FuehlerSysteme eNET International
The Brand for Sensor Technology

System Description and Configuration

Modbus RTU

Firmware 10120

Table of Contents

1 System Specific Register Structure

Used abbreviations:

Rreg	(read) read register (cannnot be written)
Rreg_mw	Rreg with specific measured values
RWreg	(read/write) read- and write registers (can also be written, sometimes with storage)
Wreg	(write) write register (can only be written, no storage)
Fcode	function code
Adr	address
Reg	register
H_Byte	(high) Byte (the more significant 8 bits; 0xHH00)
L_Byte	(low) Byte (the lowest 8 bits; 0x00LL)
0x0000	Number in 16 bit HEX format (2 Byte)
0x00	Number in 8 bit HEX format (1 Byte)

2 Changes / Software Updates

2.1 Software Version since 11/18/2019

Register for particulate matter added Rreg_16/17

2.2 Software Version since 09/01/2019

System for flush-mounted added i.a. RWreg_39
Differential pressure / volume flow calculation (air density / pressure)
DIN rail – actuators added
Address-DIP corrected

2.3 Software Version since 07/01/2019

Humidity measurement:

RWreg_07: air pressure specification for correcting the mixing ratio
RWreg_08: humidity correction (-100...+100 => Offset)
enter actual value of humidity (>100) => (autom. offset calculation -100…+100)
RWreg_09: temperature correction (-100...100 => Offset)
enter actual value of temperature (>100) =>(autom. offset calculation -100...+100)

Pressure measurement:

RWreg_28: offset atm. sensor (-50..50 =>offset)
enter atm. actual value (>900...1050) => (autom. offset calculation -50..+50)

2.4 Software Version since 09/01/2018

Pressure measurement:
RWreg_26 new function for setting zero point
RWreg_25 incline correction (by using he key)
Set the zero point also using the key/display
(see „Special features with differential pressure")

RWreg_59:
Bugfix: Entering delete

VOC:
Set the zero point also by using he key
(see "Special features of VOC measurement")

CO2:
Set the zero point also by using he key
(see „ Special features of CO2 measurement")

RWreg_90 to _99:
inserted for free use as a buffer

Access to master register: Bug fixing

Extension of the master command structure: see description „Master-Operation“

3 Register Assignment for Sensor Systems

The following register structure is specific for all FuehlerSysteme sensor systems with Modbus. Register contents are measured values as well as device-specific data for calibration or for defining the mode of operation.
Sometimes the registers are assigned not only measured values, but also fixed units. This ensures a clear display of the measured values, including the associated units, on a device-specific display or in the overall system.

Note: the specified range of values within the following register description does not simultaneously reflect the measuring range of the sensor system (observe the specific device description). The use of the register also depends on the measuring system (see also system code).

3.1 Read Register

Rreg Nr. (Fcode 0x04)	***Range of values***	***Assigned size and partly unity***	***Measured value property***
00	0 ... 0xffff		see used system code
01	0 ... 999	0.0 ... 99.9 % r.h.	relative humidity (with decimal point)
02	-999 ... 2999	-99.9 ... 299.9 °C	temperature (with decimal point)
03	0 ... 999	0 ... 99.9 g/m^3	absolute humidity (with decimal point)
04	0 ... 999	0 ... 99.9 g/kg	mixed ratio (with decimal point)
05	-999 ... 999	-99.9 ... 99.9 °C	dew point temperature (with decimal point)
06	-999 ... 999	-99.9 ... 99.9 °C	wet bulb temperature (with decimal point)
07	0 ... 999	0 ... 99.9 kJ/kg	enthalpy (with decimal point)
10	0 ... 9999	0 ... 9999 ppm	CO2 concentration
11	0 ... 999	0 ... 99.9 %	VOC contamination (with decimal point)
12	0 ... 9999	0 ... 9999 ppm	CO concentration
13	0 ... 9999	0 ... 99.9 V	voltage value in Volt (with decimal point)
14	0 ... 9999	0 ... 99.9 mA	current value in mA (with decimal point)
15	0 ... 999	0 ... 99.9 Vol%	oxygen in Vol% (with decimal point)
16	0 ... 9999	0 ... 9999 µg/m3	particulate matter > 2,5 µm
17	0 ... 9999	0 ... 9999 µg/m3	particulate matter > 10 µm
19	0 ... 0x007f	Bit_2 ... Bit_0 Bit_6 ... Bit_4	value ‚1‘: CO2 (Bit_0); VOC (Bit_1); O2 (Bit_3) the sensor is being calibrated value ‚1‘: CO2 (Bit_4); VOC (Bit_5); O2 (Bit_6) on auto-calibration
20	750 ... 1150	750 ... 1150 mbar	atmospheric pressure
21	750 ... 1500	750 ... 1500 mbar	barometric pressure
22	-9999 ... 9999	-999.9 ... 999.9 Pa	differential pressure (with decimal point)
23	-9999 ... 9999	-9999 ... 9999 Pa	differential pressure
30	0 ... 999	0 ... 999 Lux	brightness low range
31	0 ... 999	0 … 99.9 kLux	brightness high range (with decimal point)
35	0 ... 1	motion symbol	no motion / motion detected
36	0 ... 1	door symbol	door: CLOSED / OPEN
37	0 ... 1	window symbol	window: CLOSED / OPEN
38	0 ... 1	switch symbol	switch: CLOSED / OPEN

39	0 ... 999	0..999 (poti)	potentiometer value for some flush-mounted devices
40	0 ... 9999	0 ... 9999 m^3/h	volume flow
41	0 ... 9999	0 ... 9999 m^3/min	volume flow
42	0 ... 9999	0 ... 9999 m^3/s	volume flow
43	0 ... 9999	0 ... 9999 l/h	volume flow
44	0 ... 9999	0 ... 9999 l/min	volume flow
45	0 ... 9999	0 ... 9999 l/s	volume flow
47	0 ... 200	0 ... 20.0 m/s	flow (with decimal point)
50	0 ... 1	0 / 1	circuit board relay: OFF / ON
55	According to the assigned Rreg	see RWreg_57	1.minimum value
56	„	see RWreg_57	1.maximum value
57	„	see RWreg_58	2.minimum value
58	„	see RWreg_58	2.maximum value
60	-9999... 29999	-99.99 ... 299.99 °C	Special register: temperature (with two decimal places)
78	0 ... 65535	device number	
79	0 ... 65535	software version	e.g. 01018 for 01/01/2018
80	0 ... 0xffff	Error Code	error coding (bits are set in the event of an error)
100	0 ... 0xffff	special register for measuring values	see Master-Function (also writable with Wreg_500)
101	0 ... 0xffff	special register for customer code	see Master-Function (also writable with Wreg_501)
102 to 139	0 ... 0xffff	special register ongoing (see 100,101)	see Master-Function (also writable with Wreg_502 to 539)

Note:

- Rreg_mw is highlighted in gray

3.2 Read/Write Register

Rwreg Nr. (Fcode: 0x03, 0x06)	***Range of values***	***Assigned size and partly unity***	***Measured value property***
00	**0 ... 0x7aff [0x5300]**	**H_Byte: character [R] L_Byte: number [0]**	**customer code: for free assignment e.g. room code R000***
01	0 ... 999 [11111]	0.0 ... 99.9 % r.h.	relative humidity
02	-999 ... 2999 [11111]	-99.9 ... 299.9 °C	temperature
03	0 ... 999 [11111]	0 ... 99.9 g/m^3	absolute humidity
04	0 ... 999 [11111]	0 ... 99.9 g/kg	mixed ratio
05	-999 ... 999 [11111]	-99.9 ... 99.9 °C	dew point temperature
06	-999 ... 999 [11111]	-99.9 ... 99.9 °C	wet bulb temperature
07	750 ... 1500 [1013]	750 ... 1500 mbar	specify atm. air pressure (Influence on mixing ratio & CO2)
08	**-100 ... 100 [0]**	**-10.0 ... 10.0 %r.h.**	**offset for rel. Humidity measurement ***
09	**-100 ... 100 [0]**	**-10.0 ... 10.0 °C**	**offset for temperature measurement ***
10	0 ... 9999 [11111]	0 ... 9999 ppm	CO2
11	0 ... 999 [11111]	0 ... 99.9 %	VOC
12	0 ... 9999 [11111]	0 ... 9999 ppm	CO
13	0 ... 9999 [11111]	0 ... 99.9 V	voltage value
14	0 ... 9999 [11111]	0 ... 99.9 mA	current value
15	0 ... 999 [11111]	0 ... 99.9 Vol%	oxygen
16	0 ... 9999 [11111]	0 ... 9999 µg/m3	particulate matter > 2,5µm
17	0 ... 9999 [11111]	0 … 9999 µg/m3	particulate matter > 10µm

18	**0 ... 2 [1]**	**0, 1, 2**	**VOC amplification: ‚0' low; ‚1' middle; ‚2' high** *
19	**0 ... 0x007f [v]**	**Bit_6 to Bit_4** Bit_3 to Bit_0 **set accordingly**	Bit_0 = 1: CO_2 calibr. ON; Bit_1 = 1: VOC calibr. ON Bit_2 = 1: O_2 calibr. ON (are reset upon completion) **Bit_4 = 0/1: CO_2-AUTO calibr. OFF/ON** **Bit_5 = 0/1: VOC-AUTO calibr. OFF/ON** **Bit_6 = 0/1: O_2-AUTO calibr. OFF/ON** *
20	750 ... 1150 [11111]	750 ... 1150 mbar	specify atmospheric pressure
21	750 ... 1500 [11111]	750 ... 1500 mbar	specify barometric pressure
22	-9999 ... 9999 [11111]	-999.9 ... 999.9 Pa	differential pressure
23	-9999 ... 9999 [11111]	-9999 ... 9999 Pa	differential pressure
24	**0 ... 1999 [75]**	**0 ... 1999**	**k-factor for volume flow calculation** *
25	**800 ... 1200 [1000]**	**800 ... 1200**	**slope for pressure measurement (factor 0,800 to 1,200)**
26	**0, 1 [0]**	**-100 ... 100** 1	**zero offset for pressure measurement (0 => delete value)*** at 1 => one-time setting of the zero point offset
27	**1 ... 50 [10]**	**1 ... 50**	**damping for pressure measurement (number of measured values for averaging)** *
28	**-50 ... 50 [0]**	**-50 ... 50 mbar**	**offset for atm. air pressure measurement** *
29	**0 ... 3000 [0]**	**0 ... 3000 m**	**height above zero (sea level)** *
30	0 ... 999 [11111]	0 ... 999 Lux	brightness
31	0 ... 999 [11111]	0 ... 99.9 kLux	brightness
32	**0 ... 1500 [60]**	**0 ... 1500 sec.**	**follow-up time for motion detection** *
33	**0 ... 999 [11111]**	**0 ... 999**	**motion detection only below this brightness (not active with 11111)** *
34	**30, 31 [30]**	**30, 31**	**RWreg_33 refers to Rreg_30 od. 31** *
35	0 ... 1 [11111]	11111 ‚0' NO / ‚1' YES	no motion given preset motion; is reset at the follow-up time
36	0 ... 1 [11111]	11111 ‚0' CLOSE / ‚1' OPEN	no door switching specified preset door switch
37	0 ... 1 [11111]	11111 ‚0' CLOSE / ‚1' OPEN	no window switching specified preset window switch
38	0 ... 1 [11111]	11111 ‚0'CLOSE / ‚1'OPEN	no switching contact specified preset switching contact
39	0 ... 999 [11111]	0 ... 999 (poti)	potentiometer value
40	0 ... 9999 [11111]	0 ... 9999 m^3/h	volume flow
41	0 ... 9999 [11111]	0 ... 9999 m^3/min	volume flow
42	0 ... 9999 [11111]	0 ... 9999 m^3/s	volume flow
43	0 ... 9999 [11111]	0 ... 9999 l/h	volume flow
44	0 ... 9999 [11111]	0 ... 9999 l/min	volume flow
45	0 ... 9999 [11111]	0 ... 9999 l/s	volume flow
47	0 ... 200 [11111]	0 ... 20.0 m/s	flow
48	**0 ... 9999 [0]**	**0 ... 9999 cm^2**	**specify cross-sectional area for volume calculation** *
50	0 ... 1 [11111]	11111 ‚0' OFF / ‚1' ON	relay control according to Reg-Progr. (RWreg_51) circuit board relay: Fixed OFF / ON
51	**0 ... 255 [v]**	**Rreg: 0 ... 255**	**assign circuit board relay to an Rreg_x** (zero => none) * only measured value register (Rreg_mw)
52	**-9999 ... 9999 [v]**	**value**	**OFF switching value for circuit board relay** *
53	**-9999 ... 9999 [v]**	**value**	**ON switching value for circuit board relay** *
54	**0 ... 1800 [0]**	**0 ... 1800 sec.**	**switch-off delay time** * **The OFF condition must be fulfilled for as long**
55	**0 ... 1800 [0]**	**0 ... 1800 sec.**	**switch-on delay time** * **ON condition must be fulfilled for as long**
57	**0 ... 255 [0]**	**Rreg: 0 ... 255**	**1. Min / Max analysis for Rreg_x** (zero => none) * only measured value register (Rreg_mw)
58	**0 ... 255 [0]**	**Rreg: 0 ... 255**	**2. Min / Max analysis for Rreg_x** (zero => none) * only measured value register (Rreg_mw)

59	**1 ... 24 [6]**	**1, 6, 12, 24 h**	**interval time for min / max analysis** * when entering values => deleting the interval values [is reset])
60	**0 ... 3 [2]**	**0 ... 3**	**display direction of view (0, 2 horizontal, 1, 3 vertical)** *
61	**0 ... 63 [24]**	**0 ... 63**	**display contrast** *
62	**0 ... 1 [1]**	**‚0' OFF; ‚1' ON**	**display backlight** *
63	0 ... 1 [0]	‚0' OFF; ‚1' ON	priority of the display value (RWreg_75) as a single value
64	**1 ... 3 [v]**	**1 ... 3**	**simultaneous display values in the display** *
65	**1.. 60 [0]**	**1 ... 9, 10 ... 60 sec.**	**assignment of the following display value (1 ... 9). from value 10, the display values rotate in seconds. (from 1 to the last active)** *
66	**0 ... 255 [v]**	**Rreg: 0 ... 255**	**1. Display value Rreg_x assignment** (zero => not active) only measured value register (Rreg_mw)
67	**0 ... 255 [v]**	**Rreg: 0 ... 255**	**2. Display value Rreg_x assignment** (see RWreg_66) *
68	**0 ... 255 [v]**	**Rreg: 0 ... 255**	**3. Display value Rreg_x assignment** (see RWreg_66) *
69	**0 ... 255 [0]**	**Rreg: 0 ... 255**	**4. Display value Rreg_x assignment** (see RWreg_66) *
70	**0 ... 255 [0]**	**Rreg: 0 ... 255**	**5. Display value Rreg_x assignment** (see RWreg_66) *
71	**0 ... 255 [0]**	**Rreg: 0 ... 255**	**6. Display value Rreg_x assignment** (see RWreg_66) *
72	**0 ... 255 [0]**	**Rreg: 0 ... 255**	**7. Display value Rreg_x assignment** (see RWreg_66) *
73	**0 ... 255 [0]**	**Rreg: 0 ... 255**	**8. Display value Rreg_x assignment** (see RWreg_66) *
74	**0 ... 255 [0]**	**Rreg: 0 ... 255**	**9. Display value Rreg_x assignment** (see RWreg_66) *
75	**0 ... 255 [0]**	**Rreg: 0 ... 255**	**priority Display value Rreg_x assignment** (see RWreg_66)*
79	0 ... 0xffff [0]	10 20	restart restart with factory settings
80	**0 ... 9999 [0911]**	**0 ... 9999**	**servicecode (password) for settings via display***
90 to 99	0 ... 0xffff	special register	for free use (as a buffer with read / write function)
200 to 239	**0 ... 0xffff**	**special register**	**see Master-Function** *

Note:

- ***** Values entered **(bold)** are also saved **(Caution: do not write continuously!)**
- [x] Value after switching on or with presetting (factory setting)
- [v] Value with default setting (factory setting) - depending on the device type
- OFF switching value RWreg_52 less than ON switching value RWreg_53
- Customer code
  High_Byte: ASCII-characters A ... Z [0x41 ... 0x5a], a ... z [0x61 ... 0x7a]
  LOW_Byte: Number [0 ... 255(0xff)]
- Error Code
  Bit_0: humidity/temperature sensor
  Bit_1: CO2 sensor
  Bit_2: VOC sensor
  Bit_10: particulate matter sensor

3.3 Examples

For the application of some writable RWreg e.g. RWreg_01:

The default value on RWreg_01 is 11111 (0x2b67).
If the Rreg_01 is read out, the calculated value of the measuring system is located here (if it is a humidity measuring system, otherwise the value is zero).
If a value is entered in the RWreg_01 via the Modbus Master, this will overwrite the calculated value of the measuring system and adopt it as a fixed value in the Rreg_01.
This can be used, for example, to enable a display of a measuring system, to show the values of another system (external values) and also to assign a defined unit.

Options for display presentations

If the Rreg has a defined size and unit stored in the display value assignment RWreg_66 to 75 (use of Rreg_mw), this unit is also shown on the display.
Example:

RWreg_64 = 2 (two values are shown in the display)
RWreg_65 = 1 (start at 1. Display value {no rotation of display values})
RWreg_66 = 1 (the Rreg_1 - relative humidity with the unit % - is displayed
RWreg_67 = 2 (the Rreg_2 - temperature with the unit °C - is displayed

Another possibility is to set the display assignment to Rreg_100, 102 etc. These registers are written in the master mode or via Wreg_500 etc. In the following registers (Rreg_101, 103 etc.) the customer codes (room code) must be written or entered automatically via the master mode. This code is then also shown in the display behind the external symbol.
The type of measured values or the unit and decimal place must be defined via the low byte of RWreg_200, 202 etc. According to the tabular assignment of Rreg_mw.
Using this variant of the display assignment, it is possible, for example, to show several temperature measurements from various connected systems at the same time.

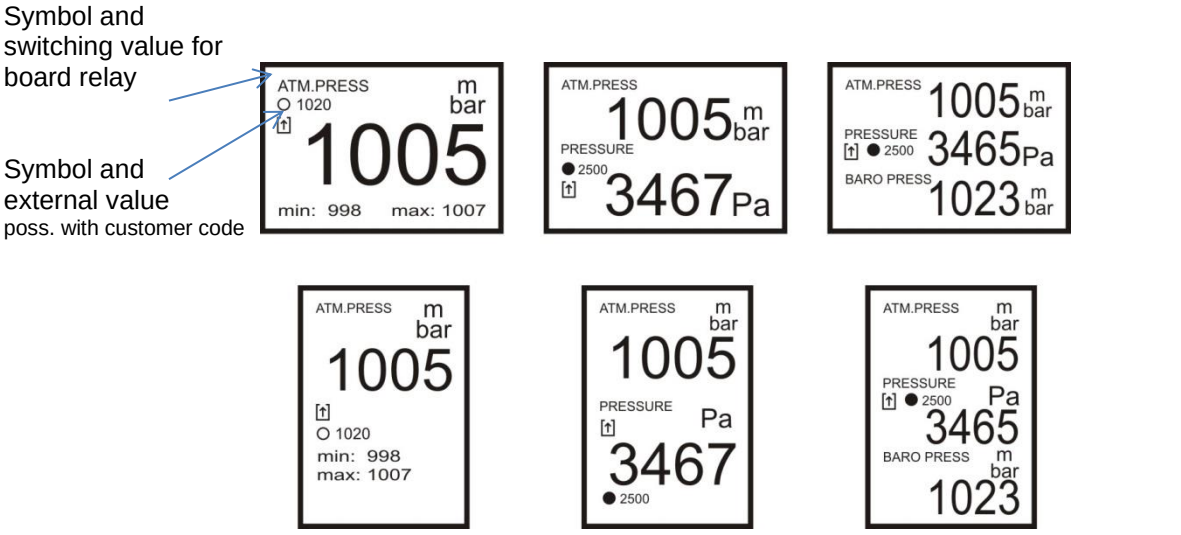
Priority display value:

If the RWreg_63 is written with 1, the measured value assignment via the RWreg_75 is shown as a single value on the display. The other settings RWreg_64 to _74 do not work until then.
Example: RWreg_75 = 37 (value of Rreg_37 window contact is shown when RWreg_63 = 1)

Display format:

- Direction of view for display via RWreg_60
- Number of values displayed simultaneously via RWreg_64 [1, 2, 3]
- Displayed values in this example

  RWreg_66 = 20 (atmospheric air pressure)
  RWreg_67 = 23 (differential pressure measuring range up to 5000Pa, without decimal point)
  RWreg_68 = 21 (barometric air pressure)



Schematic representation of the display functions

Relays are only displayed, if a relay function is programmed for the displayed measured value and the device supports a relay function.
In the display example above with two or three lines,

RWreg_51 = 23 (relay refers to the differential pressure)
RWreg_53 = 2500 (switch-on value for relay threshold)

Filled circle - relay switched; empty circle - relay not switched

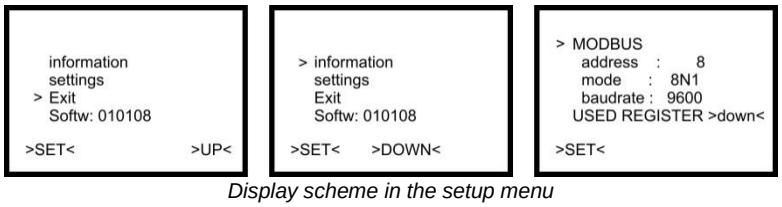
A min / max display is only available in single-line mode. The same Rreg_mw must be assigned in the Min / Max register (RWreg_57 or RWreg_58) as in the display assignment.
In the above single-line display example:

RWreg_57 = 20 (atmospheric air pressure)

An external value is displayed if the displayed measured value is a fixed specification.
In the example, the RWreg_20 was written with 1005 instead of the value [11111]. This value is then transferred to Rreg_20 and displayed.

4 Settings via Display Menu

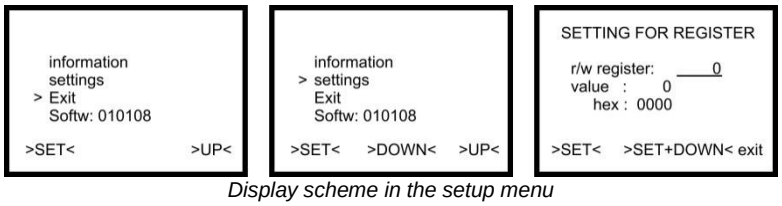
With the help of the three function buttons it is possible to get information and adjust settings for the device via the display menu (if display is available).



Display scheme in the setup menu

The current Modbus settings are visible in the "Information" menu item.
"USED REGISTER" provides information about the value registers active for this device (Rreg_mw).

Within the "setting" menu item, it is possible to select individual registers and write a value to them. This function is also protected by a password (service code).



Display scheme in the setup menu

In addition to the registers of the own system, the registers of other devices in the Modbus system can be accessed.
The function of the respective read / write register "r/w register" can be found in the table "Register assignment of the read / write registers (sensors)".

The registers (RWreg_200 to 239) are special registers for master operation. The function can be found in the point "Master settings".

Note: The requests and settings for registers are designed for the FuehlerSysteme-Modbus system and can therefore only be used to a limited extent for communication with third-party systems.

5 Special Features

5.1 Humidity/Temperature

To determine the other humidity parameters for registers Rreg_3 to Rreg_7 the basic values (humidity and temperature) from Rreg_1 and Rreg_2 are used. This means that the measuring system can also be used as a "humidity calculator" with default values in RWreg_1 and RWreg_2.

The following formulas are used:

$$dew\ point\ temperature\ [°C] = \frac{243{,}12 * (\log\left(\frac{r.h.}{100}\right) + \frac{17{,}62 * t}{243{,}12 + t})}{17{,}62 - (\log\left(\frac{r.h.}{100}\right) - \frac{17{,}62 * t}{243{,}12 + t})}$$

$r.h. = relative\ humidity\ in\ \%$
$t = temperature\ in\ °C$

The vapor pressure is required for further calculations:

$$vapor\ pressure\ [Pa] = 611{,}2 * \exp\left(\frac{17{,}62 * t_d}{243{,}12 + t_d}\right)$$

$t_d = dew\ point\ temperature\ in\ °C$

$$absolute\ humidity\left[\frac{g}{m^3}\right] = \frac{vapor\ pressure}{461{,}51\left[\frac{J}{kgK}\right] * (273.15 + t)} * 1000$$

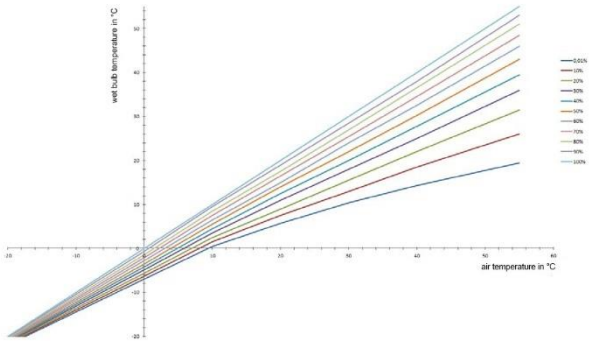
When calculating the mixing ratio, the air pressure (p) from register WRreg_7 is also included. This is 1013 mbar as standard and can be described with other values.

$$mixed\ ration\left[\frac{g}{kg}\right] = 0{,}622 * \frac{e}{p * 100 - e}$$

$e = vapor\ pressure$

$$enthalpy\left[\frac{kJ}{kg}\right] = 1{,}005\left[\frac{kJ}{kg\ k}\right] * \mathrm{t} + \text{mixed ratio}\left[\frac{kg}{kg}\right] * (2500\left[\frac{kJ}{kg}\right] + 1{,}86\left[\frac{kJ}{kg\ k}\right] * t)$$

The determination of the wet bulb temperature can be derived from the following diagram.



5.2 Humidity Heated/Temperature

A heating of the humidity sensor serves to protect against condensation and the associated longer measurement failure due to the formation of water droplets.
However, the heating results in a slightly increased inertia of the measured value.
The PT100 temperature sensor must be in the measuring medium of the humidity sensor.
Rreg_01 is the calculated relative humidity.
Rreg_02 is the temperature of the PT100.

5.3 Differential Pressure / Volume Flow

The Rreg_22 or Rreg_23 is used depending on the final measurement value (up to 500 Pa or up to 5000 Pa).

The following formulas are used to calculate the volume flow (Rreg_40 to _45):
The highest-pressure value from Rreg_22 and Rreg_23 is used as the differential pressure for the volume calculations (absolute value of the pressure value - unsigned)

Note: The differential pressure can also be specified via WRreg_22 and WRreg_23.

With RWreg_24 (1 .. 999), k-factor (k_F) = 1 .. 999

$$volume\ flow\left[\frac{m^3}{h}\right] = \sqrt{differential\ pressure\ [Pa] * \frac{2}{density_{air}}} * k_F * 1000$$

$$density_{air}\left[\frac{kg}{m^3}\right] = \frac{p_0}{Rs * T} = \frac{RWreg\_7 * 100}{287{,}058 * (273{,}15 + RWreg\_46)}$$

RWreg_7: Preset air pressure (standard = 1013 mbar)

RWreg_46: Preset air temperature (standard = 200 = 20.0°C)

With RWreg_24 (1000 .. 1999) k-factor (k_F) = 1 .. 999

$$volume\ flow\left[\frac{m^3}{h}\right] = \sqrt{diffential\ pressure[Pa]} * k_F * 1000$$

Please note that the value ranges of Rreg_40 to _45 each go up to max. 9999 go.

A zero point calibration of the differential pressure sensor can be carried out in addition to setting RWreg_26 using the >DOWN< key and display (hold down for 3 sec).

5.4 Flow Velocity / Volume Flow

An air flow (V) of 0 ... 5 (20) m / s can be measured with the flow meter (see Rreg_47).
To calculate the volume flows (Rreg_40 to _45), the cross section (A) (see RWreg_48) is included. The entry is made in [cm^2] and the following formula is used.

$$volume\ flow\left[\frac{m^3}{h}\right] = V\left[\frac{m}{s}\right] * 3600 * \frac{A[cm^2]}{10000}$$

Please note that the value ranges of Rreg_40 to _45 each up to max. 9999 go.

5.5 Brightness

When measuring brightness, the device's internal measuring ranges are automatically switched over so that measurements in the range from a few lux to 100kLux are possible.
The automatic switchover takes place via a hysteresis. There may be a short jump in measured values at the switchover point (<1 sec).
Rreg_30 and _31 are updated simultaneously according to the light intensity.

5.6 Carbon Dioxide CO_2

With the help of an optical sensor, the CO_2 content in the air is measured in ppm. Since this sensor is subject to an aging process over a longer period of time, automatic calibration is recommended. Bit_4 is set by RWreg_19 for this purpose. The CO_2 concentration is analysed over 7 days and the internal calibration data is updated. The basis for this is the statement that the general CO_2 concentration is 400ppm and that this value should be reached at least once within the 7 days. If the measured value is above 400ppm, a one-time manual calibration via Bit_0 of RWreg_19 or by pressing the >UP< button and display representation (press and hold for 3 seconds) is recommended ("fresh air" must be available). In AUTO mode, a calibration is also carried out at less than 300ppm. (earliest 10 minutes after switching on)

5.7 Mixed Gas VOC

With a sensor, the air pollution caused by volatile organic compounds (VOC) is measured. An output signal of 0 ... 100% is generated. However, this is not an absolute value for a gas, but for a gas mixture and is therefore dependent on sensation. It is thus possible to set the sensitivity (low / medium / high) via RWreg_18.
Since this sensor is subject to an aging process over a longer period of time, automatic calibration is recommended. To do this, Bit_5 is set by RWreg_19. The VOC concentration is analysed over 7 days and the internal calibration data is updated. The basis here is the statement that the general VOC concentration is 10% and that this value should be reached at least once within the 7 days. If the measured value is above 10%, a one-time manual calibration via Bit_1 of RWreg_19 or by pressing the >DOWN< button and display (press and hold for 3 seconds) is recommended ("fresh air" must be available). In AUTO mode, a calibration is also carried out at less than 5%. (earliest 10 minutes after switching on)

Note: The sensor for VOC measurement is heated. This results in a low heat influence on the board. In the case of combination devices with temperature measurement, for example, this can lead to inaccurate temperature measurements.

5.8 Oxygen O_2

With the help of a chemical sensor, the oxygen content in the air is measured in vol%. Since this sensor is subject to an aging process over a longer period of time, automatic calibration is recommended. To do this, Bit_6 is set by RWreg_19. The O_2 concentration is analysed over 7 days and the internal calibration data is updated. The basis for this is the statement that the general O_2 concentration is 20.9 vol% and that this value should be reached at least once within the 7 days. If the measured value is above 20.9 vol%, a one-time manual calibration via Bit_2 of RWreg_19 is recommended ("fresh air" must be available here)

5.9 Motion

The detection of a motion is done by detection of changes in IR radiation. These are implemented in short bursts of a few milliseconds. The delay time RWreg_32 [in seconds] defines how long a motion detection remains valid.
In combination with a light measurement, it is also possible to detect motion only below a defined light intensity (RWreg_34 and _34).

5.10 Min/Max Function

There are two separate MIN / MAX assignment registers RWreg_57 and_58 in the systems.
The numbers of the Rreg_mw at which the MIN / MAX analysis should take place are entered here.
A sliding analysis is then carried out over the specified period RWreg_59 (1, 6, 12, 24h).
The resulting min / max values can be called up in registers Rreg_55 to Rreg_58.
The values can also be shown on the display (see display representation).

5.11 Relay Function

Regardless of whether there is a relay on the system board, the relay-specific registers WRreg_50 to _55 are processed and Rreg_50 is set accordingly.
The relay function can therefore also perform a threshold value analysis for a measured value.
If a relay is available, this switches according to the content of Rreg_50. (0 = OFF 1 = ON)

6 Register Assignment for Signal Converters

6.1 FS1701 - IN: Analogue / OUT: Modbus

There are 8 analogue inputs on the analogue circuit board. On the manufacturer side, these are set as 0 ... 10V or 4 ... 20mA (also mixed possible e.g. 3 * 0 ... 10V and 7 * 4 ... 20mA => note when ordering).
If the analogue inputs are to be used as digital inputs, they must be configured using RWreg_41 to _50. A voltage potential (high) can be switched on with the DIP switch. A bridging of the input to GND (for example by a potential-free switch) is thus recognized as (LOW level).
Without the DIP switch, the input is recognised as (LOW) at <1 V and as (HIGH) at >2.5 V.
Such a digital behaviour can only be realised with voltage inputs.

6.1.1 Read Register

Rreg Nr. (Fcode: 0x04)	***Range of values***	***Assigned size***	***Input***
00	system code	0x40yy	yy: Bit_7 to Bit_0 set for current input
01	0 ... 1000; 400 ... 2000; 0/1	0 ... 10V / 4 ... 20mA / dig.	voltage / current or LOW / HIGH at the Input_1
02	0 ... 1000; 400 ... 2000; 0/1	0 ... 10V / 4 ... 20mA / dig.	voltage / current or LOW / HIGH at the Input_2
03	0 ... 1000; 400 ... 2000; 0/1	0 ... 10V / 4 ... 20mA / dig.	voltage / current or LOW / HIGH at the Input_3
04	0 ... 1000; 400 ... 2000; 0/1	0 ... 10V / 4 ... 20mA / dig.	voltage / current or LOW / HIGH at the Input_4
05	0 ... 1000; 400 ... 2000; 0/1	0 ... 10V / 4 ... 20mA / dig.	voltage / current or LOW / HIGH at the Input_5
06	0 ... 1000; 400 ... 2000; 0/1	0 ... 10V / 4 ... 20mA / dig.	voltage / current or LOW / HIGH at the Input_6
07	0 ... 1000; 400 ... 2000; 0/1	0 ... 10V / 4 ... 20mA / dig.	voltage / current or LOW / HIGH at the Input_7
08	0 ... 1000; 400 ... 2000; 0/1	0 ... 10V / 4 ... 20mA / dig.	voltage / current or LOW / HIGH at the Input_8
11	-9999 ... 9999	-9999 ... 9999	calculated measured value from the Input_1
12	-9999 ... 9999	-9999 ... 9999	calculated measured value from the Input_2
13	-9999 ... 9999	-9999 ... 9999	calculated measured value from the Input_3
14	-9999 ... 9999	-9999 ... 9999	calculated measured value from the Input_4
15	-9999 ... 9999	-9999 ... 9999	calculated measured value from the Input_5
16	-9999 ... 9999	-9999 ... 9999	calculated measured value from the Input_6
17	-9999 ... 9999	-9999 ... 9999	calculated measured value from the Input_7
18	-9999 ... 9999	-9999 ... 9999	calculated measured value from the Input_8
78	0 … 65535		device number
79	0 ... 65535	software version	e.g. 01018 for 01/01/2018
100	0 ... 0xffff	special register for measuring values	see Master-Function (also writable with Wreg_500)
101	0 ... 0xffff	special register for customer code	see Master-Function (also writable with Wreg_501)
102 to 227	0 ... 0xffff	special register ongoing (see 100,101)	see Master-Function (also writable with Wreg_502 to 627)

The values in Rreg_01 to _08 reflect the analogue input value.
The values in Rreg_11 to _18 reflect the calculated analogue measured value (based on the specified measured value ranges RWreg_21 to _38).

6.1.2 Read/Write Register

RWreg Nr. (Fcode: 0x03, 0x06)	***Range of values***	***Assigned size***	**Attribute for Input**
00	**0 ... 0x7aff [0x5300]**	**H_Byte: character [A] L_Byte: number [0]**	**customer code: for free assignment e.g. actor code A000 ***
01	0 ... 2000 [11111]	0 ... 10V / 4 ... 20mA	specify voltage / current value instead of Input_1
02	0 ... 2000 [11111]	0 ... 10V / 4 ... 20mA	specify voltage / current value instead of Input_2
03	0 ... 2000 [11111]	0 ... 10V / 4 ... 20mA	specify voltage / current value instead of Input_3
04	0 ... 2000 [11111]	0 ... 10V / 4 ... 20mA	specify voltage / current value instead of Input_4
05	0 ... 2000 [11111]	0 ... 10V / 4 ... 20mA	specify voltage / current value instead of Input_5
06	0 ... 2000 [11111]	0 ... 10V / 4 ... 20mA	specify voltage / current value instead of Input_6
07	0 ... 2000 [11111]	0 ... 10V / 4 ... 20mA	specify voltage / current value instead of Input_7
08	0 ... 2000 [11111]	0 ... 10V / 4 ... 20mA	specify voltage / current value instead of Input_8
11	0 ... 60 [10]	0 ... 60	damping / averaging Input_1 *
12	0 ... 60 [10]	0 ... 60	damping / averaging Input_2 *
13	0 ... 60 [10]	0 ... 60	damping / averaging Input_3 *
14	0 ... 60 [10]	0 ... 60	damping / averaging Input_4 *
15	0 ... 60 [10]	0 ... 60	damping / averaging Input_5 *
16	0 ... 60 [10]	0 ... 60	damping / averaging Input_6 *
17	0 ... 60 [10]	0 ... 60	damping / averaging Input_7 *
18	0 ... 60 [10]	0 ... 60	damping / averaging Input_8 *
21	-9999 ... 9999 [0]	-9999 ... 9999	zero point (zero value) for Input_1 *
22	-9999 ... 9999 [0]	-9999 ... 9999	zero point (zero value) for Input_2 *
23	-9999 ... 9999 [0]	-9999 ... 9999	zero point (zero value) for Input_3 *
24	-9999 ... 9999 [0]	-9999 ... 9999	zero point (zero value) for Input_4 *
25	-9999 ... 9999 [0]	-9999 ... 9999	zero point (zero value) for Input_5 *
26	-9999 ... 9999 [0]	-9999 ... 9999	zero point (zero value) for Input_6 *
27	-9999 ... 9999 [0]	-9999 ... 9999	zero point (zero value) for Input_7 *
28	-9999 ... 9999 [0]	-9999 ... 9999	zero point (zero value) for Input_8 *
31	-9999 ... 9999 [1000]	-9999 ... 9999	end point (end value) for Input_1 *
32	-9999 ... 9999 [1000]	-9999 ... 9999	end point (end value) for Input_2 *
33	-9999 ... 9999 [1000]	-9999 ... 9999	end point (end value) for Input_3 *
34	-9999 ... 9999 [1000]	-9999 ... 9999	end point (end value) for Input_4 *
35	-9999 ... 9999 [1000]	-9999 ... 9999	end point (end value) for Input_5 *
36	-9999 ... 9999 [1000]	-9999 ... 9999	end point (end value) for Input_6 *
37	-9999 ... 9999 [1000]	-9999 ... 9999	end point (end value) for Input_7 *
38	-9999 ... 9999 [1000]	-9999 ... 9999	end point (end value) for Input_8 *
41	0, 1, 2 [0]	0-anlog. 1-dig. , 2-dig.negated.	function of Input_1 * (with current input automatically ‚0‘)
42	0, 1, 2 [0]	like RWreg_41	function of Input_2 (with current = ‚0‘) *
43	0, 1, 2 [0]	like RWreg_41	function of Input_3 (with current = ‚0‘) *
44	0, 1, 2 [0]	like RWreg_41	function of Input_4 (with current = ‚0‘) *
45	0, 1, 2 [0]	like RWreg_41	function of Input_5 (with current = ‚0‘) *
46	0, 1, 2 [0]	like RWreg_41	function of Input_6 (with current = ‚0‘) *
47	0, 1, 2 [0]	like RWreg_41	function of Input_7 (with current = ‚0‘) *
48	0, 1, 2 [0]	like RWreg_41	function of Input_8 (with current = ‚0‘) *
49	0 ... 255 [0]	Bit_7 = Input_8 to Bit_0 = Input_1	hold function for progr. digital Inputs * Input remains switched on until the associated Rreg_01 to 08 or Rreg_11 to 18 has been called up once

50	0 ... 255 [0]	Bit_7 = Input_8 to Bit_0 = Input_1	hold function for progr. digital Inputs * Input remains switched on until the reset command on RWreg_51 to 59
51	0, 1 [0]	1 = reset	reset command for Input_1 (will connect ‚0')
52	0, 1 [0]	1 = reset	reset command for Input_2 (will connect ‚0')
53	0, 1 [0]	1 = reset	reset command for Input_3 (will connect ‚0')
54	0, 1 [0]	1 = reset	reset command for Input_4 (will connect ‚0')
55	0, 1 [0]	1 = reset	reset command for Input_5 (will connect ‚0')
56	0, 1 [0]	1 = reset	reset command for Input_6 (will connect ‚0')
57	0, 1 [0]	1 = reset	reset command for Input_7 (will connect ‚0')
58	0, 1 [0]	1 = reset	reset command for Input_8 (will connect ‚0')
59	0 ... 255 [0]	Bit_7 = Input_8 to Bit_0 = Input_1	reset command for Inputs_1 to _8 Bit then becomes ‚0' again
79	10, 20 [0]	10 20	restart restart with factory settings
200 to 327	**0 ... 0xffff**	**special register**	**see Master-Function ***

Note:

- ***** Values entered **(bold)** are also saved **(Caution: do not write continuously!)**
- [x] Presetting (factory setting)
- Zero point values ... smaller than end point values ...
- Customer code
  High_Byte: ASCII-characters A ... Z [0x41 ... 0x5a], a ... z [0x61 ... 0x7a]
  LOW_Byte: Number [0 ... 255(0xff)]

Example for a measured value calculation at Input_8:

System code 0x4080: Input_8 is a current input, the rest is voltage inputs
RWreg_28 = 0; RWreg_38 = 5000

With Input_8 = 4mA follows Rreg_18 = 0
With Input_8 = 20mA follows Rreg_18 = 5000
With Input_8 = 12mA follows Rreg_18 = 2500

6.2 FS1702 - IN: Modbus / OUT: Analogue

There are 8 analogue outputs on the analogue circuit board. On the manufacturer side, these are set as 0 ... 10V or 4 ... 20mA (also mixed possible e.g. 6 * 0 ... 10V and 2* 4 ... 20mA).
Measured values transferred to the register can thus be converted into an analogue output signal. It is also possible for the outputs to be activated with a controller function. The measured value is compared with a target value and the analogue output is readjusted.
If the analogue outputs are to have a switching character (0 / 10V or 4 / 20mA), the outputs_3, _4, _7 and _8 are to be preferred. The outputs_1, _2, _5 and _6 have a response time / inertia of approx. 2 seconds in the analogue signal.

6.2.1 Read Register

Rreg Nr. (Fcode: 0x04)	***Range of values***	***Assigned size***	***Output***
00	system code	0x30yy	yy: Bit_7 to Bit_0 set for current output
01	0 ... 1000; 400 to 2000	0 ... 10V / 4 ... 20mA	voltage / current at Output_1
02	0 ... 1000; 400 to 2000	0 ... 10V / 4 ... 20mA	voltage / current at Output_2
03	0 ... 1000; 400 to 2000	0 ... 10V / 4 ... 20mA	voltage / current at Output_3
04	0 ... 1000; 400 to 2000	0 ... 10V / 4 ... 20mA	voltage / current at Output_4
05	0 ... 1000; 400 to 2000	0 ... 10V / 4 ... 20mA	voltage / current at Output_5

06	0 ... 1000; 400 to 2000	0 ... 10V / 4 ... 20mA	voltage / current at Output_6
07	0 ... 1000; 400 to 2000	0 ... 10V / 4 ... 20mA	voltage / current at Output_7
08	0 ... 1000; 400 to 2000	0 ... 10V / 4 ... 20mA	voltage / current at Output_8
78	0 ... 65535		device number
79	0 ... 65535	software version	e.g. 01018 for 01/01/2018
100	0 ... 0xffff	special register for measuring values	see Master-Function (also writable with Wreg_500)
101	0 ... 0xffff	special register for customer code	see Master-Function (also writable with Wreg_501)
102 to 227	0 ... 0xffff	special register ongoing (see 100,101)	see Master-Function (also writable with Wreg_502 to 627)

The values in Rreg_01 to _08 reflect the analogue input value.

6.2.2 Read/Write Register

RWreg Nr. (Fcode: 0x03, 0x06)	***Range of values***	***Assigned size***	***Output***
00	**0 ... 0x7aff [0x5300]**	**H_Byte: character [A] L_Byte: number [0]**	**customer code: for free assignment e.g. actor code A000** *
01	0 ... 2000 [11111]	0 ... 10V / 4 ... 20mA	specify voltage / current at Output_1
02	0 ... 2000 [11111]	0 ... 10V / 4 ... 20mA	specify voltage / current at Output_2
03	0 ... 2000 [11111]	0 ... 10V / 4 ... 20mA	specify voltage / current at Output_3
04	0 ... 2000 [11111]	0 ... 10V / 4 ... 20mA	specify voltage / current at Output_4
05	0 ... 2000 [11111]	0 ... 10V / 4 ... 20mA	specify voltage / current at Output_5
06	0 ... 2000 [11111]	0 ... 10V / 4 ... 20mA	specify voltage / current at Output_6
07	0 ... 2000 [11111]	0 ... 10V / 4 ... 20mA	specify voltage / current at Output_7
08	0 ... 2000 [11111]	0 ... 10V / 4 ... 20mA	specify voltage / current at Output_8
10	**0 ... 0x00ff [0]**	**Bit_7 to Bit_0**	**if Bit_x = 1 is set: Channel x works as analogue controller (note RWreg_40 to _68)** *
11	-9999 ... 9999 [0]	-9999 ... 9999	measured value for Output_1
12	-9999 ... 9999 [0]	-9999 ... 9999	measured value for Output_2
13	-9999 ... 9999 [0]	-9999 ... 9999	measured value for Output_3
14	-9999 ... 9999 [0]	-9999 ... 9999	measured value for Output_4
15	-9999 ... 9999 [0]	-9999 ... 9999	measured value for Output_5
16	-9999 ... 9999 [0]	-9999 ... 9999	measured value for Output_6
17	-9999 ... 9999 [0]	-9999 ... 9999	measured value for Output_7
18	-9999 ... 9999 [0]	-9999 ... 9999	measured value for Output_8
21	-9999 ... 9999 [0]	-9999 ... 9999	zero point (zero value) for Output_1 *
22	-9999 ... 9999 [0]	-9999 ... 9999	zero point (zero value) for Output_2 *
23	-9999 ... 9999 [0]	-9999 ... 9999	zero point (zero value) for Output_3 *
24	-9999 ... 9999 [0]	-9999 ... 9999	zero point (zero value) for Output_4 *
25	-9999 ... 9999 [0]	-9999 ... 9999	zero point (zero value) for Output_5 *
26	-9999 ... 9999 [0]	-9999 ... 9999	zero point (zero value) for Output_6 *
27	-9999 ... 9999 [0]	-9999 ... 9999	zero point (zero value) for Output_7 *
28	-9999 ... 9999 [0]	-9999 ... 9999	zero point (zero value) for Output_8 *
31	-9999 ... 9999 [1000]	-9999 ... 9999	end point (end value) for Output_1 *
32	-9999 ... 9999 [1000]	-9999 ... 9999	end point (end value) for Output_2
33	-9999 ... 9999 [1000]	-9999 ... 9999	end point (end value) for Output_3
34	-9999 ... 9999 [1000]	-9999 ... 9999	end point (end value) for Output_4
35	-9999 ... 9999 [1000]	-9999 ... 9999	end point (end value) for Output_5

36	-9999 ... 9999 [1000]	-9999 ... 9999	end point (end value) for Output_6	
37	-9999 ... 9999 [1000]	-9999 ... 9999	end point (end value) for Output_7	
38	-9999 ... 9999 [1000]	-9999 ... 9999	end point (end value) for Output_7	*
40	**0 ... 0x00ff [0]**	**Bit_7 to Bit_0**	**if Bit_x = 1 is set: controller function is negated**	*****
41	-9999 ... 9999 [0]	-9999 ... 9999	target value for Output_1	*
42	-9999 ... 9999 [0]	-9999 ... 9999	target value for Output_2	*
43	-9999 ... 9999 [0]	-9999 ... 9999	target value for Output_3	*
44	-9999 ... 9999 [0]	-9999 ... 9999	target value for Output_4	*
45	-9999 ... 9999 [0]	-9999 ... 9999	target value for Output_5	*
46	-9999 ... 9999 [0]	-9999 ... 9999	target value for Output_6	*
47	-9999 ... 9999 [0]	-9999 ... 9999	target value for Output_7	*
48	-9999 ... 9999 [0]	-9999 ... 9999	target value for Output_8	*
51	1 ... 9999 [10]	1 ... 999.9	P-Parameters for controller Output_1	*
52	1 ... 9999 [10]	1 ... 999.9	P-Parameters for controller Output_2	*
53	1 ... 9999 [10]	1 ... 999.9	P-Parameters for controller Output_3	*
54	1 ... 9999 [10]	1 ... 999.9	P-Parameters for controller Output_4	*
55	1 ... 9999 [10]	1 ... 999.9	P-Parameters for controller Output_5	*
56	1 ... 9999 [10]	1 ... 999.9	P-Parameters for controller Output_6	*
57	1 ... 9999 [10]	1 ... 999.9	P-Parameters for controller Output_7	*
58	1 ... 9999 [10]	1 ... 999.9	P-Parameters for controller Output_8	*
61	0 ... 9999 [600]	0 ... 9999	I-Parameters for controller Output_1	*
62	0 ... 9999 [600]	0 ... 9999	I-Parameters for controller Output_2	*
63	0 ... 9999 [600]	0 ... 9999	I-Parameters for controller Output_3	*
64	0 ... 9999 [600]	0 ... 9999	I-Parameters for controller Output_4	*
65	0 ... 9999 [600]	0 ... 9999	I-Parameters for controller Output_5	*
66	0 ... 9999 [600]	0 ... 9999	I-Parameters for controller Output_6	*
67	0 ... 9999 [600]	0 ... 9999	I-Parameters for controller Output_7	*
68	0 ... 9999 [600]	0 ... 9999	I-Parameters for controller Output_8	*
79	10, 20 [0]	10 20	restart restart with factory settings	
90 to 99	0 ... 0xffff	special register	for free use (as a buffer with read / write function)	
200 to 327	**0 ... 0xffff**	**special register**	**see Master-Function**	*****

Note:

- ***** Values entered **(bold)** are also saved **(Caution: do not write continuously!)**
- [x] Presetting (factory setting)
- Zero point values ... smaller than end point values ...
- Customer code
  High_Byte: ASCII-characters A ... Z [0x41 ... 0x5a], a ... z [0x61 ... 0x7a]
  LOW_Byte: Number [0 ... 255(0xff)]

Example standard analogue output for Output_2:

System code 0x3002: Output_2 is a current output, the remainder are voltage outputs
RWreg_22 = 0; RWreg_32 = 1000

With RWreg_12 = 0	follows Rreg_12 = 400	and Analog_Output_2 = 4mA
With RWreg_12 = 1000	follows Rreg_12 = 2000	and Analog_Output_2 = 20mA
With RWreg_12 = 500	follows Rreg_12 = 1200	and Analog_Output_2 = 12mA

Example analogue controller for Output_8:

System code 0x3002: Output_2 is a current output, the remainder are voltage outputs (thus also Output_8)
RWreg_10 = 0x0080 (Output_8 - controller mode)

According to the function of a PI controller, the voltage at Output_8 is readjusted until the measured value (RWreg_18 with the target value RWreg_48) matches.
The following applies to the control deviation (e): e = target value - measured value (e.g. for temperature measurement in heating mode)
If the controller function is negated with RWreg_40 = 0x0080 (Output_8 - controller mode / negated) it follows for the control deviation (e): e = measured value – target value (e.g. for temperature measurement in cooling mode)

PI controller: $y(t) = P_{Param.} * \left[e(t) + \frac{1}{I_{Param.}} \int_0^t e(\tau) d\tau\right]$

The sampling time is 1sec.

Note: If the I parameter = 0, no integral component is used

=> only P controller: $y(t) = P_{Param.} * e(t)$

6.3 FS1703 - IN: Modbus / OUT: Relay

There are 8 relays on the relay board, which have different states and can also be controlled separately.
The following tables show the register functions.

6.3.1 Read Register

Rreg Nr. (Fcode: 0x04)	***Range of values***	***Assigned size***	***Relay property***
00	0x5000	0x5000	system code
01	0 / 1	1 = switched	relay_state_Rel.1
02	0 / 1	1 = switched	relay_state_Rel.2
03	0 / 1	1 = switched	relay_state_Rel.3
04	0 / 1	1 = switched	relay_state_Rel.4
05	0 / 1	1 = switched	relay_state_Rel.5
06	0 / 1	1 = switched	relay_state_Rel.6
07	0 / 1	1 = switched	relay_state_Rel.7
08	0 / 1	1 = switched	relay_state_Rel.8
09	0 ... 255	Bit_7 = 1 => switched to Bit_0 = 1 => switched	relay_state_Rel.8 to relay_state_Rel.1
78	0 ... 65535		device number
79	0 ... 65535	software version	e.g. 01018 for 01/01/2018
100	0 ... 0xffff	special register for measuring values	see Master-Function (also writable with Wreg_500)
101	0 ... 0xffff	special register for customer code	see Master-Function (also writable with Wreg_501)
102 to 227	0 ... 0xffff	special register ongoing (see 100,101)	see Master-Function (also writable with Wreg_502 to 627)

The relay status can be called up individually via Rreg_01 to _08 or bit by bit via Rreg_09.

6.3.2 Read/Write Register

RWreg Nr. (Fcode: 0x03, 0x06)	***Range of values***	***Assigned size***	***Relay property***
00	**0 ... 0x7aff [0x5300]**	**H_Byte: character [A] L_Byte: number [0]**	**customer code: for free assignment e.g. actor code A000 ***
01	0 / 1 [11111]	1 = ON switch	pretend relay status for Rel.1
02	0 / 1 [11111]	1 = ON switch	pretend relay status for Rel.2
03	0 / 1 [11111]	1 = ON switch	pretend relay status for Rel.3
04	0 / 1 [11111]	1 = ON switch	pretend relay status for Rel.4
05	0 / 1 [11111]	1 = ON switch	pretend relay status for Rel.5
06	0 / 1 [11111]	1 = ON switch	pretend relay status for Rel.6
07	0 / 1 [11111]	1 = ON switch	pretend relay status for Rel.7
08	0 / 1 [11111]	1 = ON switch	pretend relay status for Rel.8
09	0 ... 255 [11111]	Bit_7 = 1 => ON switch to Bit_0 = 1 => ON switch	specify relay status for Rel.1 to Rel. 8
11	-9999 ... 9999 [11111]	-9999 ... 9999	measured value for Rel. 1 (if != 11111 RWreg_19 Bit_0 = 0)
12	-9999 ... 9999 [11111]	-9999 ... 9999	measured value for Rel. 2 (if != 11111 RWreg_19 Bit_1 = 0)
13	-9999 ... 9999 [11111]	-9999 ... 9999	measured value for Rel. 3 (if != 11111 RWreg_19 Bit_2 = 0)
14	-9999 ... 9999 [11111]	-9999 ... 9999	measured value for Rel. 4 (if != 11111 RWreg_19 Bit_3 = 0)
15	-9999 ... 9999 [11111]	-9999 ... 9999	measured value for Rel. 5 (if != 11111 RWreg_19 Bit_4 = 0)
16	-9999 ... 9999 [11111]	-9999 ... 9999	measured value for Rel. 6 (if != 11111 RWreg_19 Bit_5 = 0)
17	-9999 ... 9999 [11111]	-9999 ... 9999	measured value for Rel. 7 (if != 11111 RWreg_19 Bit_6 = 0)
18	-9999 ... 9999 [11111]	-9999 ... 9999	measured value for Rel. 8 (if != 11111 RWreg_19 Bit_7 = 0)
19	0 ... 255 [0]	Bit_7 = 1 => ON switch to Bit_0 = 1 => ON switch	specify relay status for Rel.1 to Rel.8 if a bit is set, the respective one RWreg_11 to _18 is set to 11111
21	**-9999 ... 9999 [0]**	**-9999 ... 9999**	**switch-off threshold for measured value Rel. 1 ***
22	**-9999 ... 9999 [0]**	**-9999 ... 9999**	**switch-off threshold for measured value Rel. 2 ***
23	**-9999 ... 9999 [0]**	**-9999 ... 9999**	**switch-off threshold for measured value Rel. 3 ***
24	**-9999 ... 9999 [0]**	**-9999 ... 9999**	**switch-off threshold for measured value Rel. 4 ***
25	**-9999 ... 9999 [0]**	**-9999 ... 9999**	**switch-off threshold for measured value Rel. 5 ***
26	**-9999 ... 9999 [0]**	**-9999 ... 9999**	**switch-off threshold for measured value Rel. 6 ***
27	**-9999 ... 9999 [0]**	**-9999 ... 9999**	**switch-off threshold for measured value Rel. 7 ***
28	**-9999 ... 9999 [0]**	**-9999 ... 9999**	**switch-off threshold for measured value Rel. 8 ***
31	**-9999 ... 9999 [1]**	**-9999 ... 9999**	**switch-on threshold for measured value Rel. 1 ***
32	**-9999 ... 9999 [1]**	**-9999 ... 9999**	**switch-on threshold for measured value Rel. 2 ***
33	**-9999 ... 9999 [1]**	**-9999 ... 9999**	**switch-on threshold for measured value Rel. 3 ***
34	**-9999 ... 9999 [1]**	**-9999 ... 9999**	**switch-on threshold for measured value Rel. 4 ***
35	**-9999 ... 9999 [1]**	**-9999 ... 9999**	**switch-on threshold for measured value Rel. 5 ***
36	**-9999 ... 9999 [1]**	**-9999 ... 9999**	**switch-on threshold for measured value Rel. 6 ***
37	**-9999 ... 9999 [1]**	**-9999 ... 9999**	**switch-on threshold for measured value Rel. 7 ***
38	**-9999 ... 9999 [1]**	**-9999 ... 9999**	**switch-on threshold for measured value Rel. 8 ***
40	**0 ... 255 [0]**	**Bit_7 = Rel.8 to Bit_0 = Rel.1**	**set hold function for relay *** relay remains switched on until the reset command on RWreg_41 to 49
41	0, 1 [0]	1 = reset	reset command for Rel.1 (then becomes ‚0‘)
42	0, 1 [0]	1 = reset	reset command for Rel.2 (then becomes ‚0‘)
43	0, 1 [0]	1 = reset	reset command for Rel.3 (then becomes ‚0‘)
44	0, 1 [0]	1 = reset	reset command for Rel.4 (then becomes ‚0‘)

45	0, 1 [0]	1 = reset	reset command for Rel.5 (then becomes ‚0')	
46	0, 1 [0]	1 = reset	reset command for Rel.6 (then becomes ‚0')	
47	0, 1 [0]	1 = reset	reset command for Rel.7 (then becomes ‚0')	
48	0, 1 [0]	1 = reset	reset command for Rel.8 (then becomes ‚0')	
49	0 ... 255 [0]	Bit_7 = Rel.8 to Bit_0 = Rel:1	reset command for Rel.1 to Rel.8 Bit will then go back to ‚0'	
51	**0 ... 1800 [0]**	**0 ... 1800 sec.**	**switch-on delay for Rel. 1**	*
52	**0 ... 1800 [0]**	**0 ... 1800 sec.**	**switch-on delay for Rel. 2**	*
53	**0 ... 1800 [0]**	**0 ... 1800 sec.**	**switch-on delay for Rel. 3**	*
54	**0 ... 1800 [0]**	**0 ... 1800 sec.**	**switch-on delay for Rel. 4**	*
55	**0 ... 1800 [0]**	**0 ... 1800 sec.**	**switch-on delay for Rel. 5**	*
56	**0 ... 1800 [0]**	**0 ... 1800 sec.**	**switch-on delay for Rel. 6**	*
57	**0 ... 1800 [0]**	**0 ... 1800 sec.**	**switch-on delay for Rel. 7**	*
58	**0 ... 1800 [0]**	**0 ... 1800 sec.**	**switch-on delay for Rel. 8**	*
61	**0 ... 1800 [0]**	**0 ... 1800 sec.**	**follow-up time for Rel. 1**	*
62	**0 ... 1800 [0]**	**0 ... 1800 sec.**	**follow-up time for Rel. 2**	*
63	**0 ... 1800 [0]**	**0 ... 1800 sec.**	**follow-up time for Rel. 3**	*
64	**0 ... 1800 [0]**	**0 ... 1800 sec.**	**follow-up time for Rel. 4**	*
65	**0 ... 1800 [0]**	**0 ... 1800 sec.**	**follow-up time for Rel. 5**	*
66	**0 ... 1800 [0]**	**0 ... 1800 sec.**	**follow-up time for Rel. 6**	*
67	**0 ... 1800 [0]**	**0 ... 1800 sec.**	**follow-up time for Rel. 7**	*
68	**0 ... 1800 [0]**	**0 ... 1800 sec.**	**follow-up time for Rel. 8**	*
69	**0 ... 255 [0]**	**Bit_7 = Rel.8 to Bit_0 = Rel.1**	**negating the relay state (with Bit_x = 1)**	*
71	**0 ... 60 [0]**	**0 ... 60 sec.**	**cycle time for Rel. 1 ('0' no cycle)**	*
72	**0 ... 60 [0]**	**0 ... 60 sec.**	**cycle time for Rel. 2 ('0' no cycle)**	*
73	**0 ... 60 [0]**	**0 ... 60 sec.**	**cycle time for Rel. 3 ('0' no cycle)**	*
74	**0 ... 60 [0]**	**0 ... 60 sec.**	**cycle time for Rel. 4 ('0' no cycle)**	*
75	**0 ... 60 [0]**	**0 ... 60 sec.**	**cycle time for Rel. 5 ('0' no cycle)**	*
76	**0 ... 60 [0]**	**0 ... 60 sec.**	**cycle time for Rel. 6 ('0' no cycle)**	*
77	**0 ... 60 [0]**	**0 ... 60 sec.**	**cycle time for Rel. 7 ('0' no cycle)**	*
78	**0 ... 60 [0]**	**0 ... 60 sec.**	**cycle time for Rel. 8 ('0' no cycle)**	*
79	10, 20 [0]	10 20	restart restart with factory settings	
90 to 99	0 .. 0xffff	special register	for free use (as a buffer with read / write function)	
0 to 327	**0 ... 0xffff**	**special register**	**see Master-Function**	*

Note:

- ***** Values entered **(bold)** are also saved **(Caution: do not write continuously!)**
- [x] Presetting (factory setting)
- Zero point values ... <u>smaller</u> than end point values ...
- Customer code
  High_Byte: ASCII-characters A ... Z [0x41 ... 0x5a], a ... z [0x61 ... 0x7a]
  LOW_Byte: Number [0 ... 255(0xff)]

The status of RWreg_01 to _09 are taken over directly into registers Rreg_01 to 09 and the relay ON / OFF switch is set.
When used from RWreg_11, the relays work with the inclusion of switching thresholds of programmed ON / OFF times, negations and cycle times. Only then are Rreg_01 to _09 set and the relays switched accordingly.

7 FS1704 – Servicedisplay

The universal Servicedisplay FS1704 is an independent measuring system from the FuehlerSysteme-Modbus series. It can thus determine independently provided measured variables as well as communicate with other systems via Modbus.

Other special features:

- Larger graphic display
- Display with white or RGB backlight
- Display representation freely programmable within limits with three display pages
- automatic changes to displays or states via programmable actions
- implemented piezo buzzer (e.g. for acoustic alarms)
- WLAN for access to the internal registers (e.g. for programming or transferring register values (measured values))
- Can be used as "master" in the Modbus system with extended command memory (as with the FuehlerSysteme signal converter)
- small mathematical links with master register

7.1 Register Assignment

The register assignments have an equivalent structure to the general sensor systems, see chapter 3.

Some registers are not used or have a specific property.

Changes compared to the "Read register" table from Chapter *Fehler! Verweisquelle konnte nicht gefunden werden..Fehler! Verweisquelle konnte nicht gefunden werden.*

Rreg Nr. (Fcode 0x04)	***Range of values***	***Assigned size and partial unit***	***Measured value property***
33	0 ... 1	0 / 1	is used by actions (R.33 0-1)
34	0 ... 1	0 / 1	is used by actions (R.34 0-1)
50	0 ... 1	0 / 1	internal piezo buzzer OFF / ON
59	0 ... 2359	hour and minute	internal time with hour / minute (may have a difference of 30 seconds)
78	0... 0xffff		device number
80	0 ...0xffff	error code	error coding (bits equivalent to system code)
100 to 227	0 ... 0xffff	special register ongoing (see 100, 101)	see Master-Function (also writable with Wreg_500 to 627)
228		Special register for measured value (CLOCK)	NOT WRITABLE or READABLE!

Changes compared to the "Read / Write register" table from Chapter 3.2.

RWreg Nr. (Fcode: 0x03, 0x06)	***Range of values***	***Assigned size and partial unit***	***Measured value property***
50	0 ... 1 [11111]	11111 ‚0' OFF / ‚1' ON	piezo control according to register program. (RWreg_51) piezo buzzer: Fixed OFF / ON
51	**0 ... 255 [0]**	**Rreg: 0 ... 255**	**assign piezo buzzer to an Rreg_x (zero => none)** ***** only measured value register (Rreg_mw) Bit_15 is set on deletion (Off) by pressing the Enter key
52	**-9999 ... 9999 [0]**	**value**	**OFF switching value for piezo buzzer** *****
53	**-9999 ... 9999 [0]**	**value**	**ON switching value for piezo buzzer** *****
60	**0 / 1 [0]**	**0 / 1**	**display positive/negative** *****
61	**100 ... 180 [135]**	**100 ... 180**	**display contrast** *****
62	**0 ... 0xffff [0]**		**display backlight** ***** **Bit_0 ... Bit_3: Brightness blue or white** **Bit_4 ... Bit_7: Brightness green** **Bit_8 ... Bit_11: Brightness red**

63	0, 1 ... 3		current page (0 = update)	
64	**1 ... 3**		**number of selectable pages (Pages)**	*
65	**10.. 60 [0]**	**0, 10 ... 60 sec**	**rotation time of display pages 1 to 3**	*
66	**0 ... 255 [0]**	**Rreg: 0 ... 255**	**assign DIAGRAM_1 to a Rreg_x**	*
67	**-9999 ... 9999 [0]**	**value**	**DIAGRAM_1 lower value**	*
68	**-9999 ... 9999 [0]**	**value**	**DIAGRAM_1 upper value**	*
69	**0 ... 0xffff [0]**		**DIAGRAM_1 Bit_0 to _7 (+/- auto scaling value)** **Bit_8 = 1 (auto scaling)** **Bit_9 = 1 (Slide points connected with line)** **Bit_10 = 1 (Filled under slide point)**	*
70	**0 ... 99 [0]**	**0..99 h**	**DIAGRAM_1 interval time 0...99 hours**	*
71	**0 ... 255 [0]**	**Rreg: 0 ... 255**	**assign DIAGRAM_2 to a Rreg_x**	*
72	**-9999 ... 9999 [0]**	**value**	**DIAGRAM_2 lower value**	*
73	**-9999 ... 9999 [0]**	**value**	**DIAGRAM_2 upper value**	*
74	**0 ... 0xffff [0]**		**DIAGRAM_2 Bit_0 to _7 (+/- auto scaling value)** **Bit_8 = 1 (Auto scaling)** **Bit_9 = 1 (Slide points connected with line)** **Bit_10 = 1 (Filled under slide point)**	*
75	**0 ... 99 [0]**	**0..99 h**	**DIAGRAMM_2 interval time 0...99 hours**	*
76	**0/1 [0]**	**0 / 1**	**WLAN inactive / active** Bit_15: WLAN connected Bit_14: client connected	*
78	**0 ... 0xffff [0xc201]**	**Modbus [38400, 8N1, Adr:1]**	**Bit_0 ... Bit_7: Modbus address 0 ... 255** **Bit_9: two stop bits** **Bit_10/Bit_11: [0, 0] 8E1; [1, 0] 8N1; [0, 1] 8O1** **Bit_15/Bit_16: [0, 0] 2400; [1, 0] 9600; [0, 1] 19200; [1, 1] 38400**	*
79	0 ... 0xffff [0]	10 20	restart restart with factory settings	
80	**0 ... 9999 [0911]**	**0 ... 9999**	**servicecode (password) for settings via display**	*
90 to 99	0 ... 0xffff	special register	for free use (as a buffer with read / write function)	
200 to 327	**0 ... 0xffff**	**special register**	see Master-Function	*
400 to 463	**0 ... 0xffff**	**special register**	**see Master-Function/Mathematics**	*
500 to 627	0 ... 0xffff	**special register**	WRITE REGISTER ONLY: see Rreg_100..227	
400 to 463	**0 ... 0xffff**	**special register**	**see Master-Function/Mathematics**	*
700 to 849	**0 ... 0xffff**	**special register**	**action register, see chapter 7.1.2**	*
1000 ... 1719	**0 ... 0xef**	**special register**	**display register (ONLY 8 Bit), see chapter** □	*
1720	1 ... 31	day	WRITE REGISTER ONLY, time [DAY]	
1721	1... 12	month	WRITE REGISTER ONLY, time [MONTH]	
1722	0 ... 99	year	WRITE REGISTER ONLY, time [YEAR]	
1723	0 ... 23	year	WRITE REGISTER ONLY, time [HOUR]	
1724	0 ... 59	year	WRITE REGISTER ONLY, time [MINUTE]	

Note:

- ***** Values entered **(bold)** are also saved
  (Caution: do not write continuously!)
- [x] Value after switching on or with presetting (factory setting)
- [v] Value with default setting (factory setting) - depending on the device type
- OFF switching value RWreg_52 smaller than ON switching value RWreg_53
- Customer code
  High_Byte: ASCII-characters A ... Z [0x41 ... 0x5a], a ... z [0x61 ... 0x7a]
  LOW_Byte: Number [0 ... 255(0xff)]

7.1.1 Specific Servicedisplay-Register FS1704

Other register assignments are specific to the FS1704 and are used to program the representations on the display. These (write / read) registers are graphically set accordingly via the SETUP settings (display). Access via the Modbus system or WLAN is also possible.

Caution: When writing to these registers, they are only checked for validity to a limited extent. Incorrect entries can therefore lead to malfunctions in the display.
A value between 0xf0 and 0xff may not be entered in any display register (8 bits).

The display representations are - freely programmable - within limits. The following are possible:

- 3 display pages
- 20 texts with 10 characters each (11 different types of writing)
- 20 values with assigned measured value registers (15 different written forms, decimal point position, as well as date and time display)
- 20 units with 21 different parameters (12 different fonts)
- 20 symbols with 25 different predefined types and sizes
- 20 graphics with 30 different geometric shapes
- for all elements determine the position on the display
- Set the brightness of the pixels and background level

Table of registers for the design of the texts (8 bits each)

Displ_Reg	TEXT_1	Displ_Reg	TEXT_2	Displ_Reg	TEXT_3	Displ_Reg	TEXT_4	Displ_Reg	TEXT_5
1000	x_coor.	**1014**	x_coor.	**1028**	x_coor.	**1042**	x_coor.	**1056**	x_coor.
1001	y_coor.	**1015**	y_coor.	**1029**	y_coor.	**1043**	y_coor.	**1057**	y_coor.
1002	Bright/BL	**1016**	Bright/BL	**1030**	Bright/BL	**1044**	Bright/BL	**1058**	Bright/BL
1003	Page/Font	**1017**	Page/Font	**1031**	Page/Font	**1045**	Page/Font	**1059**	Page/Font
1004	Char_1	**1018**	Char_1	**1032**	Char_1	**1046**	Char_1	**1060**	Char_1
1005	Char_2	**1019**	Char_2	**1033**	Char_2	**1047**	Char_2	**1061**	Char_2
1006	Char_3	**1020**	Char_3	**1034**	Char_3	**1048**	Char_3	**1062**	Char_3
1007	Char_4	**1021**	Char_4	**1035**	Char_4	**1049**	Char_4	**1063**	Char_4
1008	Char_5	**1022**	Char_5	**1036**	Char_5	**1050**	Char_5	**1064**	Char_5
1009	Char_6	**1023**	Char_6	**1037**	Char_6	**1051**	Char_6	**1065**	Char_6
1010	Char_7	**1024**	Char_7	**1038**	Char_7	**1052**	Char_7	**1066**	Char_7
1011	Char_8	**1025**	Char_8	**1039**	Char_8	**1053**	Char_8	**1067**	Char_8
1012	Char_9	**1026**	Char_9	**1040**	Char_9	**1054**	Char_9	**1068**	Char_9
1013	Char_10	**1027**	Char_10	**1041**	Char_10	**1055**	Char_10	**1069**	Char_10
Displ_Reg	**TEXT_6**	**Displ_Reg**	**TEXT_7**	**Displ_Reg**	**TEXT_8**	**Displ_Reg**	**TEXT_9**	**Displ_Reg**	**TEXT_10**
1070	x_coor.	**1084**	x_coor.	**1098**	x_coor.	**1112**	x_coor.	**1126**	x_coor.
1071	y_coor.	**1085**	y_coor.	**1099**	y_coor.	**1113**	y_coor.	**1127**	y_coor.
1072	Bright/BL	**1086**	Bright/BL	**1100**	Bright/BL	**1114**	Bright/BL	**1128**	Bright/BL
1073	Page/Font	**1087**	Page/Font	**1101**	Page/Font	**1115**	Page/Font	**1129**	Page/Font
1074	Char_1	**1088**	Char_1	**1102**	Char_1	**1116**	Char_1	**1130**	Char_1
1075	Char_2	**1089**	Char_2	**1103**	Char_2	**1117**	Char_2	**1131**	Char_2
1076	Char_3	**1090**	Char_3	**1104**	Char_3	**1118**	Char_3	**1132**	Char_3
1077	Char_4	**1091**	Char_4	**1105**	Char_4	**1119**	Char_4	**1133**	Char_4
1078	Char_5	**1092**	Char_5	**1106**	Char_5	**1120**	Char_5	**1134**	Char_5
1079	Char_6	**1093**	Char_6	**1107**	Char_6	**1121**	Char_6	**1135**	Char_6
1080	Char_7	**1094**	Char_7	**1108**	Char_7	**1122**	Char_7	**1136**	Char_7
1081	Char_8	**1095**	Char_8	**1109**	Char_8	**1123**	Char_8	**1137**	Char_8
1082	Char_9	**1096**	Char_9	**1110**	Char_9	**1124**	Char_9	**1138**	Char_9
1083	Char_10	**1097**	Char_10	**1111**	Char_10	**1125**	Char_10	**1139**	Char_10
Displ_Reg	**TEXT_11**	**Displ_Reg**	**TEXT_12**	**Displ_Reg**	**TEXT_13**	**Displ_Reg**	**TEXT_14**	**Displ_Reg**	**TEXT_15**
1140	x_coor.	**1154**	x_coor.	**1168**	x_coor.	**1182**	x_coor.	**1196**	x_coor.
1141	y_coor.	**1155**	y_coor.	**1169**	y_coor.	**1183**	y_coor.	**1197**	y_coor.
1142	Bright/BL	**1156**	Bright/BL	**1170**	Bright/BL	**1184**	Bright/BL	**1198**	Bright/BL
1143	Page/Font	**1157**	Page/Font	**1171**	Page/Font	**1185**	Page/Font	**1199**	Page/Font
1144	Char_1	**1158**	Char_1	**1172**	Char_1	**1186**	Char_1	**1200**	Char_1
1145	Char_2	**1159**	Char_2	**1173**	Char_2	**1187**	Char_2	**1201**	Char_2
1146	Char_3	**1160**	Char_3	**1174**	Char_3	**1188**	Char_3	**1202**	Char_3
1147	Char_4	**1161**	Char_4	**1175**	Char_4	**1189**	Char_4	**1203**	Char_4
1148	Char_5	**1162**	Char_5	**1176**	Char_5	**1190**	Char_5	**1204**	Char_5
1149	Char_6	**1163**	Char_6	**1177**	Char_6	**1191**	Char_6	**1205**	Char_6
1150	Char_7	**1164**	Char_7	**1178**	Char_7	**1192**	Char_7	**1206**	Char_7
1151	Char_8	**1165**	Char_8	**1179**	Char_8	**1193**	Char_8	**1207**	Char_8
1152	Char_9	**1166**	Char_9	**1180**	Char_9	**1194**	Char_9	**1208**	Char_9
1153	Char_10	**1167**	Char_10	**1181**	Char_10	**1195**	Char_10	**1209**	Char_10
Displ_Reg	**TEXT_16**	**Displ_Reg**	**TEXT_17**	**Displ_Reg**	**TEXT_18**	**Displ_Reg**	**TEXT_19**	**Displ_Reg**	**TEXT_20**
1210	x_coor.	**1224**	x_coor.	**1238**	x_coor.	**1252**	x_coor.	**1266**	x_coor.
1211	y_coor.	**1225**	y_coor.	**1239**	y_coor.	**1253**	y_coor.	**1267**	y_coor.
1212	Bright/BL	**1226**	Bright/BL	**1240**	Bright/BL	**1254**	Bright/BL	**1268**	Bright/BL
1213	Page/Font	**1227**	Page/Font	**1241**	Page/Font	**1255**	Page/Font	**1269**	Page/Font
1214	Char_1	**1228**	Char_1	**1242**	Char_1	**1256**	Char_1	**1270**	Char_1
1215	Char_2	**1229**	Char_2	**1243**	Char_2	**1257**	Char_2	**1271**	Char_2
1216	Char_3	**1230**	Char_3	**1244**	Char_3	**1258**	Char_3	**1272**	Char_3
1217	Char_4	**1231**	Char_4	**1245**	Char_4	**1259**	Char_4	**1273**	Char_4
1218	Char_5	**1232**	Char_5	**1246**	Char_5	**1260**	Char_5	**1274**	Char_5
1219	Char_6	**1233**	Char_6	**1247**	Char_6	**1261**	Char_6	**1275**	Char_6
1220	Char_7	**1234**	Char_7	**1248**	Char_7	**1262**	Char_7	**1276**	Char_7
1221	Char_8	**1235**	Char_8	**1249**	Char_8	**1263**	Char_8	**1277**	Char_8
1222	Char_9	**1236**	Char_9	**1250**	Char_9	**1264**	Char_9	**1278**	Char_9
1223	Char_10	**1237**	Char_10	**1251**	Char_10	**1265**	Char_10	**1279**	Char_10

Explanation:

- Displ_Reg save write / read register *(Caution: do not write continuously!)*
- x_coor. horizontal pixel coordinate (0...239)
- y_coor. vertical pixel coordinate (0...127)
- Bright/BL brightness and background level
  (Bit_7..._5 – Bright 0...7; Bit_2..._0 – BL 0...7)
  Level 0=transparent
- Page/Font display page (Bit_7 = Page 3; Bit_6 = Page 2; Bit_5 = Page 1)
  font (Bit_3..._0) [0...11]
- Char_x represented character ASCII-Code
  (0...9 A…Z a…z Ä Ö Ü ä ö ü - . : empty)
  Note: Special characters [ä = code 05; ö = code 06; ü = code 07]

Fonts used for texts:

0. standard
1. standard double
2. standard double narrow
3. standard double / italic
4. standard triple
5. standard triple narrow
6. standard triple / italic
7. standard triple narrow / italic
8. Arial double
9. Arial double / italic
10. Arial triple
11. Arial triple / italic

Table of registers for designing the values (8 bits each)

Displ_Reg	VALUE_1	Displ_Reg	VALUE_2	Displ_Reg	VALUE_3	Displ_Reg	VALUE_4	Displ_Reg	VALUE_5
1280	x_coor.	**1286**	x_coor.	**1292**	x_coor.	**1298**	x_coor.	**1304**	x_coor.
1281	y_coor.	**1287**	y_coor.	**1293**	y_coor.	**1299**	y_coor.	**1305**	y_coor.
1282	Bright/BL	**1288**	Bright/BL	**1294**	Bright/BL	**1300**	Bright/BL	**1306**	Bright/BL
1283	Page/Font	**1289**	Page/Font	**1295**	Page/Font	**1301**	Page/Font	**1307**	Page/Font
1284	Comma	**1290**	Comma	**1296**	Comma	**1302**	Comma	**1308**	Comma
1285	Value Reg.	**1291**	Value Reg.	**1297**	Value Reg.	**1303**	Value Reg.	**1309**	Value Reg.
Displ_Reg	**VALUE_6**	**Displ_Reg**	**VALUE_7**	**Displ_Reg**	**VALUE_8**	**Displ_Reg**	**VALUE_9**	**Displ_Reg**	**VALUE_10**
1310	x_coor.	**1316**	x_coor.	**1322**	x_coor.	**1328**	x_coor.	**1334**	x_coor.
1311	y_coor.	**1317**	y_coor.	**1323**	y_coor.	**1329**	y_coor.	**1335**	y_coor.
1312	Bright/BL	**1318**	Bright/BL	**1324**	Bright/BL	**1330**	Bright/BL	**1336**	Bright/BL
1313	Page/Font	**1319**	Page/Font	**1325**	Page/Font	**1331**	Page/Font	**1337**	Page/Font
1314	Comma	**1320**	Comma	**1326**	Comma	**1332**	Comma	**1338**	Comma
1315	Value Reg.	**1321**	Value Reg.	**1327**	Value Reg.	**1333**	Value Reg.	**1339**	Value Reg.
Displ_Reg	**VALUE_11**	**Displ_Reg**	**VALUE_12**	**Displ_Reg**	**VALUE_13**	**Displ_Reg**	**VALUE_14**	**Displ_Reg**	**VALUE_15**
1340	x_coor.	**1346**	x_coor.	**1352**	x_coor.	**1358**	x_coor.	**1364**	x_coor.
1341	y_coor.	**1347**	y_coor.	**1353**	y_coor.	**1359**	y_coor.	**1365**	y_coor.
1342	Bright/BL	**1348**	Bright/BL	**1354**	Bright/BL	**1360**	Bright/BL	**1366**	Bright/BL
1343	Page/Font	**1349**	Page/Font	**1355**	Page/Font	**1361**	Page/Font	**1367**	Page/Font
1344	Comma	**1350**	Comma	**1356**	Comma	**1362**	Comma	**1368**	Comma
1345	Value Reg.	**1351**	Value Reg.	**1357**	Value Reg.	**1363**	Value Reg.	**1369**	Value Reg.
Displ_Reg	**VALUE_16**	**Displ_Reg**	**VALUE_17**	**Displ_Reg**	**VALUE_18**	**Displ_Reg**	**VALUE_19**	**Displ_Reg**	**VALUE_20**
1370	x_coor.	**1376**	x_coor.	**1382**	x_coor.	**1388**	x_coor.	**1394**	x_coor.
1371	y_coor.	**1377**	y_coor.	**1383**	y_coor.	**1389**	y_coor.	**1395**	y_coor.
1372	Bright/BL	**1378**	Bright/BL	**1384**	Bright/BL	**1390**	Bright/BL	**1396**	Bright/BL
1373	Page/Font	**1379**	Page/Font	**1385**	Page/Font	**1391**	Page/Font	**1397**	Page/Font
1374	Comma	**1380**	Comma	**1386**	Comma	**1392**	Comma	**1398**	Comma
1375	Value Reg.	**1381**	Value Reg.	**1387**	Value Reg.	**1393**	Value Reg.	**1399**	Value Reg.

Explanation:

- Displ_Reg save write / read register *(Caution: do not write continuously!)*
- x_coor. horizontal pixel coordinate (0...239)
- y_coor. vertical pixel coordinate (0...127)
- Bright/BL brightness and background level
  (Bit_7..._5 – Bright 0...7; Bit_2..._0 – BL 0...7)
  Level 0=transparent; *Not useful when displaying values!*
- Page/Font display page (Bit_7 = Page 3; Bit_6 = Page 2; Bit_5 = Page 1)
  font (Bit_3..._0) [0...11]
- Comma value is shown with a decimal point (comma) (0, 1st or 2nd decimal place)
- Value Reg. used value register (Rreg_1 ..._ 227)
  When using the value registers 102, 104 ... 227, the
  customer code in the form e.g. R012 shown instead of a measured value.
  (The maximum written form is Nr. 11)
  Special case: value register = 228 (the display is made with DATE / TIME.)

Comma = 0: hh:mm:ss
Comma = 1: DD:MM:20YY
Comma = 2: hh:mm
Comma = 3: DD:MM:20YY hh:mm
Comma = 4: DD:MM:20YY hh:mm:ss
The date and time are set by writing to the register.
Wreg_**1720** (Day), **_1721** (Month), **_1722** (Year), **_1723** (Hour), **_1724** (Minute)

Note: The device-internal timer is not a real-time clock. It must be set and, if necessary, corrected at intervals.

Fonts used for values:

0. standard
1. standard double
2. standard double narrow
3. standard double / italic
4. standard triple
5. standard triple narrow
6. standard triple / italic
7. standard triple narrow / italic
8. Arial double
9. Arial double / italic
10. Arial triple
11. Arial triple / italic
12. Arial triple bold
13. Arial triple bold / italic
14. Arial fourfold bold
15. Arial fourfold bold / italic

Table of registers for designing the units (8 bits each)

Displ_Reg	UNIT_1	Displ_Reg	UNIT_2	Displ_Reg	UNIT_3	Displ_Reg	UNIT_4	Displ_Reg	UNIT_5
1400	x_coor.	**1405**	x_coor.	**1410**	x_coor.	**1415**	x_coor.	**1420**	x_coor.
1401	y_coor.	**1406**	y_coor.	**1411**	y_coor.	**1416**	y_coor.	**1421**	y_coor.
1402	Bright/BL	**1407**	Bright/BL	**1412**	Bright/BL	**1417**	Bright/BL	**1422**	Bright/BL
1403	Page/Font	**1408**	Page/Font	**1413**	Page/Font	**1418**	Page/Font	**1423**	Page/Font
1404	Unit	**1409**	Unit	**1414**	Unit	**1419**	Unit	**1424**	Unit
Displ_Reg	**UNIT_6**	**Displ_Reg**	**UNIT_7**	**Displ_Reg**	**UNIT_8**	**Displ_Reg**	**UNIT_9**	**Displ_Reg**	**UNIT_10**
1425	x_coor.	**1430**	x_coor.	**1435**	x_coor.	**1440**	x_coor.	**1445**	x_coor.
1426	y_coor.	**1431**	y_coor.	**1436**	y_coor.	**1441**	y_coor.	**1446**	y_coor.
1427	Bright/BL	**1432**	Bright/BL	**1437**	Bright/BL	**1442**	Bright/BL	**1447**	Bright/BL
1428	Page/Font	**1433**	Page/Font	**1438**	Page/Font	**1443**	Page/Font	**1448**	Page/Font
1429	Unit	**1434**	Unit	**1439**	Unit	**1444**	Unit	**1449**	Unit
Displ_Reg	**UNIT_11**	**Displ_Reg**	**UNIT_12**	**Displ_Reg**	**UNIT_13**	**Displ_Reg**	**UNIT_14**	**Displ_Reg**	**UNIT_15**
1450	x_coor.	**1455**	x_coor.	**1460**	x_coor.	**1465**	x_coor.	**1470**	x_coor.
1451	y_coor.	**1456**	y_coor.	**1461**	y_coor.	**1466**	y_coor.	**1471**	y_coor.
1452	Bright/BL	**1457**	Bright/BL	**1462**	Bright/BL	**1467**	Bright/BL	**1472**	Bright/BL
1453	Page/Font	**1458**	Page/Font	**1463**	Page/Font	**1468**	Page/Font	**1473**	Page/Font
1454	Unit	**1459**	Unit	**1464**	Unit	**1469**	Unit	**1474**	Unit
Displ_Reg	**UNIT_16**	**Displ_Reg**	**UNIT_17**	**Displ_Reg**	**UNIT_18**	**Displ_Reg**	**UNIT_19**	**Displ_Reg**	**UNIT_20**
1475	x_coor.	**1480**	x_coor.	**1485**	x_coor.	**1490**	x_coor.	**1495**	x_coor.
1476	y_coor.	**1481**	y_coor.	**1486**	y_coor.	**1491**	y_coor.	**1496**	y_coor.
1477	Bright/BL	**1482**	Bright/BL	**1487**	Bright/BL	**1492**	Bright/BL	**1497**	Bright/BL
1478	Page/Font	**1483**	Page/Font	**1488**	Page/Font	**1493**	Page/Font	**1498**	Page/Font
1479	Unit	**1484**	Unit	**1489**	Unit	**1494**	Unit	**1499**	Unit

Explanation:

- Displ_Reg save write / read register *(Caution: do not write continuously!)*
- x_coor. horizontal pixel coordinate (0...239)
- y_coor. vertical pixel coordinate (0...127)
- Bright/BL brightness and background level
  (Bit_7..._5 – Bright 0...7; Bit_2..._0 – BL 0...7)
  Level 0=transparent
- Page/Font display page (Bit_7 = Page 3; Bit_6 = Page 2; Bit_5 = Page 1)
  font (Bit_3..._0) [0...11]
- Unit predefined different units [0...21]
  %, °C, $^{g}/_{m3}$, $^{g}/_{kg}$, ppm, mbar, bar, Pa, Lux, kLux, $m^3/_h$, $m^3/_{min}$, $m^3/_s$, $^l/_h$, $^l/_{min}$, $^l/_s$, $^m/_s$, V, mA, vol%, $\mu g/m^3$, $^{kJ}/_{kg}$,

Fonts used for Unit:

0. standard
1. standard double
2. standard double narrow
3. standard double / italic
4. standard triple
5. standard triple narrow
6. standard triple / italic
7. standard triple narrow / italic
8. Arial triple
9. Arial triple / italic
10. Arial triple
11. Arial triple / italic
12. Arial triple light italic

Table of registers for the design of the symbols (8 bits each)

Displ_Reg	SYMBOL_1	Displ_Reg	SYMBOL_2	Displ_Reg	SYMBOL_3	Displ_Reg	SYMBOL_4	Displ_Reg	SYMBOL_5
1500	x_coor.	**1505**	x_coor.	**1510**	x_coor.	**1515**	x_coor.	**1520**	x_coor.
1501	y_coor.	**1506**	y_coor.	**1511**	y_coor.	**1516**	y_coor.	**1521**	y_coor.
1502	Bright/BL	**1507**	Bright/BL	**1512**	Bright/BL	**1517**	Bright/BL	**1522**	Bright/BL
1503	Page/Size	**1508**	Page/Size	**1513**	Page/Size	**1518**	Page/Size	**1523**	Page/Size
1504	SYMBOL	**1509**	SYMBOL	**1514**	SYMBOL	**1519**	SYMBOL	**1524**	SYMBOL
Displ_Reg	**SYMBOL_6**	**Displ_Reg**	**SYMBOL_7**	**Displ_Reg**	**SYMBOL_8**	**Displ_Reg**	**SYMBOL_9**	**Displ_Reg**	**SYMBOL_10**
1525	x_coor.	**1530**	x_coor.	**1535**	x_coor.	**1540**	x_coor.	**1545**	x_coor.
1526	y_coor.	**1531**	y_coor.	**1536**	y_coor.	**1541**	y_coor.	**1546**	y_coor.
1527	Bright/BL	**1532**	Bright/BL	**1537**	Bright/BL	**1542**	Bright/BL	**1547**	Bright/BL
1528	Page/Size	**1533**	Page/Size	**1538**	Page/Size	**1543**	Page/Size	**1548**	Page/Size
1529	SYMBOL	**1534**	SYMBOL	**1539**	SYMBOL	**1544**	SYMBOL	**1549**	SYMBOL
Displ_Reg	**SYMBOL_11**	**Displ_Reg**	**SYMBOL_12**	**Displ_Reg**	**SYMBOL_13**	**Displ_Reg**	**SYMBOL_14**	**Displ_Reg**	**SYMBOL_15**
1550	x_coor.	**1555**	x_coor.	**1560**	x_coor.	**1565**	x_coor.	**1570**	x_coor.
1551	y_coor.	**1556**	y_coor.	**1561**	y_coor.	**1566**	y_coor.	**1571**	y_coor.
1552	Bright/BL	**1557**	Bright/BL	**1562**	Bright/BL	**1567**	Bright/BL	**1572**	Bright/BL
1553	Page/Size	**1558**	Page/Size	**1563**	Page/Size	**1568**	Page/Size	**1573**	Page/Size
1554	SYMBOL	**1559**	SYMBOL	**1564**	SYMBOL	**1569**	SYMBOL	**1574**	SYMBOL
Displ_Reg	**SYMBOL_16**	**Displ_Reg**	**SYMBOL_17**	**Displ_Reg**	**SYMBOL_18**	**Displ_Reg**	**SYMBOL_19**	**Displ_Reg**	**SYMBOL_20**
1575	x_coor.	**1580**	x_coor.	**1585**	x_coor.	**1590**	x_coor.	**1595**	x_coor.
1576	y_coor.	**1581**	y_coor.	**1586**	y_coor.	**1591**	y_coor.	**1596**	y_coor.
1577	Bright/BL	**1582**	Bright/BL	**1587**	Bright/BL	**1592**	Bright/BL	**1597**	Bright/BL
1578	Page/Size	**1583**	Page/Size	**1588**	Page/Size	**1593**	Page/Size	**1598**	Page/Size
1579	SYMBOL	**1584**	SYMBOL	**1589**	SYMBOL	**1594**	SYMBOL	**1599**	SYMBOL

Explanation:

- Displ_Reg save write / read register *(Caution: do not write continuously!)*
- x_coor. horizontal pixel coordinate (0...239)
- y_coor. vertical pixel coordinate (0...127)
- Bright/BL brightness and background level
  (Bit_7..._5 – Bright 0...7; Bit_2..._0 – BL 0...7)
  Level 0=transparent
- Page/Size display page (Bit_7 = Page 3; Bit_6 = Page 2; Bit_5 = Page 1)
  size (Bit_1 [triple]; Bit_0 [double])
- SYMBOL predefined different symbols [0...25]

0. man is walking
1. man is standing
2. smiley sad
3. smiley medium
4. smiley laugh
5. switch on
6. switch off
7. door closed
8. door open
9. window open
10. window closed
11. caution sign
12. horn
13. lightbulb
14. sun
15. thermometer
16. drops of moisture
17. cloud CO_2
18. cloud CO
19. house VOC
20. cloud air pressure
21. smog
22. u-tube
23. fan
24. heating
25. cooling

Table of registers for the design of the graphics (8 bits each)

Displ_Reg	GRAPHIC_1	Displ_Reg	GRAPHIC_2	Displ_Reg	GRAPHIC_3	Displ_Reg	GRAPHIC_4	Displ_Reg	GRAPHIC_5
1600	x_coor.B.	**1606**	x_coor.B.	**1612**	x_coor.B.	**1618**	x_coor.B.	**1624**	x_coor.B.
1601	y_coor.B.	**1607**	y_coor.B.	**1613**	y_coor.B.	**1619**	y_coor.B.	**1625**	y_coor.B.
1602	x_coor.E.	**1608**	x_coor.E.	**1614**	x_coor.E.	**1620**	x_coor.E.	**1626**	x_coor.E.
1603	y_coor.E.	**1609**	y_coor.E.	**1615**	y_coor.E.	**1621**	y_coor.E.	**1627**	y_coor.E.
1604	Bright/ART	**1610**	Bright/ART	**1616**	Bright/ART	**1622**	Bright/ART	**1628**	Bright/ART
1605	Page/ART	**1611**	Page/ART	**1617**	Page/ART	**1623**	Page/ART	**1629**	Page/ART
Displ_Reg	**GRAPHIC_6**	**Displ_Reg**	**GRAPHIC_7**	**Displ_Reg**	**GRAPHIC_8**	**Displ_Reg**	**GRAPHIC_9**	**Displ_Reg**	**GRAPHIC_10**
1630	x_coor.B.	**1636**	x_coor.B.	**1642**	x_coor.B.	**1648**	x_coor.B.	**1654**	x_coor.B.
1631	y_coor.B.	**1637**	y_coor.B.	**1643**	y_coor.B.	**1649**	y_coor.B.	**1655**	y_coor.B.
1632	x_coor.E.	**1638**	x_coor.E.	**1644**	x_coor.E.	**1650**	x_coor.E.	**1656**	x_coor.E.
1633	y_coor.E.	**1639**	y_coor.E.	**1645**	y_coor.E.	**1651**	y_coor.E.	**1657**	y_coor.E.
1634	Bright/ART	**1640**	Bright/ART	**1646**	Bright/ART	**1652**	Bright/ART	**1658**	Bright/ART
1635	Page/ART	**1641**	Page/ART	**1647**	Page/ART	**1653**	Page/ART	**1659**	Page/ART
Displ_Reg	**GRAPHIC_11**	**Displ_Reg**	**GRAPHIC_12**	**Displ_Reg**	**GRAPHIC_13**	**Displ_Reg**	**GRAPHIC_14**	**Displ_Reg**	**GRAPHIC_15**
1660	x_coor.B.	**1666**	x_coor.B.	**1672**	x_coor.B.	**1678**	x_coor.B.	**1684**	x_coor.B.
1661	y_coor.B.	**1667**	y_coor.B.	**1673**	y_coor.B.	**1679**	y_coor.B.	**1685**	y_coor.B.
1662	x_coor.E.	**1668**	x_coor.E.	**1674**	x_coor.E.	**1680**	x_coor.E.	**1686**	x_coor.E.
1663	y_coor.E.	**1669**	y_coor.E.	**1675**	y_coor.E.	**1681**	y_coor.E.	**1687**	y_coor.E.
1664	Bright/ART	**1670**	Bright/ART	**1676**	Bright/ART	**1682**	Bright/ART	**1688**	Bright/ART
1665	Page/ART	**1671**	Page/ART	**1677**	Page/ART	**1683**	Page/ART	**1689**	Page/ART
Displ_Reg	**GRAPHIC_16**	**Displ_Reg**	**GRAPHIC_17**	**Displ_Reg**	**GRAPHIC_18**	**Displ_Reg**	**GRAPHIC_19**	**Displ_Reg**	**GRAPHIC_20**
1690	x_coor.B.	**1696**	x_coor.B.	**1702**	x_coor.B.	**1708**	x_coor.B.	**1714**	x_coor.B.
1691	y_coor.B.	**1697**	y_coor.B.	**1703**	y_coor.B.	**1709**	y_coor.B.	**1715**	y_coor.B.
1692	x_coor.E.	**1698**	x_coor.E.	**1704**	x_coor.E.	**1710**	x_coor.E.	**1716**	x_coor.E.
1693	y_coor.E.	**1699**	y_coor.E.	**1705**	y_coor.E.	**1711**	y_coor.E.	**1717**	y_coor.E.
1694	Bright/ART	**1700**	Bright/ART	**1706**	Bright/ART	**1712**	Bright/ART	**1718**	Bright/ART
1695	Page/ART	**1701**	Page/ART	**1707**	Page/ART	**1713**	Page/ART	**1719**	Page/ART

Explanation:

- Displ_Reg save write / read register *(Caution: do not write continuously!)*
- x_coor.B. horizontal pixel coordinate (0...239)– Beginning (top left)
- y_coor.B. vertical pixel coordinate (0...127) – Beginning
- x_coor.E. horizontal pixel coordinate (0...239) – End (bottom right)
- y_coor.E. vertical pixel coordinate (0...127) – End
- Bright/ART brightness (Bit_7..._5 – Bright 0...7);
  ART (Bit_0) => GraphicART [>=16]
- Page/ART Display page (Bit_7 = Page 3; Bit_6 = Page 2; Bit_5 = Page 1)
  GraphicART (Bit_3..._0) [0...15 bzw. 16...30]

Predefined graphics cards:

0. line
1. double line
2. triple line
3. double dashed line
4. rectangle
5. double rectangle
6. triple rectangle
7. filled rectangle
8. triangle with point left or right
9. triangle with point left or right double
10. triangle with point left or right triple
11. triangle filled with a point on the left or right
12. triangle with point above or below
13. triangle with point above or double below
14. triangle with point above or three times below
15. triangle filled with point above or below
16. circle
17. double circle
18. triple circle
19. filled circle
20. table with 2 * 2 boxes
21. table with 2 * 3 boxes
22. table with 2 * 4 boxes
23. table with 3 * 2 boxes
24. table with 3 * 3 boxes
25. table with3 * 4 boxes
26. rectangle with cross
27. diagram 1 (only with Graphic _19)
28. double diagram 1 (only with Graphic_19)
29. diagram 2 (only with Graphic _20)
30. double diagram 2 (only with Graphic _20)
    NOTE: When using diagrams, RWreg_66 ... 74 must be included!

7.1.2 Special Actions

Various actions are defined via the contents of these action registers.
50 different actions can be assigned. An effect is assigned to an action, which is activated depending on the measured values in a measured value register. See also chapter 7.3.5

Table of registers for the actions (16 bits each)

Act._Reg	Action	Parameter	Act._Reg	Action	Parameter	Act._Reg	Action	Parameter	Act._Reg	Action	Parameter	Act._Reg	Action	Parameter
700	1	Effect/Reg	730	11	Effect/Reg	760	21	Effect/Reg	790	31	Effect/Reg	820	41	Effect/Reg
701	1	Value_off	731	11	Value_off	761	21	Value_off	791	31	Value_off	821	41	Value_off
702	1	Value_on	732	11	Value_on	762	21	Value_on	792	31	Value_on	822	41	Value_on
703	2	Effect/Reg	733	12	Effect/Reg	763	22	Effect/Reg	793	32	Effect/Reg	823	42	Effect/Reg
704	2	Value_off	734	12	Value_off	764	22	Value_off	794	32	Value_off	824	42	Value_off
705	2	Value_on	735	12	Value_on	765	22	Value_on	795	32	Value_on	825	42	Value_on
706	3	Effect/Reg	736	13	Effect/Reg	766	23	Effect/Reg	796	33	Effect/Reg	826	43	Effect/Reg
707	3	Value_off	737	13	Value_off	767	23	Value_off	797	33	Value_off	827	43	Value_off
708	3	Value_on	738	13	Value_on	768	23	Value_on	798	33	Value_on	828	43	Value_on
709	4	Effect/Reg	739	14	Effect/Reg	769	24	Effect/Reg	799	34	Effect/Reg	829	44	Effect/Reg
710	4	Value_off	740	14	Value_off	770	24	Value_off	800	34	Value_off	830	44	Value_off
711	4	Value_on	741	14	Value_on	771	24	Value_on	801	34	Value_on	831	44	Value_on
712	5	Effect/Reg	742	15	Effect/Reg	772	25	Effect/Reg	802	35	Effect/Reg	832	45	Effect/Reg
713	5	Value_off	743	15	Value_off	773	25	Value_off	803	35	Value_off	833	45	Value_off
714	5	Value_on	744	15	Value_on	774	25	Value_on	804	35	Value_on	834	45	Value_on
715	6	Effect/Reg	745	16	Effect/Reg	775	26	Effect/Reg	805	36	Effect/Reg	835	46	Effect/Reg
716	6	Value_off	746	16	Value_off	776	26	Value_off	806	36	Value_off	836	46	Value_off
717	6	Value_on	747	16	Value_on	777	26	Value_on	807	36	Value_on	837	46	Value_on
718	7	Effect/Reg	748	17	Effect/Reg	778	27	Effect/Reg	808	37	Effect/Reg	838	47	Effect/Reg
719	7	Value_off	749	17	Value_off	779	27	Value_off	809	37	Value_off	839	47	Value_off
720	7	Value_on	750	17	Value_on	780	27	Value_on	810	37	Value_on	840	47	Value_on
721	8	Effect/Reg	751	18	Effect/Reg	781	28	Effect/Reg	811	38	Effect/Reg	841	48	Effect/Reg
722	8	Value_off	752	18	Value_off	782	28	Value_off	812	38	Value_off	842	48	Value_off
723	8	Value_on	753	18	Value_on	783	28	Value_on	813	38	Value_on	843	48	Value_on
724	9	Effect/Reg	754	19	Effect/Reg	784	29	Effect/Reg	814	39	Effect/Reg	844	49	Effect/Reg
725	9	Value_off	755	19	Value_off	785	29	Value_off	815	39	Value_off	845	49	Value_off
726	9	Value_on	756	19	Value_on	786	29	Value_on	816	39	Value_on	846	49	Value_on
727	10	Effect/Reg	757	20	Effect/Reg	787	30	Effect/Reg	817	40	Effect/Reg	847	50	Effect/Reg
728	10	Value_off	758	20	Value_off	788	30	Value_off	818	40	Value_off	848	50	Value_off
729	10	Value_on	759	20	Value_on	789	30	Value_on	819	40	Value_on	849	50	Value_on

Explanation:

- Displ_Reg save write / read register *(Caution: do not write continuously!)*
  Action 50 actions can be programmed
  Effect/Reg Bit_15 ... _9 (number coded) for effect 0 ... 109
  Bit_8 ..._ 0 (number coded) for register Rreg 0 ... 227
  Value_off Bit_13..._0 (number coded) for Value_off 0 ... 9999
  Bit_15 (= 1) – Value_off is negative
  Bit_14 (= 1) action is effective if >= Value_off;
  (= 0) action is effective if < Value_off
  Value _on Bit_13..._0 (number coded) for Value_on 0 ... 9999
  Bit_15 (= 1) – Value_on is negative
  Bit_14 (= 1) action is effective if < Value_on;
  (= 0) action is effective if >= Value_on

Effects used:

0. none
1. page 1 [will be displayed]
2. page 2 [will be displayed]
3. page 3 [will be displayed]
4. display BG RED [at RGB]
5. display BG GREEN [at RGB]
6. display BG BLUE [at RGB] / WHITE
7. piezo buzzer
8. R.33 0-1 [Rreg_33 0/1 set]
9. R.34 0-1 [Rreg_34 0/1 set]

[fade ON / OFF respectively)
10. text_1 to
29. text_20
30. Value_1 to
49. Value _20
50. Unit_1 to
69. Unit_20
70. Symbol_1 to
89. Symbol_20
90. Graphic_1 to
109. Graphic_20

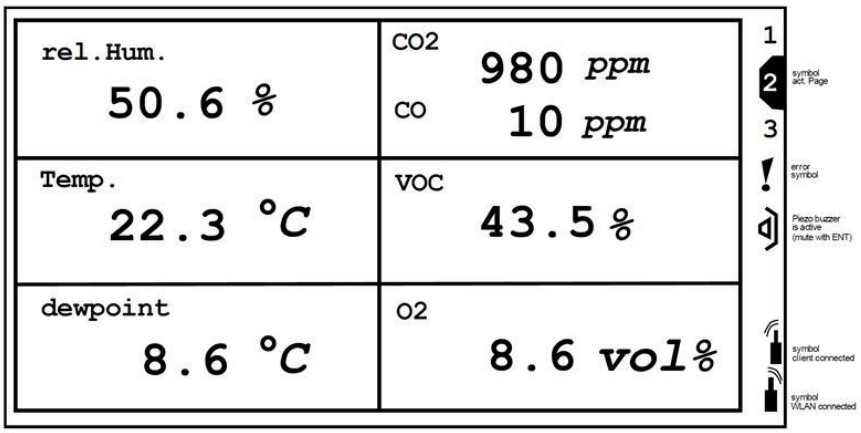
Example:
The CO sensor value (Rreg_12) is to be monitored. At > = 50 ppm the piezo buzzer should sound and switch off again below 20 ppm.

RWreg_700 (Effect = 7 and Rreg = 12) => 0x0e0c
RWreg_701 (OFF condition <20) => 0x0014
RWreg_702 (ON condition> = 50) => 0x0032

7.2 Operation

If the measuring system is in general measuring mode, no further operation is necessary. There is a rotary and SET switch on the right side. Many settings can be made with this. Turning it switches between the three display pages (which can also be time-controlled see RWreg_65). The internal buttons are also required for some settings. To do this, the housing must be opened. In addition to the largely freely configurable display graphics, there are clearly defined symbols on the right edge.

This area should not be used when designing the display graphics!
The picture shows an example in the general measurement / display mode:



ROTARY SWITCH: Selection of the page (Page)
SET button: Switch to the setup menu

7.3 Setup

The following settings can be made in the setup. These are then described in more detail below.

- MODBUS-parameter
- REGISTER read-write
- MEASUREMENT-parameter
- DISPLAY-screen setup
- DISPLAY-screen activation
- DISPLAY-backlight and contrast
- Delete pages and factory settings
- WLAN settings
- PASSWORD: ----
- DATE and TIME setting (only with programmed representation on a display page)

A password must be entered in order to be able to make settings within the setup menu. The correct password is valid for one hour or until it is switched on again.
If no setting is made within the setup, the system switches back to general measurement / display mode after 3 minutes.

Note: There may be time delays on the Modbus within the setup settings (malfunction)!

7.3.1 MODBUS-Parameter

The following Modbus parameters are set within this menu item:

- Baud rate: 2400, 9600, 19200, 38400
- Mode: 8N1 (8 data bits, no parity, 1 stop bit)
- Mode: 8E1 or 2 (8 data bits, even parity, 1 or 2 stop bits)
- Mode: 8O1 or 2 (8 data bits, odd parity, 1 or 2 stop bits)
- MB-Address: 0... 255

The individual parameters can be set with the ROTARY and SET switches.
The parameter is saved with the SET button at the same time.

7.3.2 REGISTER Read/Write

In this menu item it is possible to query and also to change the various registers within the measuring system as well as connected MB systems.
You choose the point:

- this system
- ext. system with ADR: xxx

The read or read / write register is then selected.
If it is a read register (read only), it is not possible to change the value.
With a read / write register (read and write) the value can also be changed.

Note: Changes can only be made using the internal cursor keyboard!
Caution: A change in register values is not checked for plausibility. Incorrect entries can lead to malfunctions in the respective measuring system!

7.3.3 MEASUREMENT-Parameter

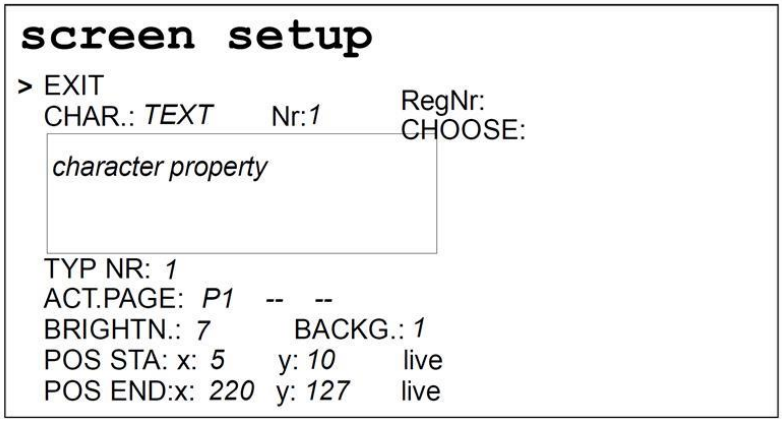
The universal measuring and display system has various internal sensors (according to the factory configuration).
In this setup menu, some important sensor-typical parameters for the measurement can be set directly (e.g. offset values, calibrations, zero point setting).
All values can be activated and changed using the rotary switch and SET switch. Some parameters are also saved with the change!

7.3.4 DISPLAY-Screen Setup

This sub-program enables the user to create 3 own display pages. In doing so, RWreg_1000 ... 1719 are changed.
In total, the following can be placed on the 3 display pages:

- 20 texts with 10 characters each (11 different types of writing)
- 20 values with assigned measured value registers (15 different written forms and decimal point positions as well as date and time display)
- 20 units with 21 different parameters (12 different written forms)
- 20 symbols with 25 different predefined types and sizes
- 20 graphics with 30 different geometric shapes
- Set the brightness of the pixels and background level

The setup menu has the following structure.



In general, the individual points can be reached and set with the rotary and SET switch. The cursor and a grey background indicate which parameter is currently active. To be on the safe side, some parameters can only be changed using the internal keyboard! This is indicated by the key symbols. (for a further description see also chapter □)

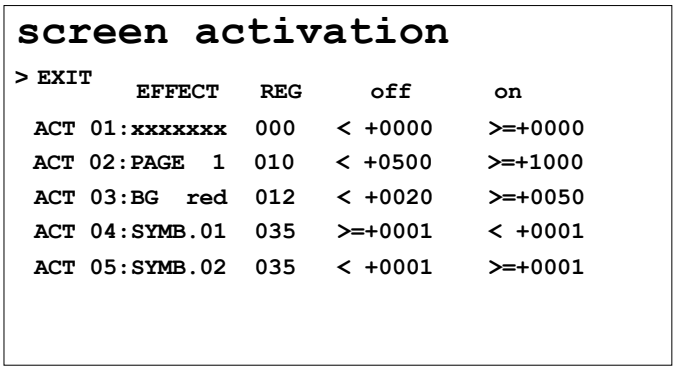
[CHAR.]	is selected: TEXT, VALUE (value), UNIT (unit), SYMBOL, GRAPHIC
[NR]	relates to the selected [CHAR.] and can therefore be 1 ... 20.
[RegNR]	becomes active with VALUE. The measured value register is specified here 0 ... 227 (228) whose value is to be displayed (e.g. Rreg_10 corresponds to the CO_2 value).
[CHOOSE]	becomes active at UNIT and is used to select the unit to be displayed.
[TYP NR]	different fonts are selected for TEXT, VALUE and UNIT. In addition, the decimal point for the numerical value is specified for VALUE. With SYMBOL different symbols and sizes (simple, * 2, * 3) are defined. With GRAPHIC there are various geometrical predefined graphics to choose from.
[ACT.PAGE]	The selected [CHAR.] Is assigned to a display page (P1, P2 and / or P3). *Note:* If several pages are selected, the same parameters (POS, brightness, background) apply to all pages.
[BRIGHTN.]	Pixel brightness for the selected [CHAR.]
[BACKG.]	Pixel brightness of the background for the selected [CHAR.] (0 = transparent)
[POS STA]	Start position for the [CHAR.] (Upper left corner)
[POS END]	End position (only with [CHAR.] = GRAPHIC) (lower right corner)
[live]	Positioning takes place directly on the display page with the internal cursor keys. SET takes you to the next activated display page [ACT.PAGE] or back again. *Note:* With live positioning, the display pages are always newly refreshed. This can lead to "flickering" of the display, especially with large-area graphics

A preview of the activated [CHAR.] is shown within the [Character property] area. If [CHAR.] = TEXT is selected, the property field can be selected and changed with SET. Using the cursor keys it is then possible to change the TEXT (10 characters).
The following characters are available:

- 0 ... 9, A ... Z, Ä, Ö, Ü, a ... z, ä, ö, ü, minus, period, narrow space, space

7.3.5 DISPLAY-Screen Actions

This universal measuring system enables actions to be carried out independently, which are programmed accordingly and continuously updated.
50 different actions can be programmed.
An action is assigned to an effect and a measured value register and is carried out when a value therein is exceeded or not reached.
The individual menu items are accessed using the rotary and SET switches.
In the parameters EFFECT, REG, off and on changes can only be made with the cursor keys.
The underlined parameters are selected for setting. If EFFECT is underlined, the line can be deleted by pressing the LEFT button for 2 seconds!



[ACT xx]	Action number 01 to 50 (scroll with rotary switch or UP / DOWN button)
[EFFECT]	The following parameters are possible for the effect specification (PAGE 1, PAGE 2, PAGE 3, BG red, BG gre., BG blue (whi), PIEZO, R.33 0-1, R.34 0-1, TEXT 01 to 20, VALUE 01 to 20, UNIT 01 to 20, SYMB. 01 to 20, GRAPH 01 to 20)
[REG]	the associated measured value register 0 .. 227
[off]	Condition for switching off the effect (< or >= the REG value)
[on]	Condition for switching on the effect (>= or < the REG value)
	Caution: The values are **not** checked for plausibility.

Note: The effects BG red, green, blue can only work with an RGB display. BG whi applies to a W display.

Example - picture above

ACT 01:	not assigned
ACT 02:	Display page 2 is shown as “fixed” when the CO_2-value (Rreg_10) exceeds 1000. If the value falls below this, the previous display page is shown again (in the example: 1).
ACT 03:	The display background lighting (with RGB) is only switched to RED when the CO value of 50 (Rreg_12) is exceeded (e.g. as an optical alarm function). Undo again at 20!
ACT 04:	Symbol 01 (in the example: "running man") is shown when a motion (Rreg_35 = 1) is detected.
ACT 05:	Symbol 02 (in the example: "standing man") is shown when there is no motion (Rreg_35 = 0).

Note: If the symbols 01 and 02 are in the same position, the display changes according to the motion detection.

Special case

If the register Rreg_59 (hour / minute) is used for an action, the corresponding action is only carried out in the event of "equality".

Example:

Effect = Page 3; off >= 1900; on >= 0700
Page 3 is permanently switched on at 7:00 a.m. and switched back at 7:00 p.m.

7.3.6 DISPLAY-Backlight and Contrast

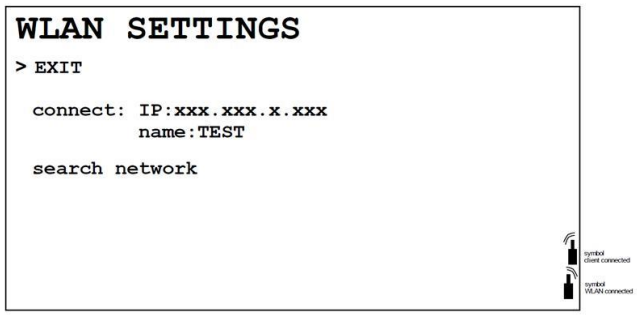
Display settings are made in this menu item.
The background lighting white or RGB can be set in 15 steps.
Furthermore, a contrast setting and an inverted display are possible.
All points can be set using the rotary and SET switches.

7.3.7 Delete Pages and Factory Settings

At this point there is the option of deleting the content of display pages or the actions or also resetting the entire system to the factory settings.
All points can be set using the rotary and SET switch. However, the "DOWN" key must be pressed to acknowledge the delete function (open the housing).
A general restart of the system is also possible.

7.3.8 WLAN settings

This menu item is used to set up a WLAN connection.



The point [connect] is switched ON or OFF [no connect].
If it is switched OFF, this state is retained even after a system restart. The system then does not attempt to log in.
With the point [search network] and acknowledgment with the SET button, the WLAN networks in the area are searched (max. 10). It is then possible to activate one of these networks and the password will be requested.

Note: The characters that can be used within the password are limited (max. 20 characters: 0...9; a...z; A..Z; -; +; . ; :; ! ; (;)) [No space]. This must be taken into account in the WLAN password assignment! The password is entered using the cursor keys in the device.

Once a connection has been established, it is always automatically re-established (even after a restart). The symbol "WLAN connected" indicates this. If there is a connection to a client, this can be recognised by the second symbol "client connected".

It is possible to access the registers through the WLAN connection. This is not done by a Modbus protocol, but by a simplified data transfer in ASCII format.

Note: A checksum or a value check does not take place!
The WLAN connection is not designed for "fast" data transfer. Internal Modbus processing has priority.

7.3.9 Examples

7.3.9.1 Quering a Read Register

Send (space as separation)
l_reg_a(0x0d)
- l = code
- reg = Register [Rreg]
- a = number (max. 50)

Answer
Aaaa,Lrrrr,wwwww(0x0d) (if number> 1 then continuously)
- Aaaa = identifier [A] and address [aaa] (3 digits)
- Lrrrr = identifier [L] and Rreg [rrrr] (4 digits)
- wwwww = register value 5 places (16 bit)

7.3.9.2 Quering a Read/Write register

Send (space as separation)
r_reg_a(0x0d)
- L = code
- reg = Register [RWreg]
- a = number (max. 50)

Answer
Aaaa,Lrrrr,wwwww(0x0d) (if number> 1 then continuously)
- Aaaa = identifier [A] and address [aaa] (3 digits)
- Rrrrr = identifier [R] and RWreg [rrrr] (4 digits)
- wwwww = register value 5 places (16 bit)

7.3.9.3 Writing a Read/Write Register

Send (space as separation)
w_reg_a_w(0x0d)
- L = code
- reg = Register [RWreg]
- a = number (max. 10)
- w = values (with number> 1 consecutive with space separation)

Answer
Aaaa,Lrrrr,wwwww(0x0d) (if number> 1 then continuously)
- Aaaa = identifier [A] and address [aaa] (3 digits)
- Rrrrr = identifier [R] and RWreg [rrrr] (4 digits)
- wwwww = register value 5 places (16 bit)

7.3.10 DATE and TIME setting

If there is a date and / or time display on a display page, this menu item can be used to set the day, month, year, hours and minutes.

Note: The system does not have an internal real-time clock. This means that the setting is lost after a restart. There may also be deviations in seconds over time. To keep this low, it is possible to enter a second correction per day.

This menu item is not password-protected and can therefore always be called up.

8 Master-Operating / Master Settings

The systems within the FuehlerSysteme-Modbus series offer the possibility of working as an independent "Master".
In general, all systems (sensors as well as actuators) work in "Slave" mode.
A connected master thus calls a slave via the address and processes its data.
For small Modbus networks with the FuehlerSysteme-Modbus series, a "sensor" or "actuator" can be declared as a master. After switching the DIP switch of (Switch A) to address 255, the master mode is switched on.
The device now independently processes commands (if they exist) which are coded on RWreg_200 to _239 (actuator: to _327).
Attention: only one device may have the address 255.

Coding of the master register:

Note: The device must be in Slave mode for coding the master register!

RWreg_200: 0xyyzz
- yy stands for an address (0 ... 254 or 255) [254 or 255 is your own device],
- zz stands for the register (RWreg_x or Rreg_x)
  from which a value is read (value and customer code)
  cached in Rreg_100 and Rreg_101

RWreg_201: 0xyyzz
- yy stands for an address (0 ... 255) [255 is your own device],
- zz stands for the register (RWreg_x)
  to which the cached value (from Rreg_100) is written
  refers is zz> = 100 is again written to an Rreg> = 100 and also the
  customer code passed on and entered in the following register

RWreg_202, _204, _206 ... _238 (actuator: ..._ 326) (see RWreg_200)
RWreg_203, _205, _207 ... _239 (actuator: ..._327) (see RWreg_201)
Rreg_100, _102, _104 ... _128 (actuator: ..._226) (see Rreg_100) - temporarily stored measured values of the device
Rreg_101, _103, _105 ... _129 (actuator: ..._227) (see Rreg_101) - cached customer code of the device

Note: Used address 00 and reg. 00 have no functions here.

Example: **The humidity / temperature measuring system has address 01**
as well as
actuator analogue outputs which have the address 255 (configured as Master)
The following Master registers are set in the actuator
RWreg_200: 0x0101 (read from Adr_01, Reg_01 - humidity value)
RWreg_201: 0xff0b (write the humidity value on own device in RWreg_11)
RWreg_202: 0x0102 (read from Adr_01, Reg_02 - temperature value)
RWreg_201: 0xff0c (write the temperature value on own device in RWreg_12)

According to the set parameters in the actuator (RWreg_21, _31, _22, _32) the analogue output is now automatically updated.

Note: If the device detects a single character 0xfe (254) on the bus with address 255 (master) (during a processing pause), the master operation is suspended for 2 minutes. A 0xfd (253) starts the master mode again (or automatically after 2 minutes).

SPECIAL CASE

only device-related data transfer

If the paired registers (RWreg_200 / 201; _202 / 203;…) each contain the address 254/255 (as a code for the own device), the command read / write is always carried out on the corresponding

register. The DIP switch addressing to 255 is not required for this. This makes it possible to constantly move register contents in your own device.
Example: a device continuously shows three measured values in the display (e.g. CO_2, VOC, TEMP)

- The relay function has the reference to the CO_2 value (RWreg_51 = 10) and set switching thresholds (RWreg_52, _53)
- Display priority on CO_2 (RWreg_75 = 10)
- Master register (RWreg_200 = 0xff32 => Add.255; Reg.50) - read
- Master register - write (RWreg_201 = 0xff3f => Add.255; Reg.63) - write

When the CO_2 threshold is exceeded, the relay register Rreg_50 is set.
This value (0/1) is transferred to the priority register RWreg_63 for the display using the master register setting. The CO_2 value (PRIO) is now shown individually in the display as the "alarm value".

With a display device incl. keys, the command sequence can also be entered in the master registers RWreg_200 to _239 / _327.

Display scheme in the setup menu

Note: RWreg_90 to _99 can be used as a customer-specific buffer, for example to automatically create a chain of measured values with the appropriate sequence in order to read them as a MODBUS block of possibly up to 10 values at the same time.

Example for setting the master register:
Measuring system for HUM / TEMP / CO_2 / VOC
RWreg_200: 0xff01 (read from Rreg_01 - humidity value)
RWreg_201: 0xff5a (write the humidity value to RWreg_90)
RWreg_202: 0xff02 (read from Rreg_02 - temperature value)
RWreg_203: 0xff5b (write the temperature value to RWreg_91)
RWreg_204: 0xfe08 (read from RWreg_08 - humidity offset value)
RWreg_205: 0xff5c (write the humidity offset value to RWreg_92)
RWreg_206: 0xfe09 (read from RWreg_09 - temperature offset value)
RWreg_207: 0xff5d (write the temperature offset value to RWreg_93)
RWreg_208: 0xff0a (read from Rreg_10 - CO_2 value)
RWreg_209: 0xff5e (write the CO_2 value to RWreg_94)
RWreg_210: 0xff0b (read from Rreg_11 - VOC value)
RWreg_211: 0xff5f (write the VOC value to RWreg_95)
RWreg_212: 0xff13 (read from Rreg_19 - calibration mode)
RWreg_213: 0xff60 (write the calibration mode to RWreg_96)

RWreg_90 to _96 can now be read simultaneously with a MODBUS command.

ATTENTION when using the master register:
The system does not check the entries of addresses and registers for plausibility and logical links. In the case of undefined assignments, malfunctions can occur that affect the measuring system or the entire BUS chain and its systems!

9 Additional Master Register for Mathematical Calculations

In addition to the chapter already described, 8 additional mathematical links can be implemented with the FS1701, FS1702, FS1703, FS1704 and FS1600.

If values are fetched from another or a separate device via specifications in the master registers 200, 202 etc., this value can still be linked with the math registers 400, 401 etc.

A math register (16 bits) has the following structure, consisting of the operant (number) and the operation property. With this function the fetched value is offset. The result is then the “new value”.

Bit_0 ... Bit_7 Basic number (number) for the mathematical calculations (number = 0 ... 255)

Bit_8 ... Bit_11 Base number expansion (number)

B_11	B_10	B_9	B_8	
0	0	0	0	not used
0	0	0	1	number = number * 10 (=> 0…2550)
0	0	1	0	not used
0	0	1	1	number = number * 100 (=> 0...25500)
0	1	0	0	not used
0	1	0	1	number = number / 10 (=> 0...25,5)
0	1	1	0	not used
0	1	1	1	number = number / 100 (=> 0...2,55)
1	x	x	x	number = - number (Change of sign)

Bit_12 .. Bit_15 (Arithmetic operation Bit_15 = 0)

B_15	B_14	B_13	B_12	
0	0	0	0	new value = fetched value + number
0	0	0	1	new value = fetched value * number
0	0	1	0	new value = fetched value / number **note:** if number = 0 => new value = 32767
0	0	1	1	new value = fetched value MOD number **note:** if number = 0 => new value= 32767
0	1	0	0	new value = number / fetched value **note:** if fetched value = 0 => new value = 32767
0	1	0	1	new value = number – fetched value
0	1	1	0	not used
0	1	1	1	moving averaging average = average – (average + fetched value) / number new value = average **attention:** number >= 1
	at Bit_11 = 1			if fetched value = 0 => average immediately 0 (without moving averaging)

Bit_12 .. Bit_15 (Comparison operation Bit_15 = 1)

B_15	B_14	B_13	B_12		
1	0	0	0	if (fetched value = number)	then new value = 0
				if (fetched value <> number)	then new value = no changes
1	0	0	1	if (fetched value < number)	then new value = 0
				if (fetched value >= number)	then new value = no changes
1	0	1	0	if (fetched value > number)	then new value = 0
				if (fetched value <= number)	then new value = no changes
1	0	1	1	new value = fetched value & number (bit by bit AND operation)	
1	1	0	0	if (fetched value = number)	then new value = 1
				if (fetched value <> number)	then new value = 0
1	1	0	1	if (fetched value < number)	then new value = 1
				if (fetched value >= number)	then new value = 0
1	1	1	0	if (fetched value < number)	then new value = 1
				if (fetched value >= number)	then new value = 0
1	1	1	1	new value = fetched number | number (bit by bit OR operation)	

The new value is limited to a value range from -32767 to 32767 (whole number) and then stored in the corresponding Master Rreg 100, 102 etc.

Bit_15 to Bit_9 = 1 (0xFExx) Processing pause in Master Mode in xx = number (1/10 sec)
Bit_15 to Bit_9 = 1 (0xFFxx) Processing pause in Master Mode in xx = number (sec)

10 Transmission Setup

Start	Slave address	Function	Data	Checksum	End
3.5 * character time	8 Bit	8 Bit	N* 8 Bit	16 Bit	3.5 * character time

Start / End:

If there is no data on the Modbus or if there is a data pause of 3.5 * of the character time, the data acquisition is reset.
A new character on the bus is recognised and evaluated as the first character (address).
Example: 9600baud, no parity, one stop bit
0.93ms / character => about 3.3ms for the start detection

Slave Address (8bit = 1Byte):

The slave address of the device can be set via DIP switches in hex code using 8 switches.
The lowest bit is on the left (DIP 1). The highest bit is on the right (DIP 8).
The following addresses can be set (see also table address settings):
With the FS1704, the setting is made via the setup menu (see also chapter 7.3.1)
Address 0: no function
Address 1 to 247: specific device address (may only be used once in the system)
If the slave address '0' is sent, all devices that are switched to 1 to 247 accept the command (broadcast; however, there is no response!)
Address 248 to 254: no function
Address 255: Special address (device works as master and processes commands according to a pre-programmed structure. This may only be used once in the system.) [See point: Master setting]

Function code (8 bit = 1 byte):

The following function codes from the general Modbus protocol are implemented.
Code 03: Read register content (16bit) (of a read and write register)
Code 04: Read register content (16bit) (one only read register)
Code 06: Write to register (16bit) - one register
Code 16: Write to registers (16bit) - several consecutive registers (max. 10)

Register (16bit = 2Byte):

For a description, see chapter 3 Register assignment.

Number of registers (16bit = 2byte):

To limit the transmission time / character strings, the number of registers is limited to a maximum of 10 [0x0001 to 0x000a]

Checksum (16Bit = 2Byte):

The checksum is determined according to the guidelines of a Modbus protocol.
This creates a 16-bit value which is appended to the character string with the LO and HI byte.

11 Command structure for registers

11.1 Reading from Read/Write registers

03 (0x03) read register [read/write register] (16bit)

Request:

slave address	0x00 … 0xff
function code	**0x03**
start register	register HI
start register	register LO
number of registers	number of registers HI
number of registers	number of registers LO
checksum	check LO
checksum	check HI

Answer:

slave address	0x00 … 0xff
function code	**0x03**
number of bytes	Number [n] of register values (bytes = n * 2)
1. register value	value HI
1. register value	value HO
n. register value	value HI
n. register value	value LO
checksum	check LO
checksum	check HI

If the register is faulty (see register assignment)

slave address	0x00 … 0xff
function code	**0x83**
error code	**0x02**
checksum	check LO
checksum	check HI

If the number of registers is incorrect (>= 0x000a) [max. 10 *]

slave address	0x00 … 0xff
function code	**0x83**
error code	**0x03**
checksum	check LO
checksum	check HI

11.2 Read from Read Register

04 (0x04) read register [only read register] (16bit)

Request:

slave address	0x00 … 0xff
function code	**0x04**
start register	register HI
start register	register LO
number of registers	number of registers HI
number of registers	number of registers LO
checksum	check LO
checksum	check HI

Answer:

slave address	0x00 … 0xff
function code	**0x04**
number of bytes	number [n] of register values (bytes = n * 2)
1. register value	value HI
1. register value	value HO
n. register value	value HI
n. register value	value LO
checksum	check LO
checksum	check HI

If the register is faulty (see register assignment)

slave address	0x00 … 0xff
function code	**0x84**
error code	**0x02**
checksum	check LO
checksum	check HI

If the number of registers is incorrect (>= 0x000a) [max. 10 *]

slave address	0x00 … 0xff
function code	**0x84**
error code	**0x03**
checksum	check LO
checksum	check HI

11.3 Writing to a Write Register

06 (0x06) write simply registers (16bit)

Request:

slave address	0x00 … 0xff
function code	**0x06**
register	register HI
register	register LO
register value	value HI
register value	value LO
checksum	check LO
checksum	check HI

Answer:

slave address	0x00 … 0xff
function code	**0x06**
register	register HI
register	register LO
register value	value HI
register value	value LO
checksum	check LO
checksum	check HI

If the register is faulty (see register assignment)

slave address	0x00 … 0xff
function code	**0x86**
error code	**0x02**
checksum	check LO
checksum	check HI

If the value range is incorrect

slave address	0x00 … 0xff
function code	**0x84**
error code	**0x03**
checksum	check LO
checksum	check HI

If values are transmitted that are outside the measuring range, they are limited to the measuring range and used. The error message (error code 0x03) is still sent.

11.4 Writing to several Write Registers

16 (0x10) write multiple registers (16bit)

Request:

slave address	0x00 … 0xff
function code	**0x10**
start register	register HI
start register	register LO
number of registers	number of registers HI
number of registers	number of registers LO
number of bytes	number of register (n) * 2
1. register value	value HI
1. register value	value LO
n. register value	value HI
n. register value	value LO
checksum	check LO
checksum	check HI

Answer:

slave address	0x00 … 0xff
function code	**0x10**
start register	register HI
start register	register LO
number of registers	number of registers HI
number of registers	number of registers LO
checksum	check LO
checksum	check HI

If the register is faulty (see register assignment)

slave address	0x00 … 0xff
function code	**0x90**
error code	**0x02**
checksum	check LO
checksum	check HI

If the number of registers is incorrect (> = 0x000a) [max. 10 *] or if the value range is incorrect

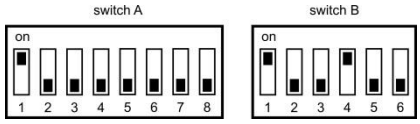
slave address	0x00 … 0xff
function code	**0x90**
error code	**0x03**
checksum	check LO
checksum	check HI

If values are transmitted that are outside the measuring range, they are limited to the measuring range and used. The error message (error code 0x03) is still sent.

Note: The response time after the correct request has been made depends on the measuring system. In general, this is less than 250ms (usually less than 50ms). If it is not possible for the system to answer within 250ms, there will be no further answer. It is recommended to set a maximum response time of 300ms in the Master.

12 BUS Parametrisation

Two DIP switches are used to set interface parameters.



The switch (Switch A) is used to specify a device-specific address.
It is essential to ensure that only one address is used in a BUS system. The address settings are shown in the table below.

The data transmission is set with the switch (Switch B).

DIP 1: ON	=>	then DIP 2 no function
	=>	8N1 mode (8 data bits, 1 stop bit, no parity)
DIP 1: OFF		
DIP 2: OFF	=>	8E1 (8 data bits, 1 stop bit, even parity)
DIP 2: ON	=>	8O1 (8 data bits, 1 stop bit, odd parity)
DIP 3: ON	=>	two stop bits are used
DIP 4: OFF and DIP 5: OFF	=>	Baud rate: 2400
DIP 4: ON and DIP 5: OFF	=>	Baud rate: 9600
DIP 4: OFF and DIP 5: ON	=>	Baud rate: 19200
DIP 4: ON and DIP 5: ON	=>	Baud rate: 38400

Note: With large BUS systems or when using long cables, interference can occur at high baud rates. In such a case, it is recommended to use a lower baud rate.

DIP 6: ON => Connection of a terminating resistor (220 Ohm)
This is used at the end of a BUS chain.
The connection of several terminating resistors in a BUS system can lead to a current overload on the data bus and thus to transmission errors.

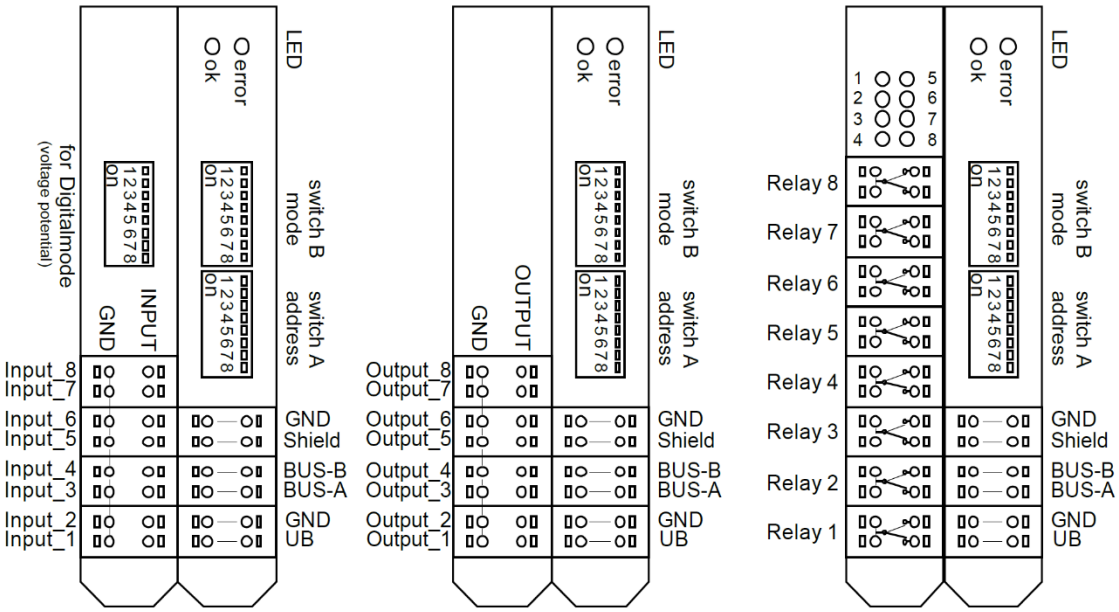
12.1 Parameterisation in the DIN rail Housing FS1701 / FS1702 / FS1703

Compared to the general sensor systems, the actuators in the DIN top hat rail housing with switch B have eight switch positions.

DIP 1 to DIP 5 are unchanged from the section **Fehler! Verweisquelle konnte nicht gefunden werden.**.

DIP 6: ON	=>	Connection of a pull-up resistor (1.5 kOhm) Output A against + 5V
DIP 7: ON	=>	Connection of a pulldown resistor (1.5 kOhm) Output B against GND
DIP 8: ON	=>	Connection of a terminating resistor (220 Ohm) This is used at the end of a BUS chain.

FS1701 **FS1702** **FS1703**



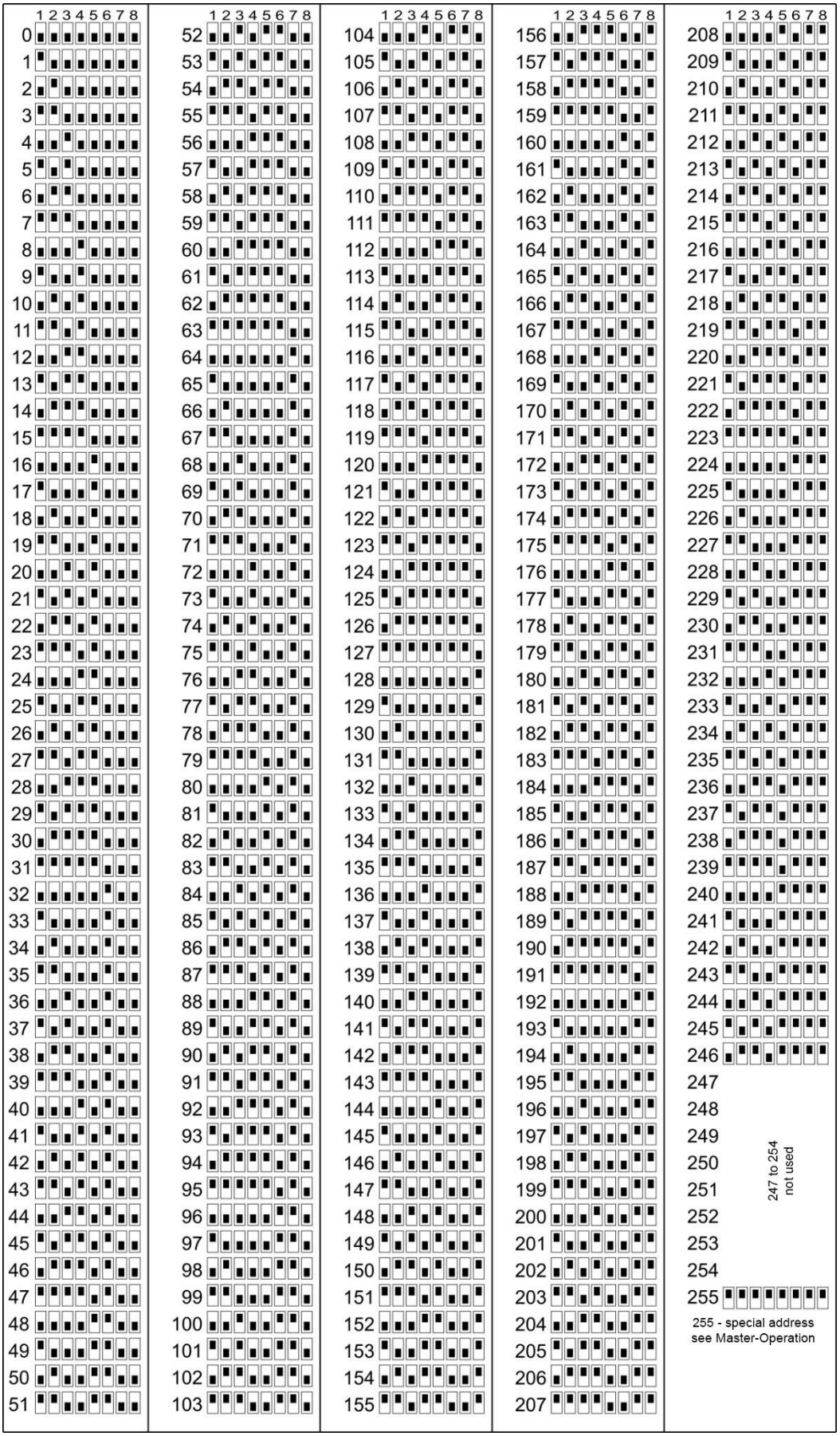
Scheme of DIP switches and connections for actuators

12.2 Settings in the Modbus service display FS1704 and Multi-Sensor transmitter FS1600

The settings of MODBUS addresses and the data transfer are made via the software. See also the chapter 7.3.1

A hardware-side connection of pullup, pulldown and terminating resistor takes place via DIP switches on the circuit board (labelled).

Address setting table via switch A



13 Used System Code

13.1 General

The measuring system is programmed by the manufacturer with regard to the measuring properties. A defined system code is stored for this and can also be called up via Rreg_0.

case Room	system code	case On-Wall	system code	Measurement property	Relay	Display
X	03	X	4099	only display and relay function (no sensors)	X	X
X	04	X	4100	temperature PT100		
X	05	X	4101	temperature PT100		X
X	06	X	4102	temperature PT100	X	
X	07	X	4103	temperature PT100	X	X
X	08	X	4104	humidity/temperature		
X	09	X	4105	humidity/temperature		X
X	10	X	4106	humidity/temperature	X	
X	11	X	4107	humidity/temperature	X	X
	12	X	4108	humidity heated/temperature		
	13	X	4109	humidity heated/temperature		X
	14	X	4110	humidity heated/temperature	X	
	15	X	4111	humidity heated/temperature	X	X
X	16	X	4112	CO_2		
X	17	X	4113	CO_2		X
X	18	X	4114	CO_2	X	
X	19	X	4115	CO_2	X	X
X	20	X	4116	CO_2 / temperature		
X	21	X	4117	CO_2 / temperature		X
X	22	X	4118	CO_2 / temperature	X	
X	23	X	4119	CO_2 / temperature	X	X
X	24	X	4120	CO_2 / humidity / temperature		
X	25	X	4121	CO_2 / humidity / temperature		X
X	26	X	4122	CO_2 / humidity / temperature	X	
X	27	X	4123	CO_2 / humidity / temperature	X	X
X	32	X	4128	CO		
X	33	X	4129	CO		X
X	34	X	4130	CO	X	
X	35	X	4131	CO	X	X
X	36	X	4132	CO / temperature		
X	37	X	4133	CO / temperature		X
X	38	X	4134	CO / temperature	X	
X	39	X	4135	CO / temperature	X	X
X	40	X	4136	CO / humidity / temperature		
X	41	X	4137	CO / humidity / temperature		X
X	42	X	4138	CO / humidity / temperature	X	
X	43	X	4139	CO / humidity / temperature	X	X
X	64	X	4160	VOC		
X	65	X	4161	VOC		X
X	66	X	4162	VOC	X	
X	67	X	4163	VOC	X	X
X	68	X	4164	VOC / temperature		
X	69	X	4165	VOC / temperature		X
X	70	X	4166	VOC / temperature	X	
X	71	X	4167	VOC / temperature	X	X
X	72	X	4168	VOC / humidity / temperature		
X	73	X	4169	VOC / humidity / temperature		X

X	74	X	4170	VOC / humidity / temperature	X	
X	75	X	4171	VOC / humidity / temperature	X	X

Case Room	system code	Case On-Wall	system code	Measurement property	Relay	Display
X	80	X	4176	CO_2 / VOC		
X	81	X	4177	CO_2 / VOC		X
X	82	X	4178	CO_2 / VOC	X	
X	83	X	4179	CO_2 / VOC	X	X
X	84	X	4180	CO_2 / VOC / temperature		
X	85	X	4181	CO_2 / VOC / temperature		X
X	86	X	4182	CO_2 / VOC / temperature	X	
X	87	X	4183	CO_2 / VOC / temperature	X	X
X	88	X	4184	CO_2 / VOC / humidity / temperature		
X	89	X	4185	CO_2 / VOC / humidity / temperature		X
X	90	X	4186	CO_2 / VOC / humidity / temperature	X	
X	91	X	4187	CO_2 / VOC / humidity / temperature	X	X
X	96	X	4192	CO / VOC		
X	97	X	4193	CO / VOC		X
X	98	X	4194	CO / VOC	X	
X	99	X	4195	CO / VOC	X	X
X	100	X	4196	CO / VOC / temperature		
X	101	X	4197	CO / VOC / temperature		X
X	102	X	4198	CO / VOC / temperature	X	
X	103	X	4199	CO / VOC / temperature	X	X
X	104	X	4200	CO / VOC / humidity / temperature		
X	105	X	4201	CO / VOC / humidity / temperature		X
X	106	X	4202	CO / VOC / humidity / temperature	X	
X	107	X	4203	CO / VOC / humidity / temperature	X	X
		X	4224	air flow 5m/s		
		X	4225	air flow 5m/s		X
		X	4226	air flow 5m/s	X	
		X	4227	air flow 5m/s	X	X
		X	4228	air flow 5m/s / temperature		
		X	4229	air flow 5m/s / temperature		X
		X	4230	air flow 5m/s / temperature	X	
		X	4231	air flow 5m/s / temperature	X	X
		X	4480	air flow 20m/s		
		X	4481	air flow 20m/s		X
		X	4482	air flow 20m/s	X	
		X	4483	air flow 20m/s	X	X
		X	4484	air flow 20m/s / temperature		
		X	4485	air flow 20m/s / temperature		X
		X	4486	air flow 20m/s / temperature	X	
		X	4487	air flow 20m/s / temperature	X	X
		X	4608	oxygen O_2		
		X	4609	oxygen O_2		X
		X	4610	oxygen O_2	X	
		X	4611	oxygen O_2	X	X
		X	4612	oxygen O_2 / temperature		
		X	4613	oxygen O_2 / temperature		X
		X	4614	oxygen O_2 / temperature	X	
		X	4615	oxygen O_2 / temperature	X	X

Case Room	system code	Case On-Wall	system code	Measurement property	Relay	Display
		X	8196	atm. / baro air pressure		
		X	8197	atm. / baro air pressure		X
		X	8198	atm. / baro air pressure	X	
		X	8199	atm. / baro air pressure	X	X
		X	8200	differential pressure up to 100Pa		
		X	8201	differential pressure up to 100Pa		X
		X	8202	differential pressure up to 100Pa	X	
		X	8203	differential pressure up to 100Pa	X	X
		X	8204	atm. / baro / differential pressure up to 100Pa		
		X	8205	atm. / baro / differential pressure up to 100Pa		X
		X	8206	atm. / baro / differential pressure up to 100Pa	X	
		X	8207	atm. / baro / differential pressure up to 100Pa	X	X
		X	8208	differential pressure up to 500Pa		
		X	8209	differential pressure up to 500Pa		X
		X	8210	differential pressure up to 500Pa	X	
		X	8211	differential pressure up to 500Pa	X	X
		X	8212	atm. / baro / differential pressure up to 500Pa		
		X	8213	atm. / baro / differential pressure up to 500Pa		X
		X	8214	atm. / baro / differential pressure up to 500Pa	X	
		X	8215	atm. / baro / differential pressure up to 500Pa	X	X
		X	8224	differential pressure up to 5000Pa		
		X	8225	differential pressure up to 5000Pa		X
		X	8226	differential pressure up to 5000Pa	X	
		X	8227	differential pressure up to 5000Pa	X	X
		X	8228	atm. / baro / differential pressure up to 5000Pa		
		X	8229	atm. / baro / differential pressure up to 5000Pa		X
		X	8230	atm. / baro / differential pressure up to 5000Pa	X	
		X	8231	atm. / baro / differential pressure up to 5000Pa	X	X
X	28800	X	24704	brightness		
X	28802	X	24706	brightness	X	
X	28928	X	24832	motion		
X	28930	X	24834	motion	X	
X	29056	X	24960	motion / brightness		
X	29058	X	24962	motion / brightness	X	
X	28808	X	24712	brightness / humidity / temperature		
X	28810	X	24714	brightness / humidity / temperature	X	
X	28936	X	24840	motion / humidity / temperature		
X	28938	X	24842	motion / humidity / temperature	X	
X	29064	X	24968	motion / brightness / humidity / temperature		
X	29066	X	24970	motion / brightness / humidity / temperature	X	

Case In-Wall	*system code*			Measurement property	*Relay*	*Display*
X	36872			humidity / temperature		
X	36880			CO_2		
X	36884			CO_2 / temperature		
X	36888			CO_2 / humidity / temperature		
X	36928			VOC		
X	36932			VOC / temperature		
X	36936			VOC / humidity / temperature		
X	36944			CO_2 / VOC		
X	36948			CO_2 / VOC / temperature		
X	36952			CO_2 / VOC / humidity / temperature		

X	32896			brightness		
X	32904			brightness / humidity / temperature		
X	33024			motion		
X	33032			motion / humidity / temperature		
X	33152			brightness / motion		
X	33160			brightness / motion / humidity / temperature		

13.2 Modbus-Servicedisplay

The measuring system is programmed by the manufacturer with regard to the measuring properties. A defined system code is stored for this and can be called up via Rreg_0.
The measurement properties are bit-coded. One or more set bits in Rreg_0 define the system code and thus the measurement property.

system code	***Measurement property***
Bit_0	humidity and temperature measurement
Bit_1	CO_2 measurement
Bit_2	VOC measurement
Bit_3	CO measurement
Bit_4	O_2 measurement
Bit_5	atm. and barom. air pressure
Bit_6	differential pressure (excludes PT100 temperature measurement)
Bit_7	brightness sensor
Bit_8	motion sensor (excludes heated humidity sensor)
Bit_9	air flow sensor
Bit_10	particulate matter measurement
Bit_11	PT100 temperatur measurement if Bit_0 = 1 then heated humidity sensor (excludes differential pressure and motion sensor)
Bit_14	90° rotated display for din rail mounting
Bit_15	always to 1 (coding for FS1600)

Example:

Rreg_0 (system code) = 0x8007
It is a universal measuring and display system (FS1600 - Multi-Sensor-Measuring Device), which has sensors for: humidity and temperature, CO_2, VOC

With regard to the special features of individual measured variables, reference is made to the respective section descriptions in chapter **Fehler! Verweisquelle konnte nicht gefunden werden.** and are equivalent to these.

Address

FuehlerSysteme eNET International GmbH
Roethensteig 11
D-90408 Nürnberg

Phone

+49 911 37322-0

Fax

+49 911 37322-111

E-Mail & Web

info@fuehlersysteme.de
www.fuehlersysteme.de

Technical Support

+49 1805 858511*

** 14 ct/min. from german network - max. 42 ct/min. from german mobile phone*