

# **Arcus-EDS**

## **Application Description**

### **SK08-T8**



KNX Controller 8-Channel Temperature for PT1000

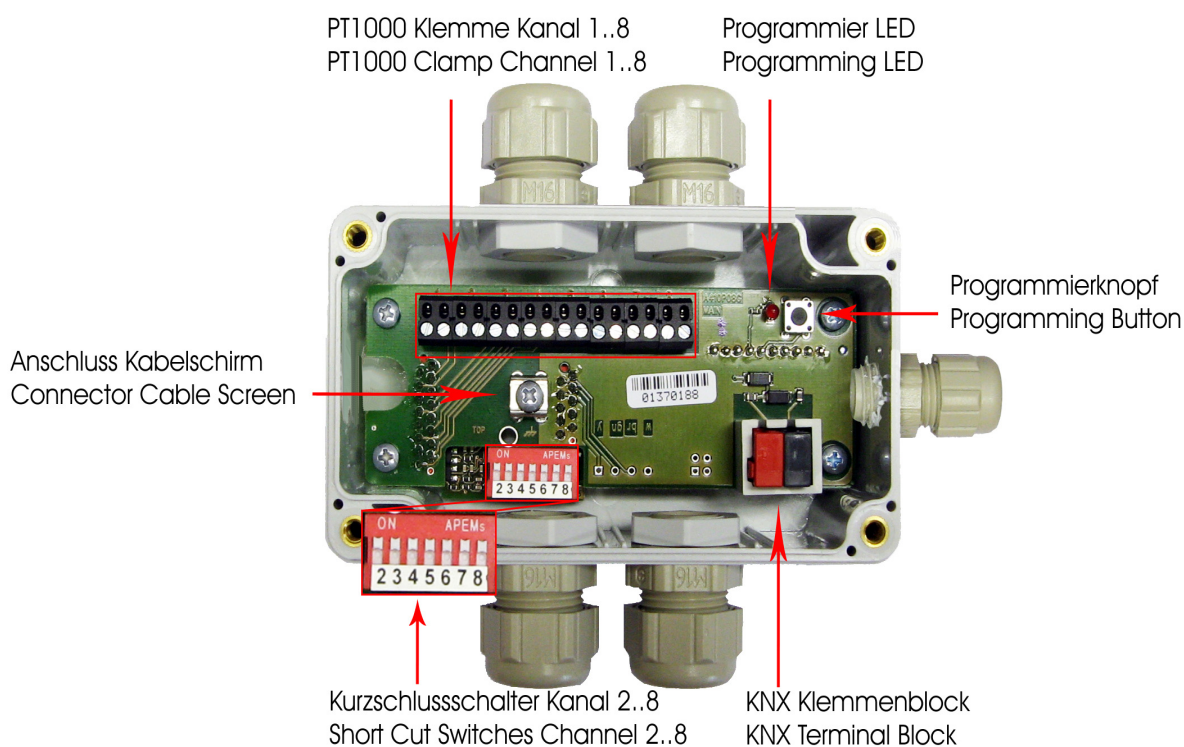
**Operating Principals and Areas of Application:**

The production series S8 uses sensors and controllers for a number of physical and chemical measurements for indoor and outdoor areas.

The measuring system SK08-T8 records the temperatures using PT1000 for 8 different measuring points. The sensors change their resistance with the temperature. This change is converted electronically and displayed on the KNX bus. Any standard temperature sensor can be used, provided it is PT1000 compatible.

A number of controller models with various functions are available.

The devices in the series MS08 come in a housing for surface mounting with PG single connection for EIB/KNX insert and several connections for the sensor cable.



## Application and Functions:

KNX sensors are set up using the ETS (KNX Tool Software) with the associated application program SK08-T8. The device is delivered unprogrammed. All functions are parameterized and programmed by ETS. The controller can be switched on or off by activation or locking via the KNX bus.

### Functions:

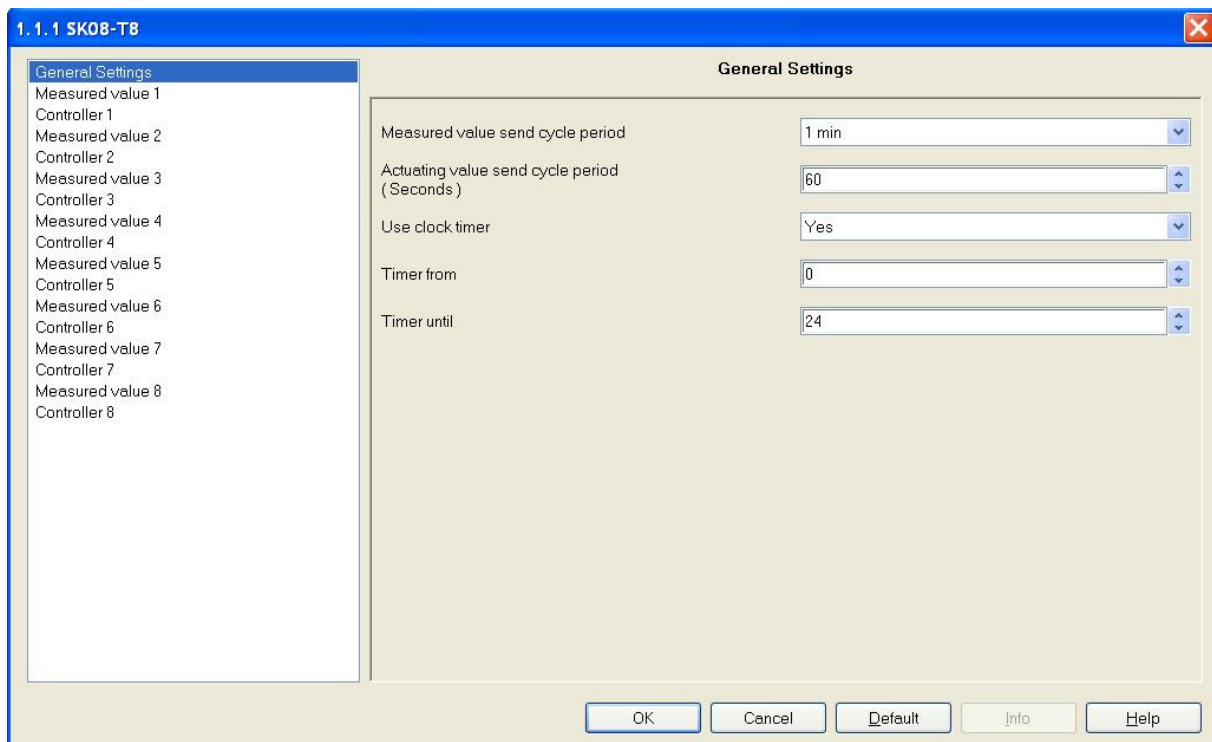
- 8 \* Temperature measurements with the following:
- Two position controller with switch and pulse 1-bit output or
- PI controller with continuous 8-bit or pulse-width modulated 1-bit output
- Adjustable periodic display of control variable : no periodic display /10-250 seconds
- Adjustable release and lock with all controllers
- Threshold alarm for upper and lower thresholds
- Auxiliary quantity of set value or threshold via the bus
- Calibration of the sensor ( offset cancellation )

### General Settings:

**Periodic Measured Data Cycle:** Measured data to be periodically displayed can be configured from a length of 1 to 120 minutes.

**Periodic Actuating Variable Cycle:** The actuating variable cycle can be set between 10 and 250 seconds.

To display the measured data periodically use the measured data settings; to display the control variable periodically use the controller settings.



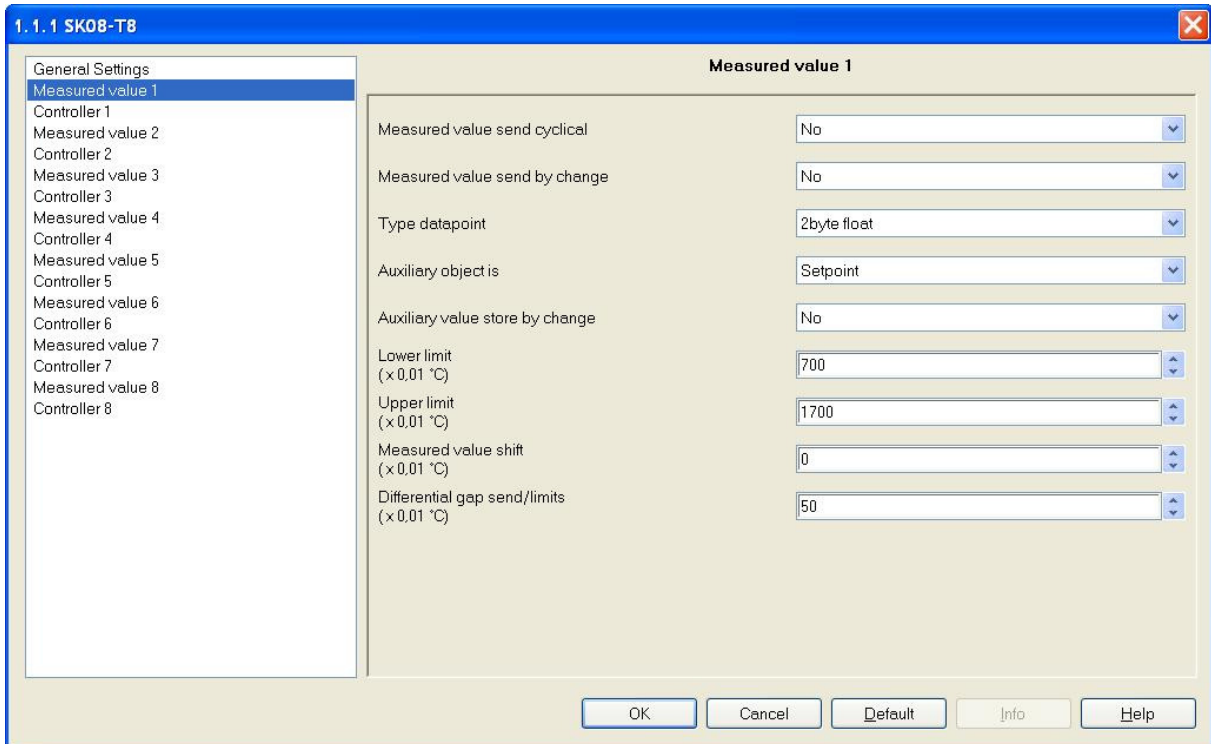
When using the **internal timer**, there are two additional objects for the system-time and -date available. Each controller can be locked independent from each other depending on the time. In this page you can set whether the timer is used at all and the span of time the controllers are active. Whether the timer is used for a controller is determined at his parameter page.

## Measured Data T1 ... T8 :

**Measured value send cyclical: Yes/No** The display period is set in General Settings.

**Measured value send by change: Yes/No** The threshold is defined in "Display Differential Gap/Threshold".

**Value Type: 1-byte Integer / 2-byte Integer / 2-byte float / 4-byte float** Measured Data Output and Auxiliary data are defined concurrently.



**Auxiliary Object is: Set point/Upper Threshold/Lower Threshold** Every controller has an auxiliary object which can control either the set point of the controller or the limit values.

**Auxiliary value store by change: Yes/No** When the auxiliary data is changed the new value is carried over to EEPROM and saved in case of a bus voltage breakdown. This should be used only when the data is not frequently changed as EEPROM has only a limited memory cycle.

**Lower Threshold: -99,99 ... +99,99 °C**

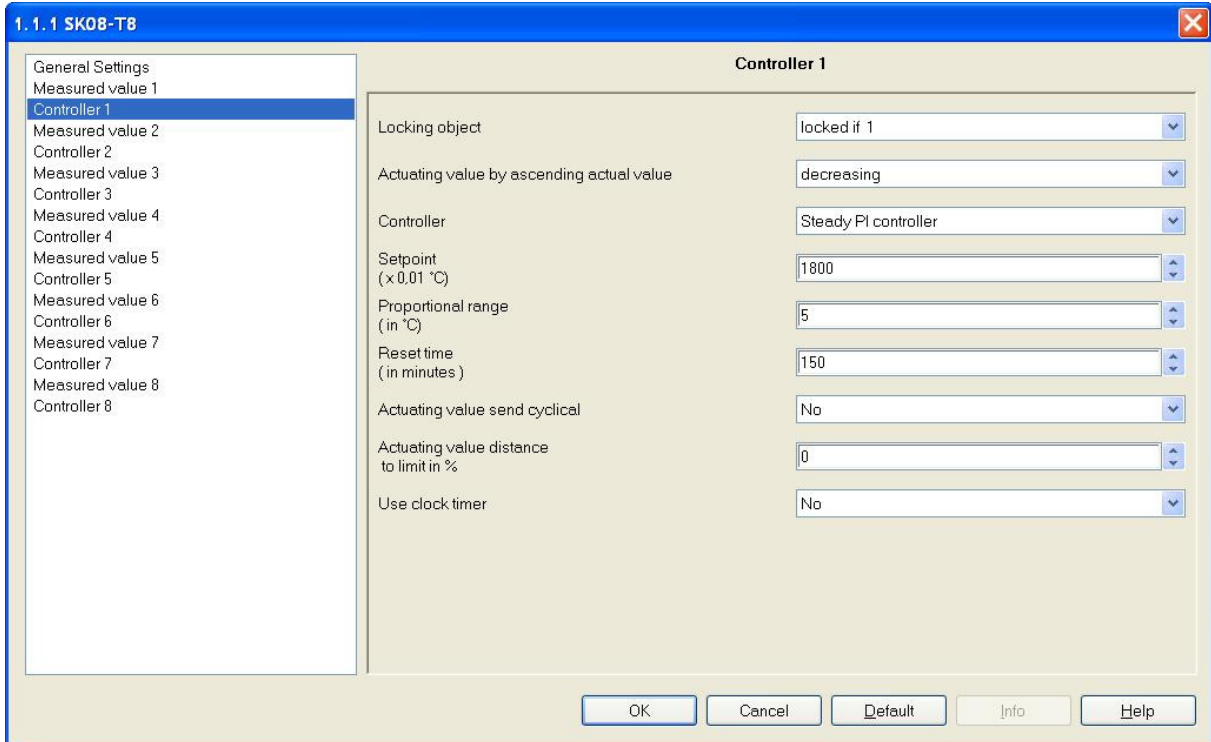
**Upper Threshold: -99,99 ... +99,99 °C**

**Measured value shift: -99,99 ... +99,99 °C** A calibration/offset adjustment of the sensors can occur when the measured displacement is offset due to cable length or other known external influences.

**Differential gap send/limits: 0 ... 10 °C** To reduce the bus load when a value is changed and to avoid multiple switching between measured data and thresholds, a hysteresis between 0,1 and 1°C should be used.

## Controller T1 ... T8 :

**Locking object: locked if 0/locked if 1** When using the lock function the controller output is deactivated. The lock function can be set up for "release" or "lock".



**1.1.1 SK08-T8**

**Controller 1**

|                  |                                           |                      |
|------------------|-------------------------------------------|----------------------|
| General Settings | Locking object                            | locked if 1          |
| Measured value 1 | Actuating value by ascending actual value | decreasing           |
| Controller 1     | Controller                                | Steady PI controller |
| Measured value 2 | Setpoint (x 0.01 °C)                      | 1800                 |
| Controller 2     | Proportional range (in °C)                | 5                    |
| Measured value 3 | Reset time (in minutes)                   | 150                  |
| Controller 3     | Actuating value send cyclical             | No                   |
| Measured value 4 | Actuating value distance to limit in %    | 0                    |
| Controller 4     | Use clock timer                           | No                   |
| Measured value 5 |                                           |                      |
| Controller 5     |                                           |                      |
| Measured value 6 |                                           |                      |
| Controller 6     |                                           |                      |
| Measured value 7 |                                           |                      |
| Controller 7     |                                           |                      |
| Measured value 8 |                                           |                      |
| Controller 8     |                                           |                      |

OK Cancel Default Info Help

**Actuating value by ascending actual value: decrease/increase** The Actuating direction of the controller can be adapted to the characteristics of the controlled system.

**Set point: -99,99 ... +99,99 °C**

**Controller: Two-position Controller / Pulsed Two-position Controller / Continuous PI Controller / Switching PI Controller** These controller models and their applicable parameters are covered in the section "Controller Algorithms".

**Actuating value send cyclical: Yes/No** The cycle period is set in "General Settings"

**Actuating value distance to limit in %: 0...50** When the lower threshold is surpassed 0% is set, when the upper threshold is surpassed 100% will be set. This is important for actuators which do not operate reliably at threshold levels.

**Use clock timer: Yes/No** The use of the clock timer can be enabled/disabled for each channel separately.

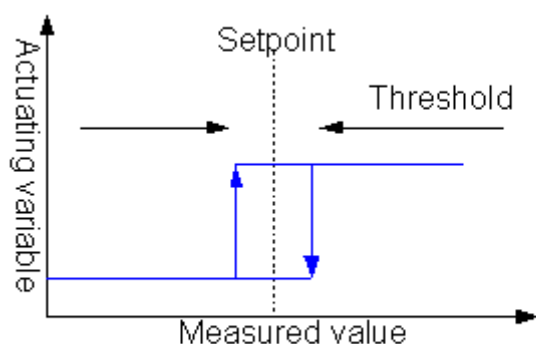
### Controller Algorithms:

Controller models available are the PI controller or a two-position controller. Both controllers are equipped with pulsed output. The pulsed two-position controller works with constant duty cycle, which like the cycle duration is parameterized. The duty cycle of the pulsed PI controller is variable and depends on the control variable (pulse-width modulation).

### Two-Position Control:

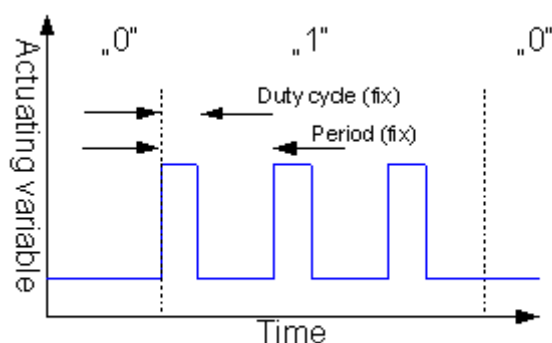
Two-position control is a very simple way of controlling. Once the actual value (+/- half the switching difference) exceeds or falls below the set point a switch-on or switch-off command is sent to the bus. Set the differential gap large enough to keep bus load to a minimum and configure the differential gap small enough to avoid extreme actual value fluctuations.

The two-position controller is parameterized using the set point and the switching threshold.



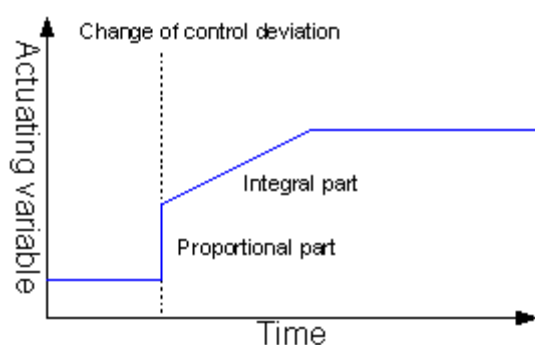
### Two-Position Control with Pulsed Output:

The controller works analogous to the two-position controller, but the actuating variable emits pulses with fixed duty cycle.



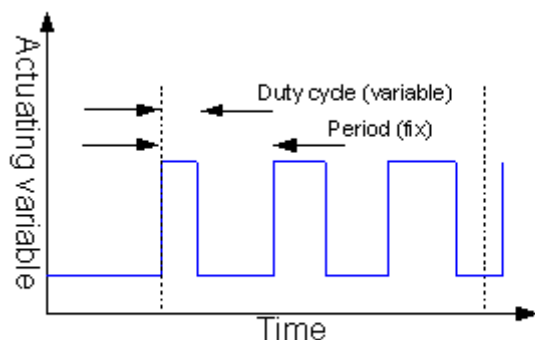
### Continuous PI Control:

To understand a PI controller one should think of an algorithm consisting of a proportional and integral part. By combining these two parts it is possible to get a quick and exact adjustment of the actuating variable. The controller calculates the control variable every second. It can constantly be updated and is displayed periodically (value parameterized) by the PI controller. Through the integral part an offset is adjusted to 0 over a certain period of time.



### Continuous PI Control with Pulsed Output (PWM):

The controller works analogous to the PI controller, but the actuating variable emits pulses with a variable duty cycle. PWM control sets the cycle duration of the transmission interval. This allows a permanent on and off within the cycle time with object 15, which reaches an average valve position. When the control variable reaches 40% in a cycle time of 10 minutes it will repeatedly turned on for 4 minutes and turned off for 6 minutes.



### General Rules for Adjusting the PI Parameter:

The reset time must be significantly larger than the delay time of the control system. The proportional area corresponds to the reinforcement of the control circuit. The smaller the proportional area, the larger the reinforcement is.

| Parameters             | Effect                                                                                               |
|------------------------|------------------------------------------------------------------------------------------------------|
| Low Proportional Area  | Large overshooting of set point balance (potential for constant oscillation ), quick set point reset |
| High Proportional Area | Little or no overshooting, but slow reset                                                            |
| Short Integration Time | Quick adjustment of control deviations (based on conditions) danger of constant oscillation          |

|                       |                                       |
|-----------------------|---------------------------------------|
| Long Integration Time | Slow adjustment of control deviations |
|-----------------------|---------------------------------------|

**Object Table for Application SK08-T8:**

| Number | Name                         | Object Function  | Length |
|--------|------------------------------|------------------|--------|
| 0      | Output, sensor status sho... | Output status    | 1 Byte |
| 1      | Output, sensor status break  | Output status    | 1 Byte |
| 2      | Output, measured value ...   | Measured value   | 2 Byte |
| 3      | Input, auxiliary object K1   | Auxiliary object | 2 Byte |
| 4      | Output, upper limit K1       | Limit            | 1 bit  |
| 5      | Output, lower limit K1       | Limit            | 1 bit  |
| 6      | Output, controller K1        | Actuating value  | 1 Byte |
| 7      | Input, enable/lock K1        | Enable/lock      | 1 bit  |
| 8      | Output, Object status K1     | Channel status   | 1 Byte |
| 9      | Output, measured value ...   | Measured value   | 2 Byte |
| 10     | Input, auxiliary object K2   | Auxiliary object | 2 Byte |
| 11     | Output, upper limit K2       | Limit            | 1 bit  |
| 12     | Output, lower limit K2       | Limit            | 1 bit  |
| 13     | Output, controller K2        | Actuating value  | 1 Byte |
| 14     | Input, enable/lock K2        | Enable/lock      | 1 bit  |
| 15     | Output, Object status K2     | Channel status   | 1 Byte |
| 16     | Output, measured value ...   | Measured value   | 2 Byte |
| 17     | Input, auxiliary object K3   | Auxiliary object | 2 Byte |
| 18     | Output, upper limit K3       | Limit            | 1 bit  |
| 19     | Output, lower limit K3       | Limit            | 1 bit  |
| 20     | Output, controller K3        | Actuating value  | 1 Byte |
| 21     | Input, enable/lock K3        | Enable/lock      | 1 bit  |
| 22     | Output, Object status K3     | Channel status   | 1 Byte |
| 23     | Output, measured value ...   | Measured value   | 2 Byte |
| 24     | Input, auxiliary object K4   | Auxiliary object | 2 Byte |
| 25     | Output, upper limit K4       | Limit            | 1 bit  |
| 26     | Output, lower limit K4       | Limit            | 1 bit  |
| 27     | Output, controller K4        | Actuating value  | 1 Byte |
| 28     | Input, enable/lock K4        | Enable/lock      | 1 bit  |
| 29     | Output, Object status K4     | Channel status   | 1 Byte |
| 30     | Output, measured value ...   | Measured value   | 2 Byte |
| 31     | Input, auxiliary object K5   | Auxiliary object | 2 Byte |
| 32     | Output, upper limit K5       | Limit            | 1 bit  |
| 33     | Output, lower limit K5       | Limit            | 1 bit  |
| 34     | Output, controller K5        | Actuating value  | 1 Byte |
| 35     | Input, enable/lock K5        | Enable/lock      | 1 bit  |
| 36     | Output, Object status K5     | Channel status   | 1 Byte |
| 37     | Output, measured value ...   | Measured value   | 2 Byte |
| 38     | Input, auxiliary object K6   | Auxiliary object | 2 Byte |
| 39     | Output, upper limit K6       | Limit            | 1 bit  |
| 40     | Output, lower limit K6       | Limit            | 1 bit  |
| 41     | Output, controller K6        | Actuating value  | 1 Byte |
| 42     | Input, enable/lock K6        | Enable/lock      | 1 bit  |
| 43     | Output, object status K6     | Channel status   | 1 Byte |
| 44     | Output, measured value K7    | Measured value   | 2 Byte |
| 45     | Input, auxiliary object K7   | Auxiliary object | 2 Byte |
| 46     | Output, upper limit K7       | Limit            | 1 bit  |
| 47     | Output, lower limit K7       | Limit            | 1 bit  |
| 48     | Output, controller K7        | Actuating value  | 1 Byte |
| 49     | Input, enable/lock K7        | Enable/lock      | 1 bit  |
| 50     | Output, Object status K7     | Channel status   | 1 Byte |
| 51     | Output, measured value ...   | Measured value   | 2 Byte |
| 52     | Input, auxiliary object K8   | Auxiliary object | 2 Byte |
| 53     | Output, upper limit K8       | Limit            | 1 bit  |
| 54     | Output, lower limit K8       | Limit            | 1 bit  |
| 55     | Output, controller K8        | Actuating value  | 1 Byte |
| 56     | Input, enable/lock K8        | Enable/lock      | 1 bit  |
| 57     | Output, Object status K8     | Channel status   | 1 Byte |
| 58     | Equipment time               | Time             | 3 Byte |
| 59     | Equipment date               | Date             | 3 Byte |

The measured value objects change their width with the chosen type

**The Status Functions 0 und 1 are coded as follows:**

| Sensor Number | Bit Number | Hexadecimal Value |
|---------------|------------|-------------------|
| 1             | 0          | 0x01              |
| 2             | 1          | 0x02              |
| 3             | 2          | 0x04              |
| 4             | 3          | 0x08              |
| 5             | 4          | 0x10              |
| 6             | 5          | 0x20              |
| 7             | 6          | 0x40              |
| 8             | 7          | 0x80              |

The values of the individual bits are added and transmitted to the bus. Short circuits are tolerated and you have to short-circuit the channel if it is not used. Interruptions are not tolerated and will lead to a distortion in the measured data of the other channels. If several interruptions appear, all short circuit switches should be set and then resolved one by one until the interrupted sensor is found.

**The Status Functions 8/15/22/29/36/43/50/57 are coded as follows:**

| Description                         | Bit Number | Hexadecimal value |
|-------------------------------------|------------|-------------------|
| Upper Threshold Exceeded            | 0          | 0x01              |
| Lower Threshold Surpassed           | 1          | 0x02              |
| Actuating Variable does not equal 0 | 2          | 0x04              |
| Lock Active                         | 4          | 0x08              |
| Save Auxiliary Quantity             | 5          | 0x10              |

The values of the individual bits are added and transmitted to the bus. The status functions monitor the controller status for purposes of reporting and troubleshooting.

**Imprint:**

Publisher: Arcus-EDS GmbH, Rigaer Str. 88, 10247 Berlin

Responsible for Content: Hjalmar Hevers, Reinhard Pegelow

Reprints, including partial reprints, can be made only with expressed permission from Arcus-EDS GmbH. This information is the best to our knowledge and is without guarantee. We reserve the right to make any technical as well as price changes at any time.

**Accountability:**

The selection of devices and the determination of the suitability of the devices for a specific purpose lie fully in the hands of the said buyer. For this we give no guarantee and do not accept accountability. The data in the catalogue and data sheets do not promise special properties, but instead are derived from experience and measurements. Arcus excludes responsibility for damage done on the part of the customer due to improper operation/projecting or malfunctions. On the contrary, the operator/projector has to make sure that improper operation, and projection and malfunctions do not lead to any further damage.

**Safety Guidelines:**

Attention! Installing and assembling electrical devices must only be done by an electronics specialist. The customer should be aware of and adhere to the safety guidelines of VDE, TÜV and the appropriate energy provider. Our guarantee does not include defects and damage caused by improper use or non-compliance of operating instructions.

**Warranty:**

We provide a warranty as required by law. Please contact us in case of malfunction and send the device with a full description of the fault to the address below.

**Manufacturer:****Incorporated Trademarks:**

The CE Trademark is an unofficial market trademark used exclusively by authorities and provides no warranty of properties.



Incorporated trademark of Konnex Association