
1	10h	Start message character	(SMC)	
2		Function field	(FF)	See chapter 3.2.4 on page 31
3		Device address	(DA)	
4		Checksum	(CS)	See chapter 3.2.4 on page 31
5	16h	End message character	(EC)	

3.2.2 Control Strings

Control strings are only used by the querying device. They are used to query all devices which cannot be queried with short strings, because they require a complete specification.

Basic Control String Layout

Character No.	Content	Meaning		Comment
1	68h	Start message character	(SC1)	
2		Length	(L1)	Number of characters from function field up to but not including the checksum
3		Length (repetition)	(L2)	
4	68h	Start message character (repetition)	(SC2)	
5		Function field	(FF)	See chapter 3.2.4 on page 31
6		Device address	(DA)	
7		Parameters index	(PI)	See chapter 3.2.4 on page 31
8		From channel	(fC)	See chapter 3.2.4 on page 31 These characters are not included in some parameters indices from main group 3.
9		To channel	(tC)	
10	00h	Recipe number	(RN)	
8 or 11		Checksum	(CS)	See chapter 3.2.4 on page 31
9 or 12	16h	End message character	(EC)	

3.2.3 Long String

Long strings are used:

- to transmit commands and parameters to a device
- to receive data and parameters from a device

Basic Long String Layout

Character No.	Content	Meaning		Comment	
1	68h	Start message character	(SC1)		
2		Length without SC1, L1, L2, SC2, CS, EC	(L1)	Number of characters from function field up to but not including the checksum	
3		Length (repetition)	(L2)		
4	68h	Start message character (repetition)	(SC2)		
5		Function field	(FF)	See chapter 3.2.4 on page 31	
6		Device address	(DA)		
7		Parameters index	(PI)	See chapter 3.2.4 on page 31	Not included in response cycle data and events data
8		From channel	(fC)	See chapter 3.2.4 on page 31 These characters are not included in some parameters indices from main group 3.	
9		To channel	(tC)		
10	00h	Recipe number	(RN)		
		n characters of user data		See chapter 3.2.4 on page 31	
L1 + 5		Checksum	(CS)		
L1 + 6	16h	End message character	(EC)		

3.2.4 Format Character Function and Value Range

Device Address (DA)

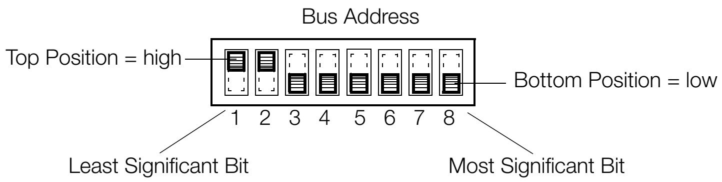


Figure 7 Example: Bus Address = 3

- 0 ... 254 Range of individual device addresses, set by means of DIP switches at the front of the housing.
- 255 This address can be used to contact all devices connected to the bus simultaneously. Data and commands transmitted to this address are accepted by all devices, and no acknowledgement is transmitted to the master.

Length (L1, L2)

Length entries L1 = L2 make reference to the number of characters from the function field (FF) up to but not including the checksum (CS), and are used in control strings and long strings. L1 and L2 are independent of the utilization of fC, tC and RN, and the number (n) of user data characters.

Correspondingly, L1 and L2 have a

- value of 3 or six in control strings and
- a value of n + 3 or n + 6 in long strings.

Function Field (FF)

The function field includes

- actual user information in short strings – its function is predefined bit by bit, and is different in the query and response directions,
- direction and control information for transmitted user data in control strings and long strings.

Function Coding for the Function Field in the Query Direction

Query Control	Code	String	Comment
Standardize data layer link	40h	Short string	Only the indicated codes are evaluated, invalid codes are responded to with an error acknowledgement.
Reset device	44h		
Query: “device OK?”	49h		
Request events data	7Ah		
Request cycle data	7Bh		
Request heating currents	7Eh		
Transmit data to controller	73h	Long string	
Request data from controller	7Bh		

Function Coding for the Function Field in the Response Direction

Bit no.	Function	Value	Meaning	
0 ... 3	Response	0 1 B	ACK: positive acknowledgement NACK: negative acknowledgement; message not accepted Response to: “device OK?”	Short string
		8	Transmit data	Long string
4	Job acknowledgement	0 1	Job executed, device ready Device not ready for this job, repeat job if required	
5	Service request	0 1	No error Error occurred (query events data)	
6	Direction bit	0		
7	—	0		

Parameters Index (PI)

The type of data to be transmitted is determined with the parameters index. The “PI” character is interpreted as follows:

Bits 7 ... 4	Bits 3 ... 0
0 ... Fh	0 ... Fh
Selection number for main parameters group	Selection number for special parameters

Functionally related data and setting parameters for a given device are included in the main parameters groups. Only those parameters indices which are documented in chapter 7 on page 59 can be accessed, all others are acknowledged with an error message.

Channel and Recipe Selection (fC, tC, RN)

Due to the fact that the controller is a multi-channel device, the entries

“from channel” fC

“to channel” tC

are used to determine which channels will transmit the requested data. The entry fC = 0 and tC = 0 indicates that all channels will be used.

Data can be requested from various parameter sets with the recipe number (RN). The controller includes only one recipe (RN = 0).

Checksum (CS)

The checksum consists of a byte-by-byte summation (without overflow summation) including all characters from the function field (FF), up to but not including the checksum (CS) for all string types.

Example: short string: CS = FF + DA

Length and Structure of User Data Blocks

Length and structure are variable, and depend upon PI, fC and tC.

Transmitted values may be structured according to bytes or words. the following formats are used:

±7 bit	Representation as 2 part compliment	Number with plus or minus sign
±15 bit	LS byte first, representation as 2 part compliment	Number with plus or minus sign
8 bit	LS byte first	Bit field

3.2.5 Criteria for the Validity of a Query Frame

If criteria are fulfilled, the controller responds with the requested data:

- No parity error in the query frame or in the response frames of other bus users.
- For short strings:

Character	Content	Meaning	Comment
1	10h	SMC	
2	 40h 44h 49h 7Ah 7Bh 7Eh	FF	Valid function coding: Standardize data layer link Reset Device OK? Event Cycle Heating currents
3	0 ... 255	DA	
4	(DA) + (FF)	CS	
5	16h	EC	

- For control strings and long strings:

Character	Content	Meaning	Comment
1	68h	SC1	
2		L1	
3	L1	L2	
4	68h	SC2	
5	73h 7Bh	FF	Write Read
6	0 ... 255	DA	Interface address
7		PI	Valid value
...		Data	
L1 + 5th character		CS	Sum of FF up to and including data
L1 + 6th character	16h	EC	

Exceptions, no response in the event of:

- Reset short string
- DA = 255 (broadcast address)

If incorrect FF, PI or CS data are received by the master computer, the controller responds with a short string with negative acknowledgement NACK.

If an error occurs at the controller (any bit set for device error or channel error), the controller responds with a short string in which the service request bit is set.

3.3 Frame Contents

3.3.1 Reset Device

The addressed device performs a hardware reset (same as for brief interruption of auxiliary power).

Example: device address = 2

Command (short string):

Character No.	Content	Meaning
1	10h	SMC
2	44h	FF (reset device)
3	02h	DA
4	46h	CS
5	16h	EC

Response:

	None, because reset is executed

3.3.2 Query: Device OK?

The addressed device transmits the function field only.

Example: device address = 3

Command (short string):

Character No.	Content	Meaning
1	10h	SMC
2	49h	FF (device OK?)
3	03h	DA
4	4Ch	CS
5	16h	EC

Response (short string):

Character No.	Content	Meaning
1	10h	SMC
2	0Bh	FF (e.g. no error occurred)
3	03h	DA
4	0Eh	CS
5	16h	EC

3.3.3 Cycle Data

The most important controller measurement and evaluation data are contained in a single data packet. Cyclical queries for these values are thus possible in compact form (short string command query).

Example: device address 3

Command (short string):

Character No.	Content	Meaning
1	10h	SC
2	7Bh	FF
3	03h	DA
4	7Eh	CS
5	16h	EC

Response (long string):

Character No.	Content	Meaning	U/M	Format	Comment
1	69h	SC1			
2	2Ch	L1			Number of characters from character 5 ... 48
3	2Ch	L2			
4	68h	SC2			
5	08h	FF			(e.g. no error)
6	03h	DA			
7, 8			0.1 °	± 15 bit	Momentary controlled variable, channel 1
...			0.1 °	...	...
21, 22			0.1 °	± 15 bit	Momentary controlled variable, channel 8
23			%	± 7 bit	Momentary manipulated variable, channel 1
...			%	...	...
30			%	± 7 bit	Momentary manipulated variable, channel 8
31, 32			0.1 A	± 15 bit	Momentary heating current, channel 1
...			0.1 A	...	...
45, 46			0.1 A	± 15 bit	Momentary heating current, channel 8
47, 48			0.1 V	± 15 bit	Momentary heating voltage
49		CS			
50	16h	EC			

3.3.4 Heating Current Data

These data include heating current for the 2nd and 3rd controllers in one data packet (see also chapter 2.8.6 on page 22, monitoring 16/24 channels).

Example: device address 3

Command (short string):

Character No.	Content	Meaning
1	10h	SZ
2	7Eh	FF
3	03h	GA
4	81h	PS
5	16h	EZ

Response (long string):

Character No.	Content	Meaning	U/M	Format	Comment
1	69h	SZ1			
2	22h	L1			Number of characters from character 5 ... 38
3	22h	L2			
4	68h	SZ2			
5	08h	FF			(e.g. no error)
6	03h	GA			
7, 8			0.1 A	± 15 bit	Momentary heating current channel 1, 2nd controller
...			0.1 A	...	...
21, 22			0.1 A	± 15 bit	Momentary heating current channel 8, 2nd controller
23, 24			0.1 A	± 15 bit	Momentary heating current channel 1, 3rd controller
...			0.1 A	...	...
37, 38			0.1 A	± 15 bit	Momentary heating current channel 8, 3rd controller
39		PS			
40	16h	EZ			

3.3.5 Events Data

Events data include all error messages and alarms generated by the device. They can be queried by means of a short string for the identification of a specific error alarm, for example if the BA bit (group error) was previously set in the function field (FF) of any given response frame.

Example: device address 3:

Command (short string):

Character No.	Content	Meaning
1	10h	SC
2	7Ah	FF
3	03h	DA
4	7Dh	CS
5	16h	EC

Response (long string):

Character No.	Content	Meaning	U/M	Format	Comment
1	68h	SC1			
2	1Ah	L1			Number of characters from character 5 ... 30
3	1Ah	L2			
4	68h	SC2			
5	28h	FF			(e.g. bit 6 = 1, one or several errors)
6	03h	DA			
7, 8			Bit	16 bit	Error status, channel 1
...			Bit	...	...
21. 22			Bit	16 bit	Error status, channel 8
23, 24			Bit	16 bit	Device error status
25			Bit	8 bit	Output error 1
...			Bit	...	...
30			Bit	8 bit	Output error 6
31		CS			
32	16h	EC			

Bit assignments for the error status word and the output error are described in chapter 7.4.3 on page 63.

3.3.6 Requesting Data from the Controller

All values, parameters, configurations, statuses, device IDs etc. can be queried with this type of communication.

Data are queried individually per parameters index. A complete list of all parameters indices is included in chapter 7 on page 59.

Querying a Device Specification

The parameters index is in main group 3. The characters “from / to channel” and “recipe number” are thus omitted for some parameters indices in control strings and long strings.

Example: Read device characteristic from device no. 3

Query (control string without fC, tC, RN):

Character No.	Content	Meaning
1	68h	SC1
2	03h	L1
3	03h	L2
4	68h	SC2
5	7Bh	FF (e.g. = 7Bh: read data)
6	03h	DA (e.g. = 3)
7	31h	PI (e.g. = 31h: device characteristic)
8	AFh	CS
9	16h	EC

Response (long string without fC, tC, RN):

Character No.	Content	Meaning
1	68h	SC1
2	04h	L1
3	04h	L2
4	68h	SC2
5	08h	FF (e.g. = 08h: no error)
6	03h	DA
7	31h	PI
8	08h	Device characteristic = 08h
9	44h	CS
10	16h	EC

Requesting, for Example, a Control Parameter

The parameters index is not part of main group 3, and the characters “from / to channel” and “recipe number” are thus included in control strings and long strings.

Example: Read sensor error manipulating factor from device no. 3, channel 1, value = 20%

Command (control string):

Character No.	Content	Meaning
1	68h	SC1
2	06h	L1
3	06h	L2
4	68h	SC2
5	7Bh	FF (e.g. = 7Bh: read)
6	03h	DA (e.g. = 3)
7	1Eh	PI (e.g. = 1Eh: sensor error manipulating factor)
8	01h	fC
9	01h	tC
10	00h	RN
11	9Eh	CS
12	16h	EC

Response (long string):

Character No.	Content	Meaning
1	68h	SC1
2	07h	L1
3	07h	L2
4	68h	SC2
5	08h	FF (e.g. = 08h: = no error)
6	03h	DA (e.g. = 3)
7	1Eh	PI (e.g. = 1Eh: sensor error manipulating factor)
8	01h	fC
9	01h	tC
10	00h	RN
11	14h	Information field where n = 1 character
12	3Fh	CS
13	16h	EC

3.3.7 Transmitting Data to the Controller

All parameters, configurations and operating states can be set with this type of communication. Data are queried individually per parameters index.

A complete list of all parameters indices is included in chapter 7 on page 59.

The setting range of the transmitted value is checked by the controller. If the value is not within the allowable range, it is not saved to memory. In the event of an error, the “parameter error” bit is set, and the “service request” bit is set in the function field of the acknowledgement short string.

Complete configuration must be performed before parameters are set, because the configuration effects usage and the setting ranges of individual “temperature parameters”.

Transmitting a Device Specification

The parameters index is in main group 3. The characters “from / to channel” and “recipe number” are thus omitted for some parameters indices in control strings and long strings.

Example: Set controlled variable quantity at device no. 3 to °F

Command (long string):

Character No.	Content	Meaning
1	68h	SC1
2	04h	L1
3	04h	L2
4	68h	SC2
5	73h	FF (read data)
6	03h	DA (= 3)
7	32h	PI
8	01h	Value
9	A9h	CS
10	16h	EC

Response (short string):

Character No.	Content	Meaning
1	10h	SMC
2	00h	FF (no error)
3	03h	DA
4	03h	CS
5	16h	EC

Transmitting, for Example, a Temperature Parameter

The parameters index (PI) is not part of main group 3, and the characters “from / to channel” and “recipe number” are thus included in long strings.

Example: Transmit setpoint = 25.0° to device no. 3, channel 3

Command (long string):

Character No.	Content	Meaning
1	68h	SC1
2	08h	L1
3	08h	L2
4	68h	SC2
5	73h	FF (e.g. = 73h: transmit data)
6	03h	DA (e.g. = 3)
7	00h	PI (e.g. = 00h: setpoint)
8	03h	fC
9	03h	tC
10	00h	RN
11, 12	FAh, 00h	Information field where n = 2 characters, format: ± 15 bit, LSB first
13	72h	CS
14	16h	EC

Response (short string):

Character No.	Content	Meaning
1	10h	SC
2	10h	FF (e.g. device not ready for job)
3	03h	DA
4	13h	CS
5	16h	EC

4 Modbus Interface

4.1 General

Interface connection is described in a separate set of installation instructions.

4.1.1 Interface Configuration

The controller is equipped with a serial interface with the following configuration:

- Modes RS 232 and RS 485 (2-wire)
- Baud rates 4800, 9600 and 19,200 (adjustable via interface)
- Format 8 data bits, 1 parity bit, 1 stop bit
- Parity even, odd, space or none (adjustable via interface)

4.1.2 Communication Protocol

The Modbus protocol is used for communication between the field control level and the device level.
The RTU mode and conformity class 0 (read and write words) are utilized.

4.1.3 Primary Function

A master-slave protocol is used with a permanently assigned master (master computer) and up to 255 slaves (devices).
Communication takes place in the half-duplex operating mode, i.e. a device connected to the master computer only becomes active (i.e. responds):

- If it receives a valid frame addressed to itself
- If the specified maximum response delay time (t rd) has expired, allowing the master computer enough time to become ready to receive

The master computer may not become active again until:

- It receives a valid response frame from the addressed device and the specified waiting period after completion of the response frame (t rw) has expired
- The specified maximum response delay time (t rd) has expired
- The specified character delay time has expired (t cdt = pause between 2 character transmissions). This waiting time also applies for the receipt of invalid and incomplete responses!

4.1.4 Time Response

Ready to transmit / receive after power-up	t rdy	approx. 5 s	
Character delay time (instrument)	t cdt	< 3.5 t ch	(2 ms at 19.2 kbd)
Character delay time (master)	t cdm	< 3.5 t ch	(2 ms at 19.2 kbd)
Response delay time (instrument)	t rd	< 10 ... 100 ms	
Query waiting time after response (master)	t rw	> 10 ms	

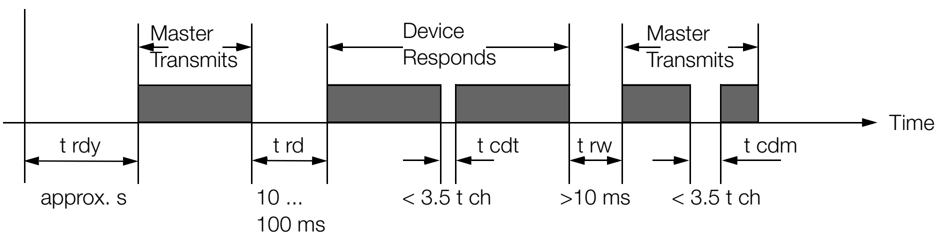


Figure 8 Basic Time Response

Character time = time for transmitting one character t ch 0.57 ms at 19.2 kbd

4.2 Frame Types and Layout

4.2.1 Basic Layout

Number of Characters	Meaning	Comment
1	Slave address (0 to 255)	Device address (not 0) 0 = to all (only where function code = 5, 16)
1	Function Code	See chapter 4.2.3 on page 40
n	Data	See chapter 4.2.4 on page 40 and chapter 4.2.6 on page 41
1	Error check (CRC-16) low byte	See chapter 4.2.5 on page 40
1	Error check (CRC-16) high byte	
(4)	Waiting time, no characters are transmitted	See chapter 4.2.2 on page 40

4.2.2 Waiting Time

- Waiting time is equal to the time it would take to transmit 4 characters.
- Waiting time serves to delineate the beginning and the end of the frame, because no explicit length specification is included in the frame.
- A frame is considered finished when waiting time has expired.
- If, for any reason, transmission of a frame is interrupted for a period which exceeds waiting time, the frame is considered finished. The first character after the interruption is interpreted as the first character of a new frame (both parts of the frame are rejected due to error check failures for this reason).

4.2.3 Function Code

The following function codes (FC) are supported:

Function Code	Meaning	Application
3	Read words	For reading values and parameters
5	Write single bit	Only for resetting the instrument
7	Read status	Query: "Device OK?"
16	Write words	For writing parameters

4.2.4 Data

Refer to chapter 4.2.6 on page 41 and chapter 4.3 on page 44 for details concerning the data field in the frame.

- Data used with Modbus are always 16 bit words.
  The high byte is transmitted first.
- Numeric values are represented as compliments of 2.
- Quantities with a ±7 bit format are expanded to ±15 bit.
- Bit fields in 8 bit format are expanded with a high byte = 0.

4.2.5 Error Check

Correct transmission of the frame is assured by means of the CRC-16 cyclical redundancy check. Both CRC-16 characters are generated as follows, based upon all of the characters included in the frame (slave address to last data byte):

1. Presetting of a 16 bit register (CRC-16 register) with FFFFh
2. Exclusive OR linking of the low bytes in the CRC-16 register to the frame's character,
   results to CRC-16 register
3. Shift the CRC-16 register one bit to the right,
   A0 is added and the displaced, least significant bit (LSB) is saved
4. Where LSB = 0, continue as of step 5.
   Where LSB = 1, establish exclusive OR linking of the CRC-16 registers to A001h.
5. Repeat steps 3 and 4 until a total of 8 shifts to the right have occurred.
   At this point, one of the frame's characters has been processed.
6. Execute steps 2 through 5 for each of the frame's remaining characters.
7. The content of the CRC-16 register, preceded by the low byte, is added to the frame after all of the frame's characters have been processed.

For example, programming in C would result in the following code:

```
/* -----------------------------------------------------------------
crc_16()                      calculate the crc_16 error check field
Input parameters:    buffer:   string to calculate CRC
                     length:   bytes number of the string
Return value:                  CRC value.
---------------------------------------------------------------- */
unsigned int crc_16 (unsigned char *buffer, unsigned int length) {
   unsigned int i, j, lsb, tmp, crc = 0xFFFF;
   for ( i = 0; i < length; i++ ) {
      tmp = (unsigned char) *buffer++;
      crc ^= tmp;
      for ( j = 0; j < 8; j++ ) {
         lsb = crc & 0x0001;
         crc >>= 1;
         if ( lsb != 0 ) crc ^= 0xA001;
      }
   }
   return (crc);
}
```

4.2.6 Support Frames

Read Words (FC = 3)

Query from Master:

Character No.	Meaning
1	Slave address (not 0)
2	FC = 3
3	Word address (high byte)
4	Word address (low byte)
5	Number of words (high byte)
6	Number of words (low byte)
7	CRC-16 (low byte)
8	CRC-16 (high byte)

Response from Slave:

Character No.	Meaning
1	Slave address
2	FC = 3
3	Number of characters (n)
4	Word data (n/2 words)
...	Respective high byte first
...	...
4 + n	CRC-16 (low byte)
5 + n	CRC-16 (high byte)

If the word address does not exist in the controller, or if the number of words is too great, the controller transmits an “error response” with corresponding error code (see also chapter 4.2.7 on page 43).

Reset (FC = 5)

Query from Master:

Character No.	Meaning
1	Slave address
2	FC = 5
3	Bit address (high byte) = 0
4	Bit address (low byte) = 0
5	Bit data (high byte) = 0
6	Bit data (low byte) = 0
7	CRC-16 (low byte)
8	CRC-16 (high byte)

Response from Slave:

	Not possible

Transmission of a request to all slaves is possible (slave address = 0).

The “write single bit function” is used exclusively for restarting the instrument.

If the bit address is not 0, or if it is not deleted, the controller transmits an “error response” with corresponding error code (see also chapter 4.2.7 on page 43).

Query: “Device OK?” (FC = 7)

Query from Master:

Character No.	Meaning
1	Slave address (not 0)
2	FC = 7
3	CRC-16 (low byte)
4	CRC-16 (high byte)

Response from Slave:

Character No.	Meaning
1	Slave address
2	FC = 7
3	Status
4	CRC-16 (low byte)
5	CRC-16 (high byte)

Bit 4 is set in the status if no write tasks (FC = 16) are currently possible.

Bit 5 is set if an error has occurred (operator prompt, read error status).

Other bits are set to 0.

Write Words (FC = 16)

Request from Master:

Character No.	Meaning
1	Slave address
2	FC = 16
3	Word address (high byte)
4	Word address (low byte)
5	Number of words (high byte)
6	Number of words (low byte)
7	Number of characters (n)
8	Word data (n/2 words)
...	Respective high byte first
...	...
8 + n	CRC-16 (low byte)
9 + n	CRC-16 (high byte)

Response from Slave:

Character No.	Meaning
1	Slave address (not 0)
2	FC = 16
3	Word address (high byte)
4	Word address (low byte)
5	Number of words (high byte)
6	Number of words (low byte)
7	CRC-16 (low byte)
8	CRC-16 (high byte)

Transmission of a request to all slaves is possible (slave address = 0), in which case no response ensues from the slaves.

If the word address does not exist in the controller, if the number of words is too great or if the contained data is invalid, the controller transmits an “error response” with corresponding error code (see also chapter 4.2.7 on page 43).

4.2.7 Error Handling

If the slave address does not exist, if a parity error has occurred, if the error check fails (CRC-16 false) or if the function code is not supported, the slave does not send a response.

If the controller is incapable of executing the request although the frame is formally correct, it generates an error response in whose error code (character 3) the reason for non-execution is specified.

The error response is recognized by the fact that the most significant bit is set in the returned function code.

Error Response

Character No.	Meaning
1	Slave address (not 0)
2	FC + 80h
3	Error code
4	CRC-16 (low byte)
5	CRC-16 (high byte)

Error Code

Value	Meaning
2	Impermissible address
3	Impermissible data content
6	Currently no write tasks possible
9	Number of words is too great
10	Writing impermissible

4.3 Reading and Writing Data

4.3.1 Addressing

All controller setting parameters and data are assigned to parameters groups according to functional relationships. Together with cycle data (measured values) and events data (errors and alarms), the controller can thus be operated entirely via the bus interface.

The parameters groups are addressed via a parameters index, which is used as the high byte of the word address. A complete list of all parameters indices is included in the chapter entitled “Device Parameters” on page 59.

Several quantities are usually included in each parameters index (as a rule those assigned to each of the 8 channels). Selection is made with the low byte of the word address.

4.3.2 Writing Parameters

Example:

Adjust the actuation set temperatures to 20% for the first 3 channels of the instrument with address 3.

Request from Master (±7 bit quantities are expanded to ±15 bit):

Character No.	Value	Meaning
1	03h	Device address
2	10h	Function code = write words
3	17h	Word address (high byte) = parameters index
4	00h	Word address (low byte) = 1st channel
5	00h	
6	03h	Number of words = 3
7	06h	Number of characters = 2 times 3
8	00h	
9	14h	Actuation set temperature, channel 1
10	00h	
11	14h	Actuation set temperature, channel 2
12	00h	
13	14h	Actuation set temperature, channel 3
16	DFh	
17	7Eh	CRC-16

Response from Slave (if no error has occurred):

Character No.	Value	Meaning
1	03h	Device address
2	10h	Function code = write words
3	17h	Word address (high byte) = parameters index
4	00h	Word address (low byte) = 1st channel
5	00h	
6	03h	Number of words = 3
7	84h	
8	5Eh	CRC-16

4.3.3 Reading Parameters

Example:

Read in output configuration of the instrument's four continuous outputs with address 3.

Query from Master:

Character No.	Value	Meaning
1	03h	Device address
2	03h	Function code = read words
3	37h	Word address (high byte) = parameters index
4	10h	Word address (low byte) = AO no. 17
5	00h	Number of words = 4
6	04h	
7	4Ah	CRC-16
8	5Ah	

Response from Slave (if no error has occurred):

Character No.	Value	Meaning
1	03h	Device address
2	03h	Function code = read words
3	08h	Number of characters = 2 times 4
4	00h	Output configuration AO no. 17 = heat channel 1 live zero
5	42h	
6	00h	Output configuration AO no. 18 = heat channel 2 live zero
7	46h	
8	00h	Output configuration AO no. 19 = heat channel 3 live zero
9	4Ah	
10	00h	Output configuration AO no. 20 = heat channel 4 live zero
11	4Eh	
12	D4h	CRC-16
13	46h	

4.3.4 Cycle Data

The most important controller measurement and evaluation data are contained in a single data packet. Cyclical querying of these values is thus possible by means of continuous addressing in compact form. These values can only be read.

Address	Unit	Comment
0008h	0.1°	Momentary controlled variable, channel 1
...	...	...
000Fh	0.1°	Momentary controlled variable, channel 8
0010h	%	Momentary manipulated variable, channel 1
...	...	...
0017h	%	Momentary manipulated variable, channel 8
0018h	0.1 A	Momentary heating current, channel 1
...	...	...
001Fh	0.1 A	Momentary heating current, channel 8
0020h	0.1 V	Momentary heating voltage
0021h	0.1 A	Momentary heating current channel 1, 2nd controller
...	...	...
0028h	0.1 A	Momentary heating current channel 8, 2nd controller
0029h	0.1 A	Momentary heating current channel 1, 3rd controller
...	...	...
0030h	0.1 A	Momentary heating current channel 8, 3rd controller

5 Profibus DP interface with Protocol per EN 50170

5.1 General

Interface connection is described in a separate set of installation instructions.

5.1.1 Interface Configuration

The R6000 is equipped with an RS 485 serial interface per EN 50170 (Profibus DP) for communication with a master computer, an SPC etc. Baud rates of up to 12 Mbit per second are supported.

The user address for Profibus operation is selected with the DIP switches at the front of the housing. User address changes do not become effective until the device has been switched off, and then back on again.

Address selection via the Profibus “SetSlaveAdress” function is not supported.

5.1.2 Communication Protocol

The data transmission protocol per EN 50170 is used for communication between the field control level and the device level.

5.1.3 Device Database File: GMC_059D.gsd

The file required for configuring the Profibus DP (DDBF multi-channel Profibus DP) can be downloaded free of charge from the GMC-I Gossen-Metrawatt GmbH website (http://www.gossenmetrawatt.com).

5.2 Frame Formats

Basic Layout of Output Data in the Data_Exchange Send Frame (Profibus master → R6000)

Byte Number	Function	Meaning
0	FF	Function field
1	PI	Parameters index
2		Security code 55 for FF = 1 and 3, otherwise 00
3		Security code AA for FF = 1 and 3, otherwise 00
4 ... 27		Data block

Basic Layout of Input Data in the Data_Exchange Response Frame (R6000 → Profibus master)

Byte Number	Function	Meaning
0	FF	Function field
1	PI	Parameters index
2, 3		Group error
4 ... 27		Data block

5.2.1 The Function Field

The function field contains directional and control information for transmitted user data.

Function Coding for the Function Field (FF) in the Send Frame (Profibus master → R6000)

Bit Number	Function	Meaning	
0 ... 3	Function code (FC)	**Value**	**Function**
		1	Read cyclical data / events data
		2	Read parameter with parameters index (PI)
		3	Write parameter with parameters index (PI)
		Otherwise	Idle, is responded to with empty frame (see chapter 5.2.4 on page 47)
4, 5		not in use	
6	S toggle	The Profibus DP master can use this bit in combination with the corresponding acknowledgement bit in the FF of the response frame in order to monitor processing of a parameter read or write operation which has been requested by the slave. The master sets this bit to the inverse value of the momentary value of the S toggle acknowledgement bit in its request to this end, and waits until the slave indicates processing of the request by adapting the status of the acknowledgement bit in the response frame to the status of the W toggle bit in the request frame. Use of the S toggle bit function is only absolutely essential for parameter write requests because the R6000 only executes internal write operations after the status of the S toggle bit has changed!	
7		not in use	

Function Coding for the Function Field (FF) in the Response Frame (R6000 → Profibus master)

Bit Number	Function	Meaning	
0 ... 3	Function code (FC)	**Value**	**Function**
		1	Read cyclical data / events data
		2	Read parameter with parameters index (PI)
		3	Write parameter with parameters index (PI)
		Otherwise	Idle, is responded to with empty frame (see chapter 5.2.4 on page 47)
4	Equal	Only where FC = 1, PI = 0 and FC = 1, PI = 1 This bit indicates that the parameters in the send frame and the parameters in the R6000 are identical.	
5	Busy	This bit indicates that no further write access is momentarily possible to the parameters EEPROM.	
6	S toggle acknowledgement	The R6000 Profibus DP slave adapts the status of this bit to the status of the S toggle bit in the request frame after processing the frame.	
7	L toggle	This bit is always inverted after the R6000 has processed a Data_Exchange send frame. With the help of the L toggle bit, the master is able to determine whether or not the requested values have been updated. The master must always remember the last status of these bits to this end. If the received value is not identical to the last value, the values have been updated.	

5.2.2 Parameters Index (PI)

The type of data to be transmitted is determined with the parameters index.

Selection is made from individual cyclical data or events data for function code 1.

Value	Function
0	Cyclical temperature values and on-time
1	Cyclical current and voltage values
2	Events data
3	Cyclical current values, 2nd controller
4	Cyclical current values, 3rd controller

The PI character is interpreted as follows for function codes 2 and 3 (read and write parameters):

Bits 7 ... 4	Bits 3 ... 0
0 ... Fh	0 ... Fh
Selection number for main parameters group	Selection number for special parameters from the main group

Functionally related data and setting parameters are included in the main parameters groups.

The parameters indices of main groups 0 to 9, which are documented in chapter 7 starting on page 59, can be accessed in the R6000, all others are responded to with an empty frame (see chapter 5.2.4 on page 47).

5.2.3 Data Block Length and Format

The data block is always 24 bytes long. User data length and format are variable, and are dependent upon PI or FF. One or several individual values with the following formats can be transmitted:

Format	Interpretation		
8 bit	Bit field		
± 7 bit	Number	–128 ... +127	
16 bit	Bit field		LS byte first
± 15 bit	Number	–32768 ... +32767	LS byte first

5.2.4 Empty Frames

If the R6000 receives an invalid request frame, it responds with an empty frame which basically only contains the group error:

Character No.	Function	Content
0	FF	Impermissible combination
1	PI	
2, 3		Group error
4 ... 27	Data	0

5.2.5 Group Errors

All errors from the individual channels, as well as errors which effect the device as a whole, are summed up in the group error word.

The group error is always transmitted in the response frame from the R6000, so that any errors which may have occurred can be recognized in every frame.

Bit Number	Meaning
0	Broken sensor
1	Polarity reversal
2	Second upper limit value fallen short of
3	First upper limit value fallen short of
4	First lower limit value exceeded
5	Second lower limit value exceeded
6	Impermissible parameter
7	Heating current monitoring error
8	Heating circuit error
9	Adaptation error
10	Analog error
11	Overload, heating current monitoring
12	—
13	Reference junction error
14	EEPROM error, parameter error
15	Group output error

5.3 Individual Functions

5.3.1 Cycle and Events Data (function code 1)

The R6000 responds to function code 1 with cycle or events data.
These include the most important controller measurement and evaluation data, as well as any errors which may have occurred.
The parameters index is used ins this case as a sub-distributor for cycle and events data.

Parameters index (PI):

Value	Function
0	Cyclical temperature values and on-time
1	Cyclical current and voltage values
2	Events data
3	Cyclical current values, 2nd controller
4	Cyclical current values, 3rd controller

The most important parameters are transmitted to the R6000 simultaneously along with cycle data. These parameters (the 8 setpoints and the 8 control function entries) can be written by the user. The parameter is set to the desired value in the Data_Exchange send frame to this end.

In order to start a write operation, the S toggle bit must be set to the inverse value of the momentary S toggle acknowledgement bit of the response frame. The desired parameters are then written, if no errors occur. In the process, the S toggle acknowledgement bit is set to the value of the S toggle bit, thus indicating that the write request has been registered.

The R6000 then writes the changed values to the EEPROM. The busy bit in the function field indicates that the R6000 is incapable of executing further write cycles. As long as this bit is set, no additional write access may be requested.

Characters 2 and 3 in the security code prevent undesired parameters writing operations. If the values 55h (character 2) and AAh (character 3) are assigned to these characters, parameters can be written. All other combinations of values prevent parameters writing in the R6000.

Cyclical Temperature Values and On-Time (FC = 1, PI = 0)

Cyclical data for temperature and on-time contain the momentary actual value and the manipulated variable for each channel.

Cyclical Temperature and On-Time Data, Request Frame

Character No.	Function	Content	U/M	Format	Comment
0	FF	01h, 41h	bit	8 bit	Function field
1	PI	0	bit	8 bit	Parameters index for cyclical temperature data
2		55h	bit	8 bit	Security code
3		AAh	bit	8 bit	Security code
4, 5			0.1 °	± 15 bit	Setpoint, channel 1
...			0.1 °	± 15 bit	...
18, 19			0.1 °	± 15 bit	Setpoint, channel 8
20			bit	8 bit	Controller function, channel 1
...			bit	8 bit	...
27			bit	8 bit	Controller function, channel 8

Cyclical Temperature and On-Time Data, Response Frame

Character No.	Function	Content	U/M	Format	Comment
0	FF	X1h	bit	8 bit	Function field
1	PI	0	bit	8 bit	Parameters index for cyclical temperature data
2, 3			bit	16 bit	Group error
4, 5			0.1 °	± 15 bit	Momentary controlled variable, channel 1
...			0.1 °	± 15 bit	...
18, 19			0.1 °	± 15 bit	Momentary controlled variable, channel 8
20			%	± 7 bit	Momentary manipulated variable, channel 1
...			%	± 7 bit	...
27			%	± 7 bit	Momentary manipulated variable, channel 8

Cyclical Current and Voltage Values (FC = 1, PI = 1)

Cyclical current and voltage data include the actual current value from the heating current monitoring function, as well as heating voltage.

Cyclical Current and Voltage Data, Request Frame

Character No.	Function	Content	U/M	Format	Comment
0	FF	01h, 41h	bit	8 bit	Function field
1	PI	1	bit	8 bit	Parameters index for cyclical current data
2		55h	bit	8 bit	Security code
3		AAh	bit	8 bit	Security code
4, 5			0.1 A	± 15 bit	Nominal heating current, channel 1
...			0.1 A	± 15 bit	...
18, 19			0.1 A	± 15 bit	Nominal heating current, channel 8
20			bit	8 bit	Controller function, channel 1
...			bit	8 bit	...
27			bit	8 bit	Controller function, channel 8

Cyclical Current and Voltage Data, Response Frame

Character No.	Function	Content	U/M	Format	Comment
0	FF	X1h	bit	8 bit	Function field
1	PI	1	bit	8 bit	Parameters index for cyclical current data
2, 3			bit	16 bit	Group error
4, 5			0.1 A	± 15 bit	Momentary heating current, channel 1
...			0.1 A	± 15 bit	...
18, 19			0.1 A	± 15 bit	Momentary heating current, channel 8
20, 21			0.1 V	± 15 bit	Momentary heating voltage
22 ... 27					not in use

Events Data (FC = 1, PI = 2)

Events data include all error messages and alarms generated by the device. They can be queried in order to identify a specific error or alarm. Error messages and alarms can be simultaneously reset. This is accomplished by linking the errors and alarms to the transmitted values, so that individual errors can be selectively acknowledged.

Events Data Request Frame

Character No.	Function	Content	U/M	Format	Comment
0	FF	01h, 41h	bit	8 bit	Function field
1	PI	2	bit	8 bit	Parameters index for events data
2		55h	bit	8 bit	Security code
3		AAh	bit	8 bit	Security code
4, 5			bit	16 bit	Error status, channel 1
...			bit	16 bit	...
18, 19			bit	16 bit	Error status, channel 8
20, 21			bit	16 bit	Device error status
22			bit	8 bit	Output error 1
...			bit	8 bit	...
27			bit	8 bit	Output error 6

Events Data Response Frame

Character No.	Function	Content	U/M	Format	Comment
0	FF	X1h	bit	8 bit	Function field
1	PI	2	bit	8 bit	Parameters index for events data
2, 3			bit	16 bit	Group error
4, 5			bit	16 bit	Error status, channel 1
...			bit	16 bit	...
18, 19			bit	16 bit	Error status, channel 8
20, 21			bit	16 bit	Device error status
22			bit	8 bit	Output error 1
...			bit	8 bit	...
27			bit	8 bit	Output error 6

Bit assignments for the error status word and the output error are described in chapter 7.4.3 on page 63.

Cyclical Current Values, 2nd Controller (FC = 1, PI = 3)

The cycle data include actual current values for heating current monitoring at the 2nd controller with monitoring of 16/24 channels (see also chapter 2.8.6 on page 22).

Cyclical Current and Voltage Data, Request Frame

Character No.	Function	Content	U/M	Format	Comment
0	FF	01h, 41h	bit	8 bit	Function field
1	PI	3	bit	8 bit	Parameters index for current cycle data
2		55h	bit	8 bit	Security code
3		AAh	bit	8 bit	Security code
4, 5			0.1 A	± 15 bit	Heating current nominal value channel 1, 2nd controller
...			0.1 A	± 15 bit	...
18,19			0.1 A	± 15 bit	Heating current nominal value channel 8, 2nd controller
20 ... 27					not in use

Cyclical Current and Voltage Data, Response Frame

Character No.	Function	Content	U/M	Format	Comment
0	FF	X1h	bit	8 bit	Function field
1	PI	3	bit	8 bit	Parameters index for current cycle data
2, 3			bit	16 bit	Group error
4, 5			0.1 A	± 15 bit	Momentary heating current channel 1, 2nd controller
...			0.1 A	± 15 bit	...
18,19			0.1 A	± 15 bit	Momentary heating current channel 8, 2nd controller
20 ... 27					not in use

Cyclical Current Values, 3rd Controller (FC = 1, PI = 4)

The cycle data include actual current values for heating current monitoring at the 3rd controller with monitoring of 16/24 channels (see also chapter 2.8.6 on page 22).

Cyclical Current and Voltage Data, Request Frame

Character No.	Function	Content	U/M	Format	Comment
0	FF	01h, 41h	bit	8 bit	Function field
1	PI	4	bit	8 bit	Parameters index for current cycle data
2		55h	bit	8 bit	Security code
3		AAh	bit	8 bit	Security code
4, 5			0.1 A	± 15 bit	Heating current nominal value channel 1, 3rd controller
...			0.1 A	± 15 bit	...
18,19			0.1 A	± 15 bit	Heating current nominal value channel 8, 3rd controller
20 ... 27					not in use

Cyclical Current and Voltage Data, Response Frame

Character No.	Function	Content	U/M	Format	Comment
0	FF	X1h	bit	8 bit	Function field
1	PI	4	bit	8 bit	Parameters index for current cycle data
2, 3			bit	16 bit	Group error
4, 5			0.1 A	± 15 bit	Momentary heating current channel 1, 3rd controller
...			0.1 A	± 15 bit	...
18,19			0.1 A	± 15 bit	Momentary heating current channel 8, 3rd controller
20 ... 27					not in use

5.3.2 Reading Parameters (function code 2)

Parameters can be read with function code 2. The desired parameters index (PI) is entered to the Data_Exchange send frame to this end. The desired parameters are then transmitted with the Data_Exchange response frame after slave response time has expired.

With the help of the L toggle bit, the master is able to determine whether or not the requested values have been updated. The master must always remember the last status of these bits to this end. If the received value is not identical to the last value, the values have been updated.

All parameters indices of main groups 0 to 9 are supported as described in chapter 7 starting on page 59.
The number of transmitted data depends upon the parameters index and can be determined based upon the “format” and the “number”.

Read

Request

Character No.	Function	Content	U/M	Format	Comment
0	FF	02h, 42h	bit	8 bit	Function field
1	PI		bit	8 bit	Parameters index
2		00	bit	8 bit	Security code
3		00	bit	8 bit	Security code
4 ... 27					not in use

Response

Character No.	Function	Content	U/M	Format	Comment
0	FF	X2h	bit	8 bit	Function field
1	PI		bit	8 bit	Parameters index
2, 3			bit	16 bit	Group error
4 ... n	Data				Requested data
n + 1 ... 27					not in use

5.3.3 Writing Parameters (function code 3)

Parameters can be written with function code 3. The desired parameters group (PI) is entered to the Data_Exchange send frame to this end, and the corresponding parameters are written with the desired value.
In order to start a write operation, the S toggle bit must be set to the inverse value of the momentary S toggle acknowledgement bit of the response frame. The desired parameters are then written if no errors occur, and changed parameters are transmitted with the Data_Exchange response frame after slave response time has expired. In the process, the S toggle acknowledgement bit is set to the value of the S toggle bit, thus indicating that the write request has been registered. The R6000 then writes the changed values to the EEPROM. The busy bit in the function field indicates that the R6000 is incapable of executing further write cycles. As long as this bit is set, no additional write access may be requested.

All parameters groups of main groups 0 to 9 are supported as described in chapter 7 starting on page 59.
The number of transmitted data depends upon the parameters index and can be determined based upon the “format” and the “number”.

If only the parameters of certain individual channels or outputs need to be changed, parameters must nevertheless be transmitted for all channels or outputs, because writing always takes place for all parameters.

Write

Request

Character No.	Function	Content	U/M	Format	Comment
0	FF	03h, 43h	bit	8 bit	Function field
1	PI		bit	8 bit	Parameters index
2		55h	bit	8 bit	Security code
3		AAh	bit	8 bit	Security code
4 ... n	Data				Data to be written
n + 1 ... 27					not in use

Response

Character No.	Function	Content	U/M	Format	Comment
0	FF	X3h	bit	8 bit	Function field
1	PI		bit	8 bit	Parameters index
2, 3			bit	16 bit	Group error
4 ... n	Data				Written data
n + 1 ... 27					not in use

6 CAN-Bus, CANopen Protocol

6.1 General

Interface connection is described in a separate set of installation instructions.
Details regarding use of the CAN interface are included in the CAN / CANopen standard.

6.1.1 Interface Configuration

- Connection: Only the two signal lines and the ground wire need to be connected. The optional, external positive power supply line is not included.
- Baud rate: The baud rates specified by CANopen, i.e. 10 kBit to 1 MBit per second, are supported. The desired rate can be selected via the service interface (PI = A1h).
- Node ID: The node address is set with “bus addr.” DIP switches 1 through 7 at the front of the housing.

6.1.2 Primary Function

Data Exchange

- In accordance with CANopen, data is exchanged by means of SDOs (service data objects) and PDOs (process data objects). A description is included in chapters 6.2 and 6.3.
- The bus master is capable of accessing all parameters, configurations and bus user (node) data with the SDOs. Communication takes place in accordance with the master-slave principle, i.e. the node responds to each request.
- The PDOs are used for continuous data exchange amongst the bus users. They must be configured by the master with an SDO after resetting the R6000, and do not become active until the R6000 is switched to the “operational mode”. These transmissions are not answered.

Network Management

CANopen utilizes a wide variety of objects in order to assure trouble-free network operation. Please refer to the CANopen standard for details. Details specific to the R6000 are included as of chapter 6.4.

Frame Layout

Correct frame layout is assured by the hardware. Only the general layout is mentioned here:

- The arbitration field is transmitted first, which includes the COB ID (communication object identifier, 11 bits).
  The lower the COB ID, the higher the priority of the message.
- Then the control field is transmitted. The control field contains the number of transmitted data bytes (LEN, 4 bits). This number lies within a range of 0 to 8.
- The data field is then transmitted, containing up to 8 data bytes which have various functions depending upon the frame.
- Finally, the CRC and the ACK fields are transmitted (no longer mentioned in later chapters).

6.1.3 ESD File

The ESD file which is required for project engineering can be downloaded from the Internet at: www.gossenmetrawatt.com

6.2 Service Data Objects (SDOs)

The bus master is capable of accessing all parameters, configurations and R6000 data with the SDOs. The data which are transmitted with the PDOs can also be accessed.

Frame Layout

	Byte	Value	Meaning
COB ID		600h + node ID 580h + node ID	Request from master Response from slave
LEN		8	Always 8 data bytes
Command	1		Type of transmission
Index	2, 3		Parameter selection (see object index, chapter 6.7 on page 58)
Sub-index	4	1 ... n 0	If the object has more than one value (e.g. channel number) If the object has only one value, or if the number of values is queried
Net data	5 ... 8	1 ... 4 bytes of data 0	In the case of writing by the master or response to a query In the case of a query from the master or response to writing

Write example: Adjust setpoint to 195.0° C at channel 3 of the R6000 with bus address 5
195.0° C => 1950 = 079Eh
The setpoint has an index of 2000h

	COB ID	LEN	Com	Index		Sub-Index	Data			
Master:	605h	8	2Bh	00h	20h	03h	9Eh	07h	00h	00h
R6000:	585h	8	60h	00h	20h	03h	00h	00h	00h	00h

Read example: Read output configuration of the 2nd continuous output at the R6000 with bus address 11
2. Continuous output = output no. 18 => sub-index 17 = 11h
The output configuration has an index of 2037h

	COB ID	LEN	Com	Index		Sub-Index	Data			
Master:	60Bh	8	40h	37h	20h	11h	00h	00h	00h	00h
R6000:	58Bh	8	47h	37h	20h	11h	32h	00h	00h	00h

Output configuration = 32h = cooling manipulated variable for channel 4, dead zero

6.3 Process Data Objects (PDOs)

PDOs are used for continuous data exchange amongst the bus users. PDOs are transmitted and accepted when the R6000 is in the "operational mode".
As opposed to SDOs, all 8 bytes are used as net data with PDOs. The contents of the PDOs are established by means of PDO mapping, which cannot be changed with the R6000.
The R6000 supports 4 send PDOs, by means of which, for example, momentary actual values can be transmitted to the master, as well as 4 receive PDOs, by means of which new setpoints can be transferred to the R6000.

6.3.1 PDO Configuration

The PDOs are configured by means of SDOs. Configuration determines whether or not the PDO is enabled, and whether it reacts synchronously or asynchronously.

	Byte	Value	Meaning
COB ID		600h + node ID 580h + node ID	Request from master Response from slave
LEN		8	Always 8 data bytes
Command	1		Type of transmission
Index	2, 3	1400h 1401h 1402h 1403h	1st receive PDO 2nd receive PDO 3rd receive PDO 4th receive PDO
		1800h 1801h 1802h 1803h	1st send PDO 2nd send PDO 3rd send PDO 4th send PDO
Sub-index	4	1 2	Establish COB ID and enable Establish synchronous or asynchronous
Data	5 ... 8	Configuration	See table

Configuration:

Sub-Index		Meaning
1	00000000h + COB ID 80000000h + COB ID	COB ID of the PDO does not have to correspond to the default value. The most significant bit is set when the PDO is disabled.
2	00h 01h ... F0h = n FFh	Synchronous, not cyclical (i.e. only if content is changed) Synchronous, cyclical transmission after each n-ten sync-signal Asynchronous

6.3.2 PDO Time Response

- Asynchronous send PDOs are always (immediately) transmitted when their contents change.
- Synchronous send PDOs are not transmitted until a SYNC is received (see also chapter 6.4 on page 57).

- The contents of asynchronous receive PDOs are activated at the R6000 immediately after receipt.
- The contents of synchronous receive PDOs are not accepted by the R6000 until a SYNC has been received.

6.3.3 PDO Frame Layout

	Byte	Value	Meaning
COB ID		Default value: 180h + node ID ... 480h + node ID	 1. send PDO ... 4. send PDO
		Default value: 200h + node ID ... 500h + node ID	 1. receive PDO ... 4. receive PDO
LEN		8	Always 8 data bytes
Data	1 ... 8	User data	PDO mapping is fixed (see chapters 6.3.4 and 6.3.5).

6.3.4 Contents of Send PDOs

The “fixed decimal” format is the “Int16” format, the value is specified in 1/10 of the physical unit.

	Byte	Value	Format	Meaning	
COB ID		180h + node ID		1st send PDO	
LEN		8			
Data	1, 2		Fixed decimal	Channel 1 actual value	Index 2100h
	3, 4		Fixed decimal	Channel 2 actual value	
	5, 6		Fixed decimal	Channel 3 actual value	
	7, 8		Fixed decimal	Channel 4 actual value	

	Byte	Value	Format	Meaning	
COB ID		280h + node ID		2nd send PDO	
LEN		8			
Data	1, 2		Fixed decimal	Channel 5 actual value	Index 2100 h
	3, 4		Fixed decimal	Channel 6 actual value	
	5, 6		Fixed decimal	Channel 7 actual value	
	7, 8		Fixed decimal	Channel 8 actual value	

The 3rd send PDO depends on bit 1 of the device control.

When bit 1 of the device control has been set to „with master PDO“:

	Byte	Value	Format	Meaning	
COB ID		380h + node ID		3rd send PDO	
LEN		8			
Data	1, 2		Fixed decimal	Master actual value of group 0	Index 2026h
	3, 4		Fixed decimal	Master actual value of 1st group	
	5, 6		Fixed decimal	Master actual value of 2nd group	
	7,8		Fixed decimal	Master actual value of 3rd group	

When bit 1 of the device control has been deleted:

	Byte	Value	Format	Meaning	
COB ID		380h + node ID		3rd send PDO	
LEN		8			
Data	1		Int8	Manipulated variable, channel 1	Index 2101h
	2		Int8	Manipulated variable, channel 2	
	3		Int8	Manipulated variable, channel 3	
	4		Int8	Manipulated variable, channel 4	
	5		Int8	Manipulated variable, channel 5	
	6		Int8	Manipulated variable, channel 6	
	7		Int8	Manipulated variable, channel 7	
	8		Int8	Manipulated variable, channel 8	

	Byte	Value	Format	Meaning	
COB ID		480h + node ID		4th send PDO	
LEN		8			
Data	1		Unsigned8	Compressed status, channel 1	Index 2121h
	2		Unsigned8	Compressed status, channel 2	
	3		Unsigned8	Compressed status, channel 3	
	4		Unsigned8	Compressed status, channel 4	
	5		Unsigned8	Compressed status, channel 5	
	6		Unsigned8	Compressed status, channel 6	
	7		Unsigned8	Compressed status, channel 7	
	8		Unsigned8	Compressed status, channel 8	

Direct querying of the channel error or acknowledgement of individual error bits is accomplished by means of SDOs at index 2021, sub-indices 1 through 8 (see also chapter 7.4.3 on page 63).

The compressed channel status has the following bit assignments:

Bit no.	Meaning
0	Broken sensor or polarity reversal
1	1st or 2nd upper limit value exceeded
2	1st or 2nd lower limit value exceeded
3	Heating current monitoring error
4	Heating circuit error
5	Adaptation error
6	Controller ON
7	Adaptation in progress

6.3.5 Contents of Receive PDOs

The “fixed decimal” format is the “Int16” format, the value is specified in 1/10 of the physical unit. As opposed to writing a setpoint with an SDO, the setpoints are not transferred to parameters memory (EEPROM). When the “setpoint 2” bit is set for the controller function, the received value is not used as a second setpoint and is saved instead to RAM as the first setpoint.

	Byte	Value	Format	Meaning	
COB ID		200h + node ID		1st receive PDO	
LEN		8			
Data	1, 2		Fixed decimal	Channel 1 setpoint	Index 2000h
	3, 4		Fixed decimal	Channel 2 setpoint	
	5, 6		Fixed decimal	Channel 3 setpoint	
	7, 8		Fixed decimal	Channel 4 setpoint	

	Byte	Value	Format	Meaning	
COB ID		300h + node ID		2nd receive PDO	
LEN		8			
Data	1, 2		Fixed decimal	Channel 5 setpoint	Index 2000h
	3, 4		Fixed decimal	Channel 6 setpoint	
	5, 6		Fixed decimal	Channel 7 setpoint	
	7, 8		Fixed decimal	Channel 8 setpoint	

	Byte	Value	Format	Meaning	
COB ID		400h + node ID		3rd receive PDO	
LEN		8			
Data	1		Unsigned8	Channel 1 controller function	Index 2020h
	2		Unsigned8	Channel 2 controller function	
	3		Unsigned8	Channel 3 controller function	
	4		Unsigned8	Channel 4 controller function	
	5		Unsigned8	Channel 5 controller function	
	6		Unsigned8	Channel 6 controller function	
	7		Unsigned8	Channel 7 controller function	
	8		Unsigned8	Channel 8 controller function	

The 4th receive PDO depends on bit 1 of the device control. When bit 1 of the device control has been set to „with master PDO“:

	Byte	Valuet	Format	Meaning	
COB-ID		500h + node ID		4th receive PDO	
LEN		8			
Data	1, 2		Fixed decimal	Master actual value of group 0	Index 2026h
	3, 4		Fixed decimal	Master actual value of 1st group	
	5, 6		Fixed decimal	Master actual value of 2nd group	
	7, 8		Fixed decimal	Master actual value of 3rd group	

When bit 1 of the device control has been deleted:

	Byte	Value	Format	Meaning	
COB ID		500h + node ID		4th receive PDO	
LEN		8			
Data	1		Unsigned8	Channel 1 controller function mask	Index 2120h
	2		Unsigned8	Channel 2 controller function mask	
	3		Unsigned8	Channel 3 controller function mask	
	4		Unsigned8	Channel 4 controller function mask	
	5		Unsigned8	Channel 5 controller function mask	
	6		Unsigned8	Channel 6 controller function mask	
	7		Unsigned8	Channel 7 controller function mask	
	8		Unsigned8	Channel 8 controller function mask	

Changed bits in the controller function are only accepted if the corresponding bits in the “controller function mask” byte have been set. Changed bits are stored to the EEPROM.

If the 4th receive PDO contains the master actual values, all bits are set in the bytes „Controller function mask“.

Bit assignments for the controller function and the mask (see also chapter 7.4.2 on page 62)

Bit No.	Meaning
0	Setpoint 2 active
1	Actuating circuit
2	Feed-forward control
3	Setpoint rise (boost)
4	Switch controller active
5	Clear error
6	Controller on
7	Start adaptation

6.4 SYNC Object

Synchronous PDOs are evaluated and transmitted by the R6000 after receiving a SYNC message. The R6000 must be in the "operational mode" to this end, and the PDOs must be configured as synchronous. The SYNC message from the master is addressed to all users within the network, and has very high priority. The frame contains no data:

	Value	Meaning
COB ID	080h	SYNC
LEN	0	No data

6.5 Emergency Object

If a "device error" occurs at the R6000 (see also index 2021, sub-index 9), it transmits an EMCY frame. After all errors have been eliminated, the R6000 transmits and EMCY error reset frame.

	Byte	Value	Meaning
COB ID		080h + node ID	EMCY
LEN		8	
Emergency error code	1, 2	FFxxh 0000h	New error has occurred One error eliminated
Error register	3	21h 00h	Error persists (generic + device specific error) No more errors
Data	4 ... 8	0	Unused

The device error is added to the low byte of the emergency error code, compressed to one byte:

Bit No.	Meaning
0	Analog error
1	Overdrive, heating current monitoring
2	Invalid characteristics combination
3	Reference junction error
4	EEPROM error, parameter error
5	Group output error
6	Mapping error
7	--

Error history can be queried from object 1003h. The number of saved errors is stored to sub-index 0, and saved emergency error codes can be queried beginning with sub-index 1, whose low bytes contain the compressed device errors.

Detailed querying of the device error or acknowledgement of individual error bits is accomplished by means of SDOs at index 2021, sub-index 9 (see also chapter 7.4.3 on page 63).

6.6 NMT Object

The master controls the slaves in the CANopen network by means of network management. The R6000 supports the stipulated command specifier (CS):

	Byte	Value	Meaning
COB ID		000h	NMT
LEN		2	
CS	1	01h 02h 80h 81h 82h	Enter operational mode Stop remote Enter pre-operational mode Reset nodes Reset communication
NODE ID	2	00h 01h ... 7Fh	For all For specified nodes only

The individual commands control performance of the R6000 in the CANopen network, and have no influence on controller functions.

Exception: CS = 81h executes a reset of the R6000 (same as after interruption of auxiliary power).

6.7 Object Index

This chapter only deals with the manufacturer-specific part of the object index (indices 2000h through 5FFFh).
The object indices are derived from the parameter indices (see also chapter 7 on page 59). A complete description can be found there.
The “fixed decimal” format specified in the type column is the “Int16” format, the value is specified in 1/10 of the physical unit.

Index (hexadecimal)	Object	Name	Type	Attribute
		Temperature Parameters		
2000	Array[8]	Setpoint	Fixed decimal	RW
2001	Array[8]	First upper limit value	Fixed decimal	RW
2002	Array[8]	First lower limit value	Fixed decimal	RW
2003	Array[8]	Setpoint 2	Fixed decimal	RW
2004	Array[8]	Second upper limit value	Fixed decimal	RW
2005	Array[8]	Second lower limit value	Fixed decimal	RW
2006	Array[8]	Minimum setpoint	Fixed decimal	RW
2007	Array[8]	Maximum setpoint	Fixed decimal	RW
2008	Array[8]	Setpoint rise (Boost)	Fixed decimal	RW
2009	Array[8]	Boost duration	Fixed decimal	RW
200A	Array[8]	Actuation setpoint	Fixed decimal	RW
200B	Array[8]	Dwell time during actuation	Fixed decimal	RW
200C	Array[8]	Actual value correction	Fixed decimal	RW
200D	Array[8]	Actual value factor	Fixed decimal	RW
200E	Array[8]	Setpoint ramp, up	Fixed decimal	RW
200F	Array[8]	Setpoint ramp, down	Fixed decimal	RW
		Control Parameters		
2010	Array[8]	Proportional zone heating	Fixed decimal	RW
2011	Array[8]	Proportional zone cooling	Fixed decimal	RW
2012	Array[8]	Dead zone	Fixed decimal	RW
2014	Array[8]	System delay	Fixed decimal	RW
2015	Array[8]	Cycle Time	Fixed decimal	RW
2016	Array[8]	Actuator manipulating factor	Int8	RW
2017	Array[8]	Actuation manipulating factor	Int8	RW
2018	Array[8]	Motor actuation time	Fixed decimal	RW
2019	Array[8]	Influencing quantity manipulating factor	Int8	RW
201C	Array[8]	Minimum manipulating factor	Int8	RW
201D	Array[8]	Maximum manipulating factor	Int8	RW
201E	Array[8]	Sensor error manipulating factor	Int8	RW
201F	Array[8]	Switching hysteresis	Fixed decimal	RW
		Control Commands		
2020	Array[8]	Controller function	Unsigned8	RW
2021	Array[12]	Error status	Unsigned16	RW
2022	Array[8]	Controller configuration	Unsigned16	RW
2023	Array[8]	Extended control configuration	Unsigned8	RW
2024	Array[9]	Controller status, message word	Unsigned16	RO
2025	Array[8]	Oscillation hold-off	Int8	RW
2026	Array[4]	Master actual value	Fixed decimal	RW
2027	Array[8]	External actual value	Fixed decimal	RW
2028	Array[8]	Manual manipulating factor	Int8	RW
2029	Array[8]	Channel error mask	Unsigned16	RW
202A	Array[8]	Group error mask	Unsigned16	RW
202D	Var	Alarm history read-out starting point	Int16	RW
202E	Array[15]	Alarm history	Unsigned16	RO
202F	Var	Number of alarm history entries	Int16	RO
		Device Specification		
2031	Var	Device characteristic	Unsigned8	RO
2032	Var	Controlled variable quantity / device control	Unsigned8	RW
2033	Array[8]	Sensor type	Unsigned8	RW
2036	Array[8]	Limit value configuration	Unsigned8	RW
2037	Array[20]	Output configuration	Unsigned8	RW
203A	Var	Power limitation	Unsigned8	RW
		Heating Current Monitoring		
2060	Array[8]	Nominal heating current	Fixed decimal	RW
2061	Array[8]	Heating current nominal value 2nd controller	Fixed decimal	RW
2062	Array[8]	Heating current nominal value 3rd controller	Fixed decimal	RW
2064	Var	Summation current transformation ratio	Fixed decimal	RW
2067	Var	Heating current sampling cycle	Fixed decimal	RW
2069	Var	Heating voltage transformer secondary voltage	Fixed decimal	RW
		Data Logger		
2090	Array[3]	Current time	Unsigned16	RW
2092	Var	Logger sampling cycle	Fixed decimal	RW
2093	Var	Logger configuration	Unsigned8	RW
2094	Var	Read-out starting point sampled actual values	Int16	RW
2095	Var	Read-out starting point sampled manipulated variables	Int16	RW
2096	Array[8]	Sampled actual values	Fixed decimal	RO
2097	Array[8]	Sampled manipulated variables	Fixed decimal	RO
2098	Var	Number of samples	Int16	RO
		Interface		
20A0	Var	RS 232 – RS 485 Interface configuration	Unsigned8	RO
		Temporary Values		
20B0	Array[8]	Momentary setpoint	Fixed decimal	RO
20E0	Array[2]	State of binary I/O’s	Unsigned16	RW
20E1	Array[4]	State of continuous outputs	Unsigned16	RW
2100	Array[8]	Momentary actual value	Fixed decimal	RO
2101	Array[8]	Momentary manipulating factor	Int8	RO
2102	Array[24]	Momentary heating current	Fixed decimal	RO
2103	Var	Momentary heating voltage	Fixed decimal	RO
2120	Array[8]	Controller function mask	Unsigned8	RW
2121	Array[8]	Compressed channel status	Unsigned8	RO

7 Device Parameters

7.1 Overview

Channel-specific Quantities

Main Group	PI	Value	Format	fC, tC, PN	Number	Comment
0		**Temperature Parameters**				
	00	Setpoint	± 15 bit	✓	8	
	01	First upper limit value	± 15 bit	✓	8	
	02	First lower limit value	± 15 bit	✓	8	
	03	Proxy setpoint	± 15 bit	✓	8	
	04	Second upper limit value	± 15 bit	✓	8	
	05	Second lower limit value	± 15 bit	✓	8	
	06	Minimum setpoint	± 15 bit	✓	8	
	07	Maximum setpoint	± 15 bit	✓	8	
	08	Setpoint rise (Boost)	± 15 bit	✓	8	
	09	Boost duration	± 15 bit	✓	8	
	0A	Actuation setpoint	± 15 bit	✓	8	
	0B	Dwell time (during actuation)	± 15 bit	✓	8	
	0C	Actual value correction	± 15 bit	✓	8	
	0D	Actual value factor	± 15 bit	✓	8	
	0E	Setpoint ramp, up	± 15 bit	✓	8	
	0F	Setpoint ramp, down	± 15 bit	✓	8	
1		**Control Parameters**				
	10	Proportional zone heating (Xpl)	± 15 bit	✓	8	
	11	Proportional zone cooling (Xpll)	± 15 bit	✓	8	
	12	Dead zone	± 15 bit	✓	8	
	14	System delay (Tu)	± 15 bit	✓	8	
	15	Cycle time	± 15 bit	✓	8	
	16	Actuator manipulating factor	± 7 bit	✓	8	
	17	Actuation manipulating factor	± 7 bit	✓	8	
	18	Motor actuation time	± 15 bit	✓	8	
	19	Influencing quantity manipulating factor	± 7 bit	✓	8	
	1C	Minimum manipulating factor	± 7 bit	✓	8	
	1D	Maximum manipulating factor	± 7 bit	✓	8	
	1E	Sensor error manipulating factor	± 7 bit	✓	8	
	1F	Switching hysteresis	± 15 bit	✓	8	
2		**Control Commands**				
	20	Controller function	8 bit	✓	8	
	21	Error status	16 bit	✓	12	Words 1 ... 8 are channel-specific
	22	Controller configuration	16 bit	✓	8	
	23	Expanded controller configuration	8 bit	✓	8	
	24	Controller status, message word	16 bit	✓	9	Read only
	25	Oscillation hold-off	8 bit	✓	8	
	27	external actual value	± 15 bit	✓	8	
	28	Manual manipulating factor	± 7 bit	✓	8	
	29	Channel error mask	16 bit	✓	8	
3		**Device Specifications**				
	33	Sensor type	8 bit	✓	8	
	36	Limit value configuration	8 bit	✓	8	
6		**Heating Current Monitoring**				
	60	Nominal heating current	± 15 bit	✓	8	
	6C	Heating current - actual value	± 15 bit	✓	8	Read only
A		**Interfaces**				Not via Profibus
	A0	Interface configuration	8 bit		1	
	A1	CAN baud rate	8 bit		1	not for CANopen
B		**Display Values**				not via serial interface
	B0	Momentary setpoint	± 15 bit	✓	8	Read only
	B1	Momentary actual value	± 15 bit	✓	8	Read only
	B2	Momentary system deviation	± 15 bit	✓	8	Read only
	B7	Momentary manipulating factor	± 15 bit	✓	8	Read only
	B8	Momentary setpoint (whole degrees)	± 15 bit	✓	8	Read only
	B9	Momentary actual value (whole degrees)	± 15 bit	✓	8	Read only
	BA	Momentary system deviation (whole degrees)	± 15 bit	✓	8	Read only

Device-specific Quantities

Main Group	PI	Value	Format	fC, tC, PN	Number	Comment
2		**Control Commands**				
	21	Error status	16 bit	✓	12	Words 9 ... 12 are device-specific
	26	Master actual value	± 15 bit	✓	4	
	2A	Group error mask	16 bit	✓	8	
3		**Device Specifications**				
	30	Device ID	8 bit		1	Read only
	31	Device characteristic	8 bit		1	Read only
	32	Device control	8 bit		1	
	35	Software version	8 bit		1	Read only
	37	Output configuration I/O 1 ... 16 continuous output 1 ... 4	8 bit	✓	20	
	3A	Power limitation	± 7 bit		1	
	3F	Parameter set ID	16 bit	✓	3	
6		**Heating Current Monitoring**				
	61	Heating current nominal value 2nd controller	± 15 bit	✓	8	
	62	Heating current nominal value 3rd controller	± 15 bit	✓	8	
	64	Summation current transformation ratio	± 15 bit	✓	1	
	67	Heating current sampling cycle	± 15 bit	✓	1	
	69	Heating voltage transformer secondary voltage	± 15 bit	✓	1	
	6D	Heating current - actual value 2nd controller	± 15 bit	✓	8	Read only
	6E	Heating current - actual value 3rd controller	± 15 bit	✓	8	Read only
	6F	Heating voltage - actual value	± 15 bit	✓	1	Read only
B		**Display Values**				
	B3	Reference junction temperature	± 15 bit	✓	1	Read only

Special functions

Main Group	PI	Value	Format	fC, tC, PN	Number	Comment
2		**Control Commands**				
	2C	Alarm history, time stamp	16 bit		3	Read only, not via serial interface
	2D	Alarm history read-out starting point	± 15 bit		1	
	2E	Alarm history	16 bit	✓	15/12	Read only
	2F	Number of alarm history entries	± 15 bit		1	Read only
9		**Data logger**				
	90	Current time	16 bit	✓	3	No real-time clock
	92	Logger sampling cycle	± 15 bit		1	
	93	Logger control	8 bit		1	
	94	Read-out starting point sampled actual values	± 15 bit		1	
	95	Read-out starting point sampled manipulated variables	± 15 bit		1	
	96	Sampled actual values	± 15 bit	✓	(1 ... 15) x 8	Read only
	97	Sampled manipulated variables	± 15 bit	✓	(1 ... 15) x 8	Read only
	98	Number of samples	± 15 bit		1	Read only
	99	Time of last sample	16 bit	✓	3	No real-time clock
E		**Control Functions**				
	E0	State of binary I/O's	16 bit	✓	2	
	E1	State of continuous outputs	16 bit	✓	4	

All setting parameters and data are assigned to parameters groups according to functional relationships.
Together with cycle data and events data, the controller can thus be operated entirely via the bus interface.

The Profibus DP interface always transmits all parameters of any given parameters index, whereas parameters can be selected from individual channels with the other interfaces.

7.2 Main Group 0: Temperature Parameters

7.2.1 Table of Parameter Indices

PI	Parameter Designation	U/M	Format	Number	Setting Range	Comment
00h	Setpoint	0.1°	± 15 bit	8	Minimum ... maximum setpoint	
01h	First upper limit value	0.1°	± 15 bit	8	0° = off, –MRS ... +MRS *)	For Relative Limit Value
					0° = off, –MRS ... +MRS	For abs. LV and differential controller
					0° C / 32° F = off, MRL ... MRU	For abs. LV and abs. value controller
02h	First lower limit value	0.1°	± 15 bit	8	Same as PI = 01h	Same as PI = 01h
03h	Proxy setpoint	0.1°	± 15 bit	8	Same as PI = 00h	Same as PI = 00h
04h	Second upper limit value	0.1°	± 15 bit	8	Same as PI = 01h	Same as PI = 01h
05h	Second lower limit value	0.1°	± 15 bit	8	Same as PI = 01h	Same as PI = 01h
06h	Minimum setpoint	0.1°	± 15 bit	8	MRL ... maximum setpoint *)	For absolute value controller
					–MRS ... maximum setpoint	For differential controller
07h	Maximum setpoint	0.1°	± 15 bit	8	Minimum setpoint ... MRU *)	For absolute value controller
					Minimum setpoint ... MRS	For differential controller
08h	Setpoint rise (Boost)	0.1°	± 15 bit	8	–MbU ... +MbU	
09h	Boost duration	0.1 s	± 15 bit	8	0.0 ... 3000.0 s	
0Ah	Actuation setpoint	0.1°	± 15 bit	8	Same as PI = 00h	Same as PI = 00h
0Bh	Dwell time (during actuation)	0.1 s	± 15 bit	8	0 ... 30000	
0Ch	Actual value correction	0.1°		8	–MRS ... +MRS *)	
0Dh	Actual value factor	‰ / 0.1°	± 15 bit	8	10.0 ... 1800.0 ‰ / °C	
0Eh	Setpoint ramp, up	0.1° / min.	± 15 bit	8	0 = off, 1 ... MRS *)	
0Fh	Setpoint ramp, down	0.1° / min.	± 15 bit	8	0 = off, 1 ... MRS *)	

*) MRL = measuring range lower limit, MRU = measuring range upper limit, MRS = measuring range span

7.2.2 Unit of Measure and Setting Range

Units of measure and setting ranges for temperature parameters are dependent upon:

- The configured **quantity** for the controlled variable (PI = 32h)
- The configured **sensor type** (PI = 33h)

Sensor Type		Measuring Range Lower Limit		Measuring Range Upper Limit		Polarity Reversal / Short-Circuit		Broken Sensor	
Value	Type	°C	°F	°C	°F	°C	°F	°C	°F
0	J	0	32	900	1652	–20	–4	942.3	1728.1
1	L	0	32	900	1652	–20	–4	900	1652
2	K	0	32	1300	2372	–20	–4	1366.7	2492.1
3	B	0	32	1800	3272	–20	–4	1802.3	3276.1
4	S	0	32	1750	3182	–20	–4	1768.1	3214.6
5	R	0	32	1750	3182	–20	–4	1768.1	3214.6
6	N	0	32	1300	2372	–20	–4	1300	2372
7	E	0	32	700	1292	–20	–4	715.3	1319.5
8	T	0	32	400	752	–20	–4	400	752
9	U	0	32	600	1112	–20	–4	600	1112
10	Linear 1)	0 mV		50 mV		–5 mV		60 mV	
11	Pt100	–200	–328	600	1112	–220	–364	700 2)	1292 2)
12	Ni100	–50	–58	250	482	–60	–76	250	482

1) Scalable temperature, observe instructions in chapter 2.3.6 on page 11!
2) Depends upon cable resistance

Units of measure depend upon the quantity °C per minute or °F per minute where setpoint ramps are concerned.

7.3 Main Group 1: Control Parameters

7.3.1 Table of Parameter Indices

PI	Parameter Designation	U/M	Format	Number	Setting Range	Comment
10h	Proportional zone heating	0.1°	± 15 bit	8	0 ... MRS *)	
11h	Proportional zone cooling	0.1°	± 15 bit	8	0 ... MRS *)	
12h	Dead zone	0.1°	± 15 bit	8	0 ... MRS *)	
14h	System delay	0.1 s	± 15 bit	8	0 ... 30000	
15h	Cycle time	0.1 s	± 15 bit	8	1 ... 3000	
16h	Actuator manipulating factor	%	± 7 bit	8	Min. ... max. manipulating factor	
17h	Actuation manipulating factor	%	± 7 bit	8	Min. ... max. manipulating factor	
18h	Motor actuation time	0.1 s	± 15 bit	8	10 ... 6000	
19h	Influencing quantity manipulating factor	%	± 7 bit	8	Min. ... max. manipulating factor	
1Ch	Minimum manipulating factor	%	± 7 bit	8	–100 ... 0	
1Dh	Maximum manipulating factor	%	± 7 bit	8	0 ... +100	
1Eh	Sensor error manipulating factor	%	± 7 bit	8	Min. ... max. manipulating factor	
1Fh	Switching hysteresis	0.1°	± 15 bit	8	0 ... MRS *)	

*) MRS = measuring range span

7.4 Main Group 2: Control Commands

7.4.1 Table of Parameter Indices

PI	Parameter Designation	U/M	Format	Number	Setting Range	Comment
20h	Controller function	bit	8 bit	8	See chapter 7.4.2 on page 62	
21h	Channel error status Device error status Output error	bit	16 bit 16 bit 8 bit	8 1 6	See chapter 7.4.3 on page 63	See events data
22h	Controller configuration	bit	16 bit	8	See chapter 7.4.4 on page 64	
23h	Expanded controller configuration	bit	8 bit	8	See chapter 7.4.5 on page 64	
24h	Controller status, message word	bit	16 bit	9	See chapter 7.4.6 on page 64	Read only
25h	Oscillation hold-off	0.1 s	8 bit	8	0.0 = off, 0.3 ... 25.0 s	
26h	Master actual value	0.1°	± 15 bit	4	See chapter 2.6.3 on page 16	
27h	External actual value	0.1°	± 15 bit	8	See chapter 2.3.2 on page 9	
28h	Manual manipulating factor	%	± 7 bit	8	Min. ... max. manipulating factor	In manual mode only
29h	Channel error mask	bit	16 bit	8	See chapter 7.4.7 on page 65	
2Ah	Group error mask	bit	16 bit	8	See chapter 7.4.8 on page 65	
2Ch	Alarm history, time stamp	–	16 bit	3	See chapter 7.4.9 on page 65	1) Read only, not via service interface
2Dh	Alarm history read-out starting point	–	± 15 bit	1	1 ... 100	1)
2Eh	Alarm history Time stamp, via service interface only Channel error status Device error status Output error	 –	 16 bit 16 bit 16 bit 8 bit	 3 8 1 6	See chapter 7.4.9 on page 65	1) Read only
2Fh	Number of alarm history entries	–	± 15 bit	1	1 ... 100	1) Read only

1) For detailed description refer to chapter 2.9.5 on page 27

7.4.2 Controller Function

PI = 20h or function selection for control via binary input

Bit Number	Meaning	Comment
0	Proxy setpoint active	
1	Actuator circuit	
2	Feed-forward control	1)
3	Temporary setpoint rise (Boost)	
4	Switching controller active	1)
5	Clear error	1)
6	Controller on	
7	Start self-tuning	see chapter 2.7.1

1) Device set deletes bit

7.4.3 Error Status

PI = 21h

Data are assigned just like events data.

The entry “from channel to channel” makes reference to 16 bit words, i.e.

Channel 1 ... 8	≙	channel error status 1 ... 8
Channel 9	≙	device error status
Channel 10 ... 12	≙	output error

Some errors must be acknowledged (compare tables):

This is accomplished by setting the corresponding error bits to 0. Transferred error status words (control loop, device) are linked to error status words in the controller, bit by bit, by means of AND logic, so that individual bits can be cleared in the error status word when errors are eliminated sequentially. Errors which occur during frame transmission are not cleared.

Bit Assignments for Channel Error Status

Bit Number	Meaning	Comment
0	Broken sensor	
1	Polarity reversal	
2	Second upper limit value exceeded	1) 3)
3	First upper limit value exceeded	1) 3)
4	First lower limit value fallen short of	1) 3)
5	Second lower limit value fallen short of	1) 3)
6	Impermissible parameter	2)
7	Heating current not off with deactivated actuating signal	
8	Too little heating current with active actuating signal	
9	Heating circuit error	2) 3)
10	Error starting adaptation	2) 3)
11	Adaptation error and abort	2) 3)
12	Too big heating current with active actuating signal	

1) Must be acknowledged in case of alarm memory
2) Must be acknowledged
3) Can be acknowledged via binary input

Bit Assignment for Device Error Status

Bit Number	Meaning	Comment
0	Analog component error	Error LED lights up
1	Overload, heating current 1	
2	Overload, heating current 2	
3	Overload, heating current 3	
4	Overload, heating voltage	
5	—	
6	Reference junction error	
7	EEPROM error	2) / error LED lights up
8	Group output error	Error LED lights up
9	Mapping error	2)
10	Parameter error	2)

2) Must be acknowledged

Bit Assignment Output Error 1 ... 3

Bits are set although the output is short-circuited, i.e. when the output is active but no signal is present at the terminal.

Output Error 1	
Bit Number	Output
0 ... 7	1 ... 8

Output Error 2	
Bit Number	Output
0 … 7	9 … 16

Output Error 3	
Bit Number	Output
0 ... 3	17 ... 20
4 ... 7	—

Bit Assignment Output Error 4 ... 6

Bits are set when the output is inactive, but a signal is present at the terminal.

Output Error 4	
Bit Number	Output
0 ... 7	1 ... 8

Output Error 5	
Bit Number	Output
0 … 7	9 … 16

Output Error 6	
Bit Number	Output
0 ... 3	17 ... 20
4 ... 7	—

7.4.4 Controller Configuration

PI = 22h

Bit Number	Value	Meaning	Comment
0 ... 2		**Controller Type**	
	0	Channel not in use	
	1	Measuring	
	2	Actuator	
	3	Limit transducer	
	4, 5	PDPI controller	
	6	Proportional actuator	
	7	Reserved	
3 ... 5		**Controller Class**	
	0	Fixed setpoint controller	
	1	Differential controller	
	2	Master controller	
	3	Slave controller	
	4	Switching controller	
	5 ... 7	Reserved	
6 ... 8	0 ... 7	Partner channel	For differential, slave and switching controllers
9, 10		**Group**	
	0	No group	
	1 ... 3	Group number	
11	0 / 1	Actual value control off / on	
12	0 / 1	Hot-runner off / on	
13	0 / 1	Water cooling off / on	
14	0 / 1	Adaptive measured value correction off / on	
15	0 / 1	Manual instead of off off / on	

7.4.5 Extended Controller Configuration

PI = 23h

Bit Number	Value	Meaning	Comment
0	0/1	internal / external actual value	
1	0/1	Actuating output normal / particularly for contactors	
2	0/1	Manual instead of boost on / off	compare chapter 2.5.3
3	0/1	D-component normal / attenuated	
4	0/1	PDPI- / PI controller	
5 ... 7		not in use	

7.4.6 Controller Status, Message Word

PI = 24h

Bit Number	Value	Meaning	Comment
0 ... 3	0, 1 ... 15	Optimization phase 0: no optimization	Controller status (channels 1 ... 8)
4, 5	0, 1, 2	Ramp active 0: no ramp 1: up 2: down	
6, 7	0, 1, 2	Actuation active 0: no actuation 1: actuation manipulating factor active 2: dwell time active	
8	0/1	Actual value control inactive/active	
9	0/1	1: slowest channel of the group with regard to actual value control	
10, 11	0	not in use	
12 ... 14	0 ... 7	Mapping address	
15	0/1	Mapping completed	
0 ... 7	0/1 ... 0/1	Status of the message inputs	Message word (channel 9)
8 ... 15	0	not in use	

7.4.7 Channel Error Mask

PI = 29h

Bit Number	Meaning
0	Broken sensor
1	Polarity reversal
2	Second upper limit value exceeded
3	First upper limit value exceeded
4	First lower limit value fallen short of
5	Second lower limit value fallen short of
6	Impermissible parameter
7	Heating current not off with deactivated actuating signal
8	Too little heating current with active actuating signal
9	Heating circuit error
10	Error starting adaptation
11	Adaptation error and abort
12	Too big heating current
13 ... 15	—

7.4.8 Group Error Mask

PI = 2Ah

Bit Number	Meaning
0	Broken sensor
1	Polarity reversal
2	Second upper limit value exceeded
3	First upper limit value exceeded
4	First lower limit value fallen short of
5	Second lower limit value fallen short of
6	Impermissible parameter
7	Heating current monitoring error
8	Heating circuit error
9	Adaptation error
10	Analog component error
11	Overload, heating current monitoring
12	—
13	Reference junction error
14	EEPROM error, Parameter error
15	Group output error, 24V error

7.4.9 Alarm History

PI = 2Eh

The first three words contain the time stamp (no real-time clock!) for the time at which the error status changed. The contents of the last 12 words are identical to those of the error status (PI = 21h).

The entry “from channel to channel” makes reference to 16 bit words, i.e.

Channel 1 ... 3 time stamp
Channel 4 ... 11 channel error status 1 ... 8
Channel 12 device error status
Channel 13 ... 15 output error

Due to the fact that only 12 words are transmitted with Profibus, the time stamp can be read where PI = 2Ch, whereas only the error status can be read with PI = 2Eh (same as PI = 21h).

Time stamp format where PI = 2Eh/2Ch or current time where PI = 90h:

Word / Channel	Character	Significance	Value Range	Comment
1	Low byte High byte	Second Minute	0 ... 59 0 ... 59	
2	Low byte High byte	Hour Day	0 ... 23 1 ... 31	
3	Low byte High byte	Month Year	1 ... 12 0 ... 99	

7.5 Main Group 3: Device Specification

7.5.1 Table of Parameter Indices

PI	Parameter Designation	U/M	Format	Number	Setting Range	Comment
30h	Device ID	bit	8 bit	1	60h	Read only
31h	Device features	bit	8 bit	1	See chapter 7.5.2 on page 66	Read only
32h	Device control	bit	8 bit	1	See chapter 7.5.3 on page 66	
33h	Sensor type	bit	8 bit	8	See chapter 7.2.2 on page 61	
35h	Software version	bit	8 bit	1	(e.g. 57h = V5.7)	Read only
36h	Limit value configuration	bit	8 bit	8	See chapter 7.5.4 on page 66	
37h	Output configuration I/O 1 ... 16 continuous output 1 ... 4	bit	8 bit	20	See chapter 7.5.5 on page 67	
3Ah	Power limitation	%	± 7 bit	1	0 = off, 12 ... 100%	compare chapter 2.2.5
3Fh	Parameter set ID	bit	16 bit	3	See chapter 7.5.6 on page 67	

7.5.2 Device Features

PI = 31h

Bit Number	Value	Meaning	Comment
0		**Version**	
	0	Standard version	
	1	OEM hardware and software version	
1		**Protocol of RS-232/RS-485 interface**	
	0	EN 60870	Feature F1, F2, F4
	1	Modbus	Feature F3, F6
2	0	**Protocol of bus interface**	
	0	**For CAN:** CANOpen	Feature F1
	0	**For Profibus DP:** EN 50170	Feature F2, F6
3 ... 5		**Bus interface**	
	0	RS 485 only	Feature F3, F4
	1	CAN	Feature F1
	2	Profibus DP	Feature F2, F6
6, 7		**A Features**	
	0	16 binary inputs / outputs	Feature A0
	1	20 binary inputs / outputs	Feature A1
	2	16 binary inputs / outputs, 4 continuous outputs	Feature A2

7.5.3 Device Control

PI = 32h

A code word including all 8 bits is written which starts the operation, and may stop it as well or set the parameter.
The 8 bits which are read include information regarding the operation in progress in the upper 4 bits, and the lower 4 bits contain parameters.

Write		Read		Meaning	
Bit number	Code / value	Bit number	Value		
0	0 / 1	0	0 / 1	Controlled variable quantity, °C / °F	
1	0 / 1	1	0 / 1	without / with master PDO	
2 ... 3	0	2 ... 3	0	not in use	
0 ... 7	0Fh	4 ... 7	Cannot be read back	Load factory default setting to momentary parameter set	
	1Eh			Save momentary parameter set to parameter set 1	
	1Fh			Load parameter set 1 to momentary parameter set	
	2Eh			Save momentary parameter set to parameter set 2	
	2Fh			Load parameter set 2 to momentary parameter set	
0 ... 7	55h	4 ... 7	5h	Determination of heating current nominal values	start / running
	—		0h		finished
	AAh		Ah	Check sensor/heater assignment	start / running
	AAh		0h		stop / finished

7.5.4 Limit Value Function and Heating Circuit Monitoring

PI = 36h

Bit Number	Value	Meaning
0	0 / 1	Alarm 1: setting relative/absolute to setpoint
1	0 / 1	Alarm 1: actuation suppression inactive/active
2	0 / 1	Alarm 2: setting relative/absolute to setpoint
3	0 / 1	Alarm 2: actuation suppression inactive/active
4	0 / 1	Heating circuit monitoring inactive/active
5	0 / 1	Limiter inactive / active
6	0 / 1	Alarm 1: Memory inactive / active
7	0 / 1	Alarm 2: Memory inactive / active

7.5.5 Output Configuration

PI = 37h

- If all bits = 0, the output is inactive and has no function as an input.
- The continuous output can only be configured for manipulated variable read-out.

Standard Output Configuration of an Output (bit 0 = 0, bit 1 = 1)

Bit Number	Value	Discontinuous Output Manipulated Variable	Discontinuous Output Alarm	Continuous Output
0	0	Output		
1	1	Standard		
2 ... 4	0 ... 7	Channel number		
5	0 / 1	Heating / cooling	– / –	Heating / cooling
6	0 / 1	More / less	Operating current / closed-circuit current	Dead / live zero
7	0 / 1	0 = manipulated variable	1 = alarm	Manipulated variable

Special Output Configuration of an Output (bit 0 = 0, bit 1 = 0)

Bit Number	Value	Discontinuous Output	Continuous Output
0	0	Output	
1	0	Special	
2 ... 6	0 ... 31	Output function (see page 67)	Read-out zero / Reserved
7	0 / 1	Operating current / closed-circuit current	Dead / live zero

Standard Output Configuration of an Input (bit 0 = 1, bit 1 = 1)

Bit Number	Value	Discontinuous Output	Continuous Output
0	1	Input	Output
1	1	Standard	
2 ... 4	0 ... 7	Channel number	Same as for configuration as output
5 ... 7	0 ... 7	Input function (see page 67)	

Special Output Configuration of an Input (bit 0 = 1, bit 1 = 0)

Bit Number	Value	Discontinuous Output	Continuous Output
0	1	Input	Output
1	0	Special	
2, 3	0 ... 3	Group number	Same as for configuration as output
4 ... 7	0 ... 15	Input function (see page 67)	

Output Function

Value	Meaning	Comment
0	Output deactivated	
1 ... 8	Group error 1 ... 8	
9	Adaptation in progress, or adaptation error	
10 ... 13	Group error 0 ... 3	
14, 15	Reserved	
16	Independently controllable output	also for continuous outputs
17 ... 27	Reserved	
28	Data, 3rd controller	External heating current monitoring with operating current only
29	Data, 2nd controller	
30	Cycle	
31	Acknowledgement	

Input Function

Value	Meaning	Comment
0	Proxy setpoint active	Channel control or control per group
1	Actuation circuit	
2	Feed-forward control	
3	Temporary setpoint rise (Boost)	
4	Switching controller active	
5	Clear error	
6	Controller on	
7	Start self-tuning	
8	Bit 0 of the message word (controller status channel 9) is set	Message input group number = 0
...	...	
15	Bit 7 of the message word (controller status channel 9) is set	
8 ... 11	—	Group number = 3
12	Logger stop	
13	Data external heating current monitoring	
14	Cycle external heating current monitoring	
15	Acknowledgement external heating current monitoring	

7.5.6 Parameter Set ID

PI = 3Fh

The parameter set ID consists of 3 words and can be read and written. It forms an integral part of each parameter set (bytes 19Ah...19Fh). The format is free, any value is permissible.

7.6 Main Group 6: Heating Current Monitoring

7.6.1 Table of Parameter Indices

PI	Parameter Designation	U/M	Format	Number	Setting Range	Comment
60h	Nominal heating current	0.1 A	± 15 bit	8	0 = off, 1 ... 10000	
61h	Heating current nominal value 2nd controller	0.1 A	± 15 bit	8	0 = off, 1 ... 2500	
62h	Heating current nominal value 3rd controller	0.1 A	± 15 bit	8	0 = off, 1 ... 2500	
64h	Summation current transformation ratio	0.1 A	± 15 bit	1	0 ... 10000	
67h	Heating current sampling cycle	0,1 s	± 15 bit	1	0 = Auto, 1 ... 30000	
69h	Heating voltage transformer secondary voltage	0.1 V	± 15 bit	1	0, 100 ... 500	
6Ch	Heating current - actual value	0.1 A	± 15 bit	8		read only
6Dh	Heating current - actual value 2nd controller	0.1 A	± 15 bit	8		read only
6Eh	Heating current - actual value 3rd controller	0.1 A	± 15 bit	8		read only
6Fh	Heating voltage - actual value	0.1 V	± 15 bit	1		read only

7.7 Main Group 9: Data Logger

Refer to chapter 2.9.3 on page 25 for a detailed description of the functions of the quantities.

7.7.1 Table of Parameter Indices

PI	Parameter Designation	U/M	Format	Number	Setting Range	Comment
90h	Current time (no real-time clock)	—	16 bit	3	See chapter 7.4.9 on page 65	
92h	Logger sampling cycle	0.1 s	± 15 bit	1	0.1 ... 600.0 s	
93h	Logger control	bit	8 bit	1	0/1 = Logger Run / Stop	
94h	Read-out starting point sampled actual values	—	± 15 bit	1	1 ... 3600	
95h	Read-out starting point sampled manipulated variables	—	± 15 bit	1	1 ... 3600	
96h	Sampled actual values	0.1 °	± 15 bit	(1 ... 15) x 8	MbA ... MbE	Read only 1)
97h	Sampled manipulated variables	%	± 15 bit	(1 ... 15) x 8	-100 ... 100	Read only 1)
98h	Number of samples	—	± 15 bit	1	0 ... 3600	Read only
99h	Time of last sample	—	16 bit	3	same as PI = 90h	

1) Refer to chapter 2.9.3 on page 25 for a detailed description

7.8 Main Group A: Interfaces

Interface parameters can be set with this function, however, not via Profibus.
Changes do not become effective until after reset has taken place.

7.8.1 Table of Parameter Indices

PI	Parameter Designation	U/M	Format	Number	Setting Range	Comment
A0h	Interface configuration	bit	8 bit	1		
A1h	CAN baud rate	bit	8 bit	1		A1h

7.8.2 Interface Configuration

Bit Number	Value	Meaning
0 ... 3		**Baud Rate**
	0 1 2	4800 9600 19.2 K
4 ... 6		**Parity**
	0 1 2 3	Even Odd None Space

7.8.3 CAN Baud Rate

Bit Number	Value	Meaning
0 ... 3		**Baud Rate (kB)**
	0 1 2 3 4 5 6 7 8	10 20 50 100 125 250 500 800 1000
4 ... 6	0	not in use

7.9 Main Group B: Display Values

7.9.1 Table of Parameter Indices

PI	Parameter Designation	U/M	Format	Number	Setting Range	Comment
B0h	Momentary setpoint	0.1 °	± 15 bit	8		Read only
B1h	Momentary actual value	0.1 °	± 15 bit	8		Read only
B2h	Momentary system deviation	0.1 °	± 15 bit	8		Read only
B3h	Momentary reference junction temperature	0.1 °	± 15 bit	1		Read only
B7h	Momentary manipulating factor	%	± 15 bit	8		Read only
B8h	Momentary setpoint	1 °	± 15 bit	8		Read only
B9h	Momentary actual value	1 °	± 15 bit	8		Read only
BAh	Momentary system deviation	1 °	± 15 bit	8		Read only

7.10 Main Group E: Control Functions

PI	Parameter Designation	U/M	Format	Number	Setting Range	Comment
E0	State of binary I/O's	Bit	16 bit	2	[2]	[1]
E1	State of continuous outputs	0.1%	16 bit	4	0 ... 1000	[1]

[1] When the output is configured as an „independently controllable output", the state can be written as well.
[2] Bits 0 ... 15 in Word 0 correspond to inputs/outputs 1 ... 16,
bits 0 ... 3 in Word 1 correspond to inputs/outputs 17 ... 20 of Feature A1

8 Index

9 Parameter index

10 Repair and Replacement Parts Service and Rental Instrument Service

When you need service, please contact:

GMC-I Gossen-Metrawatt GmbH
Service Center
Thomas-Mann-Strasse 20
90471 Nürnberg • Germany
Phone +49-(0)-911 86 02 - 410/256
Fax +49-(0)-911 86 02 - 2 53
E-mail service@gossenmetrawatt.com

This address is only valid in Germany.
Please contact our representatives or subsidiaries for service in other countries.

11 Product Support

When you need support, please contact:

GMC-I Gossen-Metrawatt GmbH
Product Support Hotline
Phone +49-(0)-911 86 02 - 784
Fax +49-(0)-911 86 02 - 709
E-mail support@gossenmetrawatt.com

Edited in Germany • Subject to change without notice • A pdf version is available on the Internet.

GMC-I GOSSEN METRAWATT

GMC-I Gossen-Metrawatt GmbH
Thomas-Mann-Str. 16-20
90471 Nürnberg • Germany

Phone +49-(0)-911-8602-0
Fax +49-(0)-911-8602-669
E-Mail info@gossenmetrawatt.com
www.gossenmetrawatt.com