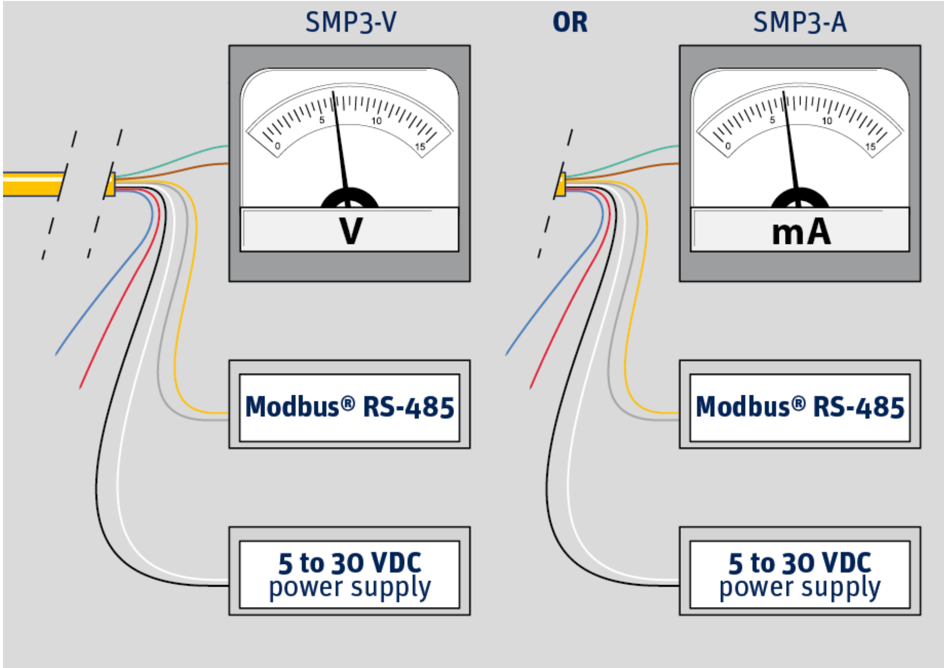
1. Electrical installation

Standard the SMP is supplied with a waterproof connector pre-wired to 10 m cable with 8 leads and a shield covered with a black sleeve. The colour code of the wires and the connector pin numbers are shown below and on the instruction sheet. Longer cables are available as options. Normally the analogue or the digital interface is connected for read-out.



It is advised not to connect the unused interface in order to reduce power consumption.

Connect power supply ground also to Modbus ground.

Figure 5 Cable functional connections

PYRANOMETER CONNECTION
ANSCHLUSS • RACCORDEMENT • CONEXIÓN

Wire Kabel Fil Cable	Function Funktion Fonction Función	Connect with Anschluss an Relier à Conectar con
3 Green Grün • Vert • Verde	Analogue out	V+/4-20 mA(+)
6 Brown Braun • Brun • Marrón	Analogue ground	V–/4-20 mA(–)
4 Yellow Gelb • Jaune • Amarillo	Modbus® RS-485	B/B'/+
5 Grey Grau • Gris • Gris	Modbus® RS-485	A/A'/–
7 White Weiss • Blanc • Blanco	Power 5 to 30 VDC (12 V recommended)	
8 Black Schwarz • Noir • Negro	Power ground / RS-485 Common	
1 Red Rot • Rouge • Rojo	None	Not connected
2 Blue Blau • Bleu • Azul	None	Not connected
Shield Abschirmung Protection Malla	Housing Gehäuse Boîte Cubierta	Ground * Erde Terre Tierra

* Connect to ground if radiometer not grounded
Mit Erde verbinden, wenn das Radiometer nicht geerdet ist
Reliez à la terre si le radiomètre n'est pas connecté
Conectar a tierra si el radiómetro no lo está

Figure 6 SMP cable wires, colours and functions

Preferably, secure the radiometer with its levelling screws to a metal support with a good connection to earth (e.g. by using a lightning conductor).

The shield of the cable is connected to the aluminium radiometer housing through the connector body. The shield at the cable end may be connected to ground at the readout equipment. Lightning can induce high voltages in the shield but these will be led off at the pyranometer and data logger.

Long cables may be used, but the cable resistance must be smaller than 0.1% of the impedance of the readout equipment.

1.1. Power connection

The minimum power requirement for the SMP's is 5V, however for optimal performance it is advised to use 12 VDC especially when long cables are used. The 5 volt power can only be used in combination with a short (10 m) cable.

It is advised to protect the output of the power supply with a fast blowing fuse of max 250mA.

Typical power consumption SMPX-V:

5V	max 50 mW	(approx. 10.0 mA)
12V	max 55 mW	(approx. 4.5 mA)
24V	max 60 mW	(approx. 2.5 mA)

Maximum power consumption and input current.

65mW at the highest input voltage
12.5 mA at the lowest input voltage

The maximum inrush current is 200 mA. After completing the installation the pyranometer will be ready for operation.

Typical power consumption SMPX-A

12V	max 100 mW	(approx. 8.5 mA)

Maximum power consumption and input current. @@

Special attention is asked to prevent power or ground loops when connecting the SMP to multiple readout devices. Connecting the RS-485 to a grounded circuit and the analogue output to a not grounded circuit can cause unacceptable ground loops. This causes voltages outside the SMP specifications and can destroy the unit. We recommend using either the analogue or the digital output and not both.

1.2. Data connection

The digital interface can only be connected to a 2 wire RS485 network as shown below.

RS485 connection

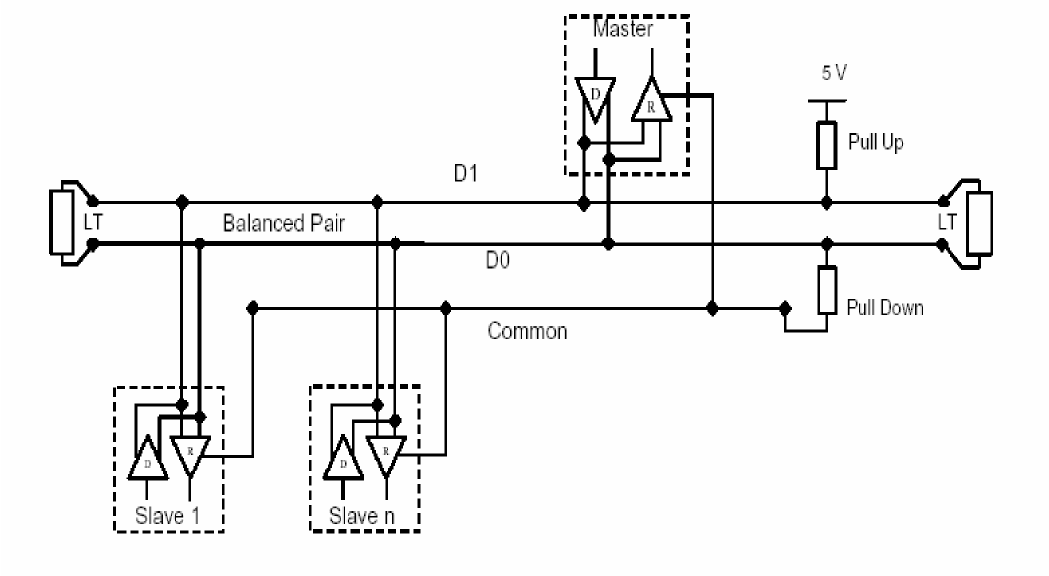


Figure 7 RS-485 connections (@@ lijn om plaatje moet weg)

The interface needs also an external power to provide the voltage for the electronics. If the interface is the last device on the network then a terminator consisting of a 120Ω or 150Ω resistor has to be connected between terminal D0 and D1. Never place the line termination on the derivation cable. It is required to install the pull up and pull down resistors as shown in figure 7. The value of these resistors has to be within 650Ω and 850Ω.

.

1.3. Analogue output connection

The SMP3-V and SMP11-V (Volt version) have been factory set to an output of -200 to 2000 W/m². This applies only to the analogue output and means that an output of 0 Volt corresponds to -200 W/m² (this will never be reached) and 1 Volt corresponds to 2000 W/m². The output range can be modified by the user with the supplied PC software. For the SMP11 the output range can be set to -200 to 4000 W/m² for 0 – 1 Volt.
The range has to start negative in order to show (small) negative readings also the analogue output itself cannot go negative. If used in atmospheric conditions it is advised to keep the range as factory set.
The same applies for the SMP3-A and SMP11-A versions that have also been factory set to -200 to 2000 W/m² for 4 – 20 mA.
Here negative inputs will make the output go under 4 mA.

1.4. Calculations

1.4.1. Calculation 0 – 1 Volt version

The output of a pyranometer can go slightly negative at night. Due to the FIR radiation the outer dome can cool down below air temperature (zero offset type A).
In order to show negative values, the output is defined from 0 – 1 Volt representing -200 to 2000 W/m^2.
The irradiance value ($E_{\downarrow Solar}$) can be simply calculated as shown below in Formula 1.
The formula assumes the factory default setting of the analogue output.
For calculation of the solar irradiance (global or reflected) the following formula must be applied:

$$E_{\downarrow Solar} = (V \times 2200) - 200 \qquad \textit{Formula 1}$$

$E_{\downarrow Solar}$	= Solar radiation	[W/m^2]
V	= Output of radiometer	[Volt]

1.4.2. Calculation 4 – 20 mA version

The output is defined from 4 – 20 mA representing 0 to 1600 W/m^2.
Negative outputs can cause the output to go slightly below 4 mA
The irradiance value ($E_{\downarrow Solar}$) can be simply calculated as shown below in Formula 2.
The formula assumes the factory default setting of the analogue output.
For calculation of the solar irradiance (global or reflected) the following formula must be applied:

$$E_{\downarrow Solar} = (mA - 4) \times 100 \qquad \text{Formula 2}$$

$E_{\downarrow Solar}$	= Solar radiation	[W/m^2]
mA	= Output of radiometer	[mA]

2. Software

The SMP's come with the program SmartSensorDemo.exe, this can be used to configure the SMP's settings. The program is for testing the instrument and setting the parameters. Basic data logging and display of data on a PC is possible.

2.1. Setup with a PC

Before installation, disconnect all Modbus devices and connect the device you want to install.

The software can handle up to 10 pyranometers at the same time connected to the same RS-485 interface.(see Fig. 7)

The setup depends on the used RS-485 adapter. An interface that could be used is from B&B, model USOPTL4, that has galvanic isolation between in- and output. One side holds the USB connector to the PC the other side has a connector for connection of the wires. This RS485 adapter is powered from its USB interface.

A power supply for the SMP is required, this can be from a 5 -30 Volt. A 12 VDC is advised.

For set-up the following connection to a PC can be used. It is advised to use a RS-485 interface with galvanic isolation between PC and RS-485.
Care has to be taken when connecting to an ungrounded (portable) PC, especially when also the analogue output is used. The Mains filter inside the PC can generate huge voltage peaks that can exceed the limits of the SMP pyranometers.

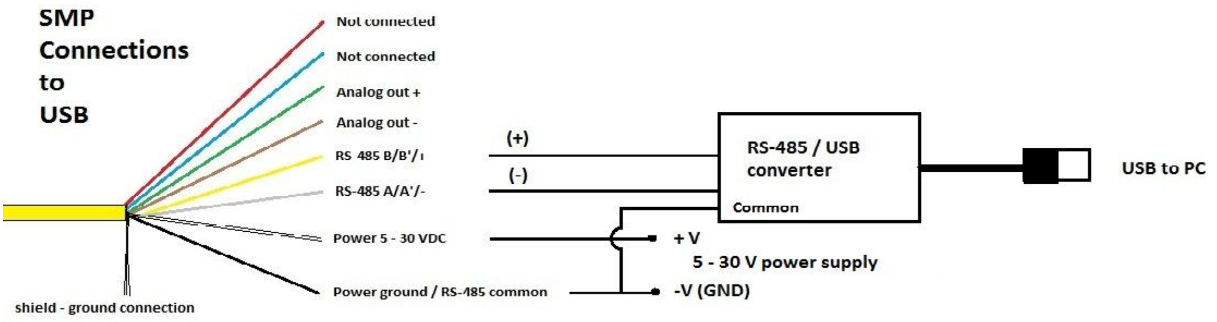


Figure 8 SMP demo set-up to PC / USB

2.1.1. Setup communication

When the SmartSensorDemo program is opened there are 3 basic functions called, Overview, Instrument and Chart, located under the tabs with these names.

With the Overview tab active the communication parameters can be set as follows:

- Select the communication port where the RS-485 is connected (here COM 7)
- Set the baud rate to 19200 baud (this is the standard configuration)
- Set the number of data bits and parity to 8 bits – even parity (this is the standard configuration)
- Press the button “Discover first connected device” in tab Overview
- Select device 01 or select Poll All Devices.
- Show Advanced features in the upper right hand corner brings up the possibility to change the Modbus address.

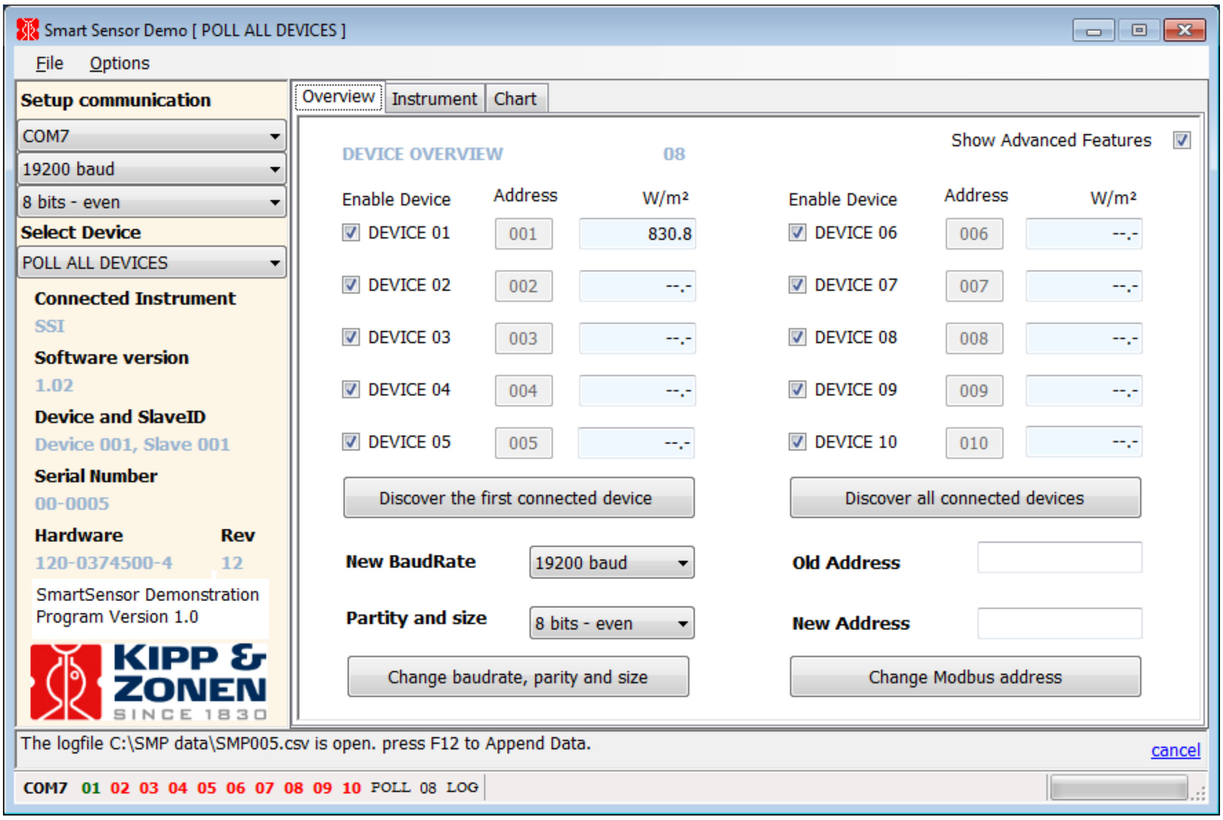


Figure 9 Overview menu

2.1.2. Change the Modbus address

The default configured MODBUS address of all SMP's is 1. Before you can use the smart sensor in your network you must reconfigure the MODBUS address to a unique address. When the device is connected to the network with other Modbus devices, each connected device must have a unique Modbus address.

- Enter the old Modbus address (this address can be found with the button: "discover connected devices" (example: enter 1).
- Enter the new Modbus address. This must be a unique number (example: enter 24).
- Press the button: "Change Modbus address".

When the Modbus address is changed the device will restarts itself.

2.1.3. Check the configuration

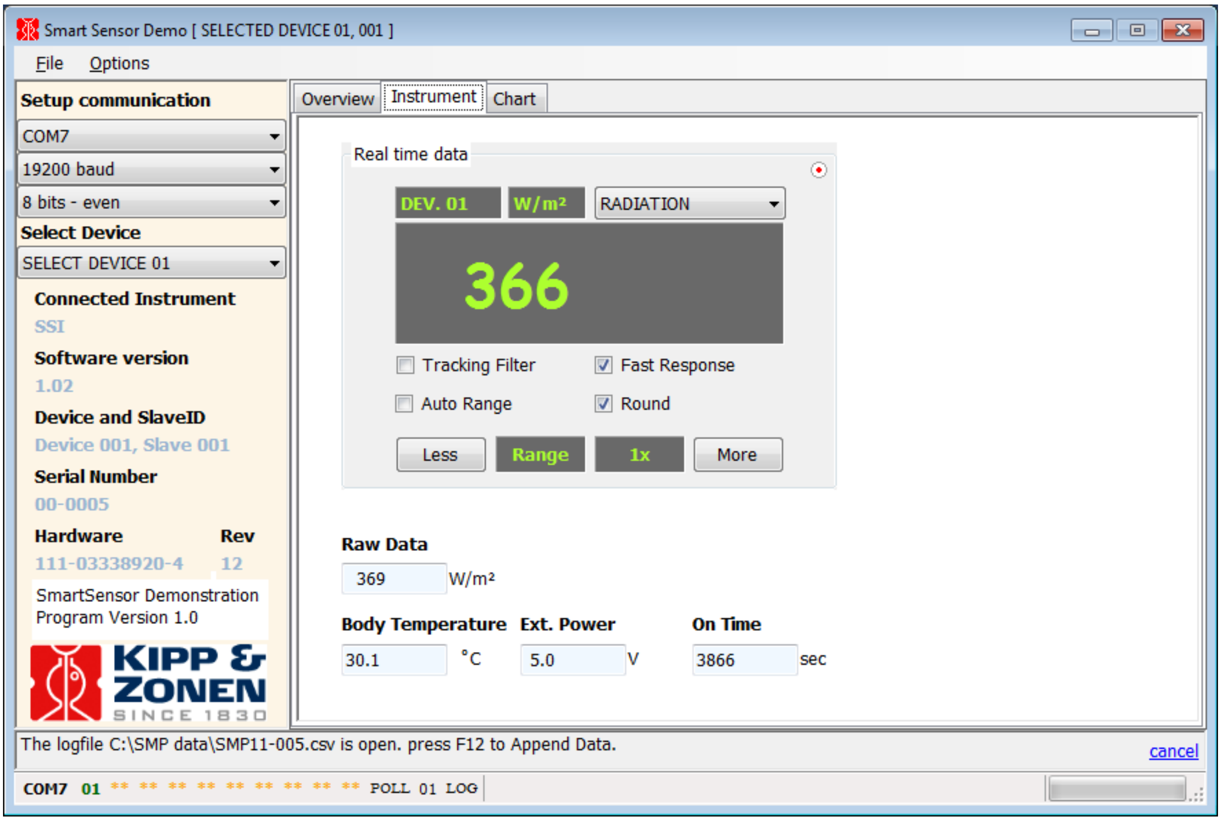


Figure 10 Instrument menu

The standard configuration is:

- Tracking filter ON
- Fast response filter ON
- Auto Range OFF
- Round off ON
- Range 1x

When the power is removed and connected the smart sensor will start up with the standard configuration:

2.1.4. Chart

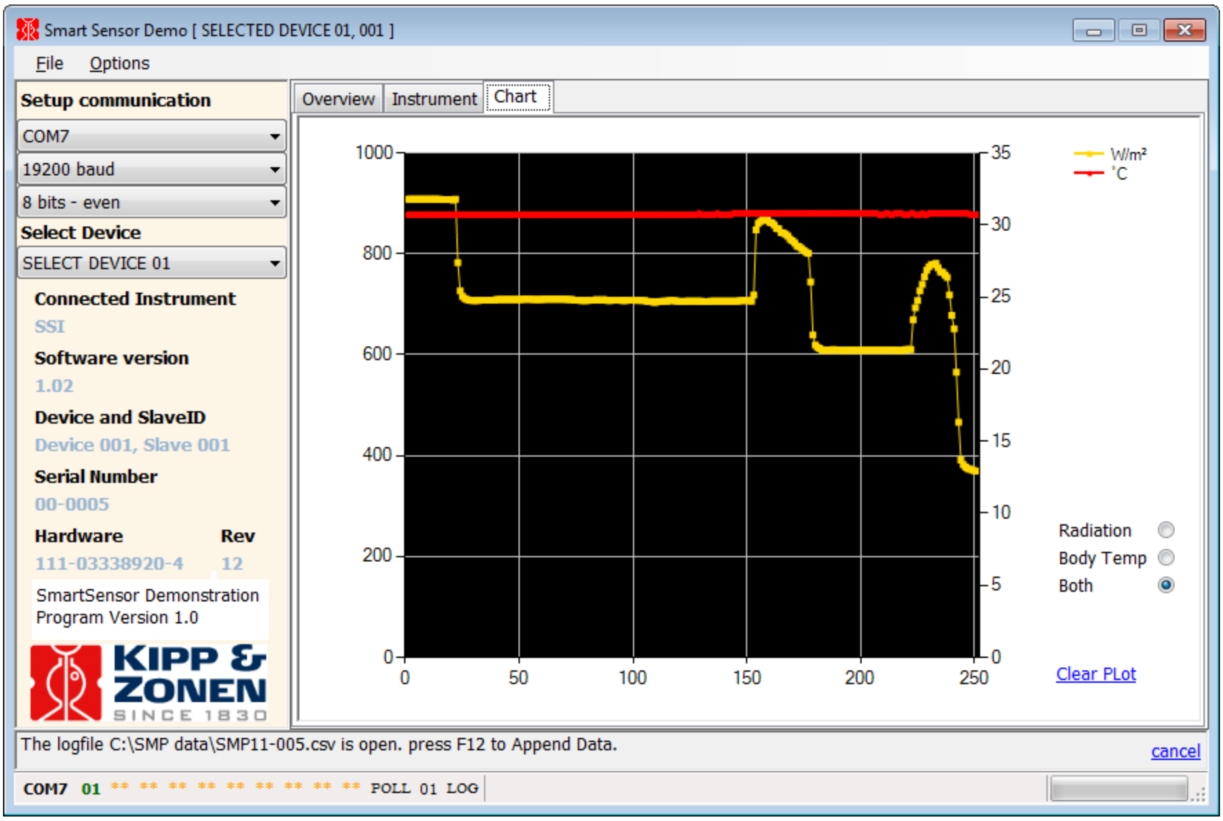


Figure 11 Chart menu

The third tab (Chart) will show the radiation data (in W/m²) and/or the body temperature (in °C) of the SMP pyranometer as an accumulating chart. Vertical scaling is automatic. The horizontal scale shows the last 250 measurements. Scaling is automatic.

2.1.5. Data logging

Under File the data logging can be set. The data format options are txt or csv. The (CSV) log file has the following format:

SMP-05.csv

	A	B	C	D	E	F	G	H
1	[11-1-2012 10:47:26]							
2								
3	DATE	TIME	SLAVE	SERIALNR	RADIANCE W/m2	TEMP C	POWER V	
4	11-1-2012	10:47:33	1	00-0005	826.6	29.9	5.0	
5	11-1-2012	10:47:43	1	00-0005	825.5	29.9	5.0	
6	11-1-2012	10:47:53	1	00-0005	826.6	29.9	5.0	
7	11-1-2012	10:48:03	1	00-0005	827.1	29.9	5.0	
8	11-1-2012	10:48:13	1	00-0005	827.9	29.9	5.0	
9	11-1-2012	10:48:24	1	00-0005	828.0	29.9	5.0	
10	11-1-2012	10:48:34	1	00-0005	827.0	29.9	5.0	

Figure 12 CSV file example

Beside the Modbus address and radiation data also body temperature and (SMP) power supply voltage are recorded.

In the lower part of the previous PC screen shots the open logfile C:\SMPdata\SMP11-005.csv is visible.

With the append function (F12) a new data set can be linked to an existing file.

Below is an example of a txt file with linked data from 2 days.

```
[11-1-2012 11:08:14]
DATE;TIME;SLAVE;SERIALNR;RADIANCE W/m2;TEMP C;POWER V
2012-01-11;11:08:15;001;00-0005;  708;30.7; 5.0;
2012-01-11;11:08:16;001;00-0005;  708;30.7; 5.0;
2012-01-11;11:08:17;001;00-0005;  708;30.7; 5.0;

[12-1-2012 9:20:17]
DATE;TIME;SLAVE;SERIALNR;RADIANCE W/m2;TEMP C;POWER V
2012-01-12;09:20:30;001;00-0005;  928;22.8; 5.0;
2012-01-12;09:20:37;001;00-0005;  929;22.8; 5.0;
2012-01-12;09:20:38;001;00-0005;  929;22.8; 5.0;
```

3. Specifications

3.1. SMP series performance specifications

Specifications	SMP3	SMP11
ISO 9060:1990 CLASSIFICATION	Second Class	Secondary Standard
Response time (63 %)	< 1.5 s	< 0.7 s
Response time (95 %)	< 12 s	< 2 s
Zero offsets (a) thermal radiation (200 W/m^2) (b) temperature change (5 K/hr)	 < 15 W/m^2 < 5 W/m^2	 < 7 W/m^2 < 2 W/m^2
Non-stability (change/year)	< 1 %	< 0.5 %
Non-linearity (0 to 1000 W/m^2)	< 1 %	< 0.2 %
Directional error (up to 80 ° with 1000 W/m^2 beam)	< 20 W/m^2	< 10 W/m^2
Temperature dependence of sensitivity	< 3 % (-20 °C to +50 °C) < 5 % (-40 °C to +70 °C)	< 1 % (-20 °C to +50 °C) < 2 % (-40 °C to +70 °C)
Tilt error (at 1000 W/m^2)	< 1%	< 0.2%
Other specifications		
Analogue output	-V version: 0 to 1 V -A version: 4 to 20 mA	-V version: 0 to 1 V -A version: 4 to 20 mA
Analogue output range	-V version: -200 to 2000 W/m^2 -A version: 0 to 1600 W/m^2	-V version: -200 to 2000 W/m^2 -A version: 0 to 1600 W/m^2
Digital output	2-wire RS-485	2-wire RS-485
Digital output maximum range	-400 to 2000 W/m^2	-400 to 4000 W/m^2 (1)
Digital communication protocol	Modbus®	Modbus®
Level accuracy	1 °	0.1 °
Operating temperature	-40 °C to +80 °C	-40 °C to +80 °C
Ingress Protection (IP)	67	67
Spectral range (50 % points)	300 to 2800 nm	285 to 2800 nm
Supply voltage	5 to 30 VDC	5 to 30 VDC
Power consumption (at 12 VDC)	-V version: 55 mW -A version: 100 mW	-V version: 55 mW -A version: 100 mW
Expected daily uncertainty	< 10 %	< 2 %
Documentation	Calibration certificate traceable to WRR, multi-language instruction sheet, manual on CD-ROM	Calibration certificate traceable to WRR, multi-language instruction sheet, manual on CD-ROM
Recommended applications	Economical solution for efficiency and maintenance monitoring of PV power installations, routine measurements in weather stations, agriculture, horticulture and hydrology	High performance for PV panel and thermal collector testing, solar energy research, solar prospecting, materials testing, advanced meteorology and climate networks

(1) The analogue output range of SMP11 can be rescaled by the user to a maximum of 0 to 4000 W/m^2

SMP instruments have a standard cable length of 10 m. Optional cable lengths 25 m and 50 m

Note: The performance specifications quoted are worst-case and/or maximum values

4. Appendices

4.1. MODBUS commands

The implemented commands are al according to the MODBUS RTU protocols described in the document: “MODBUS over serial line V1.02” and “MODBUS application protocol V1.1b” available from the MODBUS organization (www.modbus.org). The commands can be tested using software tools like the program: “MODBUS Poll” from www.modbustools.com.

The following commands are implemented:

Function	Sub function	Description
0x01	N/A	Read Coils
0x02	N/A	Read Discrete Inputs
0x03	N/A	Read Holding Registers
0x04	N/A	Read Input Register
0x05	N/A	Write Single Coil
0x06	N/A	Write Holding Register
0x10	N/A	Write multiple Registers

The SMP does not make a difference between a “coil” and a discrete input. The only difference is that a discrete input is read only.

The SMP does not make a difference between a holding register and an input registers. The only difference is that an input register is read only.

Overview input registers

Input registers are read only

REAL-TIME PROCESSED DATA					
REGISTER	PARAMETER	R/W	TYPE	MODE	DESCRIPTION
0	IO_DEVICE_TYPE	R	U16	ALL	DEVICE TYPE OF THE SENSOR
1	IO_DATAMODEL_VERSION	R	U16	ALL	VERSION OF THE OBJECT DATA MODEL 100=THIS VERSION
2	IO_OPERATIONAL_MODE	R	U16	ALL	OPERATIONAL MODE: NORMAL, SERVICE, CALIBRATION AND SO ON
3	IO_STATUS_FLAGS	R	U16	ALL	DEVICE STATUS FLAGS

REAL-TIME PROCESSED DATA					
4	IO_SCALE_FACTOR	R	S16	ALL	RANGE AND SCALE FACTOR SENSOR DATA *1
5	IO_SENSOR1_DATA	R	S16	N,S	TEMPERATURE COMPENSATED DATA SENSOR 1 IN W/M²
6	IO_RAW_SENSOR1_DATA	R	S16	N,S	SENSOR DATA SENSOR 1 IN W/M2
7	IO_STDEV_SENSOR1	R	S16	N,S	STANDARD DEVIATION SENSOR 1 IN 0.1 W/M²
8	IO_BODY_TEMPERATURE	R	S16	N,S	BODY TEMPERATURE IN 0.1 °C
9	IO_EXT_POWER_SENSOR	R	S16	N,S	EXTERNAL POWER VOLTAGE IN 0.1V
10-15	FACTORY USE ONLY				
16	IO_DAC_OUTPUT_VOLTAGE	R	U16	N,S	DAC OUTPUT VOLTAGE (ACTUAL VOLTAGE)
17	IO_SELECTED_DAC_INPUT	R	U16	N,S	DAC SELECTED INPUT DATA

REAL-TIME DATA A/D COUNTS					
REGISTER	PARAMETER	R/W	TYPE	MODE	DESCRIPTION
18 19	IO_ADC1_COUNTS	R	S32	ALL	INPUT VOLTAGE SENSOR1 IN 0.01UV (R18=MSB,R19=LSB)
20 21	IO_ADC2_COUNTS	R	S32	ALL	NOT SUPPORTED, ALWAYS 0
22 23	IO_ADC3_COUNTS	R	S32	ALL	INPUT VOLTAGE BODY TEMPERATURE SENSOR IN 0.01UV (R22=MSB,R23=LSB)
24 25	IO_ADC4_COUNTS	R	S32	ALL	INPUT VOLTAGE POWER SENSOR IN 0.01UV (R24=MSB,R25=LSB)

ERROR REPORTS					
REGISTER	PARAMETER	R/W *2	TYPE	MODE	DESCRIPTION
26	IO_ERROR_CODE	R	U16	ALL	MOST RECENT/ ACTUAL ERROR CODE
27	IO_PROTOCOL_ERROR	R	U16	ALL	PROTOCOL ERROR/COMMUNICATION ERROR
28	IO_ERROR_COUNT_PRIO1	R	U16	ALL	ERROR CODE PRIORITY 1
29	IO_ERROR_COUNT_PRIO2	R	U16	ALL	ERROR COUNT PRIORITY 2
30	IO_RESTART_COUNT	R	U16	ALL	NUMBER OF CONTROLLED RESTARTS
31	IO_FALSE_START_COUNT	R	U16	ALL	NUMBER OF UNCONTROLLED RESTARTS.
32	IO_SENSOR_ON_TIME	R	U16	ALL	ON TIME IN SECONDS (MSB WORD)
33	IO_SENSOR_ON_TIMEL	R	U16	ALL	ON TIME IN SECONDS (LSB WORD)

*1 The scale factor defines the format and number of decimal places.

*2 Writing any value to input registers 26-33 will reset the contents of the registers.

REGISTER	PARAMETER	R/W	TYPE	MODE	DESCRIPTION
41	IO_BATCH_NUMBER	R	U16	ALL	PRODUCTION BATCH NUMBER
42	IO_SERIAL_NUMBER	R	U16	ALL	SERIAL NUMBER
43	IO_SOFTWARE_VERSION	R	U16	ALL	SOFTWARE VERSION
44	IO_HARDWARE_VERSION	R	U16	ALL	HARDWARE VERSION
45	IO_NODE_ID	R	U16	ALL	(MODBUS/SMA) DEVICE ADDRESS RS485

Legend

Register	Modbus register	Modbus register 0 is the first register.	
Parameter	Name	Name of the register	
R/W	Read write	R	Read only
		R/W	Read/write
Type	Type and Size	U16	16 bit unsigned integer
		S16	16 bit signed integer
		S32	32 bit signed integer (MSB first, LSB last)
Mode	operation mode	N	available in normal mode
		S	available in service mode
		C	available in calibration mode (not for users)
		F	available in factory mode (not for users)
		All	available in all modes

Register can be read in all modes but some registers can't be written in normal mode or service mode.

Overview holding registers

DEVICE CONTROL					
REGISTER	PARAMETER	R/W	TYPE	MODE	DESCRIPTION
34	IO_DEF_SCALE_FACTOR	R/W	S16	ALL	DEFAULT SCALE FACTOR
35 - 40	FACTORY USE ONLY				

.

Read input register explained

Many of the registers and controls are for remote diagnostics. In this chapter only the most interesting registers and controls are described

Register 0 IO_DEVICE_TYPE

The device typed defines which device is connected. The default values are:

601= CMP 3 603= CMP 11

This register can be used to check the type of the connected device.

Register 1 IO_DATAMODEL_VERSION

The data-model describes the functions supported by the smart sensor. This document is valid for data-model version: “100” and “101”. A different implementation of the Modbus protocol (with new features) could result in a different data model “that is” or “that is not” compatible with the older version.

The value of this register must be “100” or “101”. If you receive another value then you should read an updated version of this document and check the differences.

Register 2 IO_OPERATIONAL_MODE

The operation mode defines the state of the smart sensor. The operational modes are 1 = Normal Mode, 2 = Service Mode, 3 = Calibration Mode, 4 = Factory Mode and 5 = Error mode. The standby mode (mode 0) is not supported.

After power on the operation mode (1) is set. When the IO_CLEAR_ERROR is set then the smart sensor always returns to the normal mode. When the Error mode (5) is set, then there is a fatal error.

Register 3 IO_STATUS_FLAGS

This register defines the status of the smart sensor and the validity of the data. Each bit has a special meaning. Bit 0 is the first (least significant) bit.

Bit 0	Quality of the signal	see IO_VOID_DATA_FLAG
Bit 1	Overflow	see IO_OVERFLOW_ERROR
Bit 2	Underflow	see IO_UNDERFLOW_ERROR
Bit 3	Error flag	see IO_ERROR_FLAG
Bit 4	ADC Error	see IO_ADC_ERROR
Bit 5	DAC Error	see IO_DAC_ERROR
Bit 6	Calibration Error	see IO_CALIBRATION_ERROR
Bit 7	Update EEPROM error	see IO_UPDATE_FAILED

Register 4 IO_SCALE_FACTOR

The scale factor defines the number of fractional digits, the range and the position of the decimal point for the following registers: IO_SENSOR1_DATA, IO_SENSOR2_DATA, IO_RAW_SENSOR1_DATA and IO_RAW_SENSOR2_DATA. The scale factor is read only. The default value of the scale factor is set during calibration mode or it can be changed during operation (see register IO_DEF_SCALE_FACTOR and coil IO_AUTO_RANGE).

If the register IO_SCALE_FACTOR is not set to 0 then you must multiply or divide the data of register (X), whereas X is one of the above mentioned registers.

Scale factor = 2	(floating point) result = (integer) register (X) / 100.0
Scale factor = 1	(floating point) result = (integer) register(X) / 10.0
Scale factor = 0	(floating point) result = (integer) register(X)
Scale factor = -1	(floating point) result = (integer) register(X) * 10.0

The default value of register IO_SCALE_FACTOR is 0. However, this value can be set to a different value if the coil IO_AUTO_RANGE is set or a different value is written to the register IO_DEF_SCALE_FACTOR (set default scale factor).

Register 5 IO_SENSOR1_DATA

This register holds the actual data (solar radiation) measured by the sensor. The solar radiation is measured in W/m².

If the register IO_SCALE_FACTOR is not set to 0 then you must multiply or divide the data as described under register 4.

The raw data from the sensor is calibrated, linearized; temperature compensated and filtered using 2 different kinds of filters (See IO_FAST_RESPONSE and IO_TRACKING_FILTER).

Register 6 IO_RAW_SENSOR1_DATA

The raw sensor data is calibrated but not linearized and temperature compensated. If the register IO_SCALE_FACTOR is not set to 0 then you must multiply or divide the data as described under register 4, IO_SCALE_FACTOR.

Register 7 IO_STDEV_SENSOR1

Register 7 is used to calculate the standard deviation over the signal. When the register is read the data is sent to the computer and at the same time a new calculation is started. The next time register 7 is read the standard deviation over the last period is sent to the computer and a new calculation is started. If the poll frequency is quite high (for example 1 poll per second) then the standard deviation will be zero or almost zero, but if the poll frequency is very low then the standard deviation can be quite high, indicating that the data in register 5 or 6 changed dramatically since the last poll. The standard deviation is measured in 0.1 W/m². To convert the data to a floating point, make the following calculation:

(floating point) result = (integer) register (IO_STDEV_SENSOR1) / 10.0

Register 8 IO_BODY_TEMPERATURE

The body temperature sensor measures the temperature of the body in 0.1 °C.

The convert the data to a floating point number, make the following calculation:

(floating point) result = (integer) register (IO_BODY_TEMPERATURE) / 10.0

Register 9 IO_EXT_POWER_SENSOR

The Ext power sensor measured the external voltage applied to the sensor in 0.1 Volt.

The convert the data to a floating point number, make the following calculation:

(floating point) result = (integer) register (IO_EXT_POWER_SENSOR) / 10.0

Example

Read registers: “operational mode to external power” from Modbus device with address 1.

Tx transmitted data to the smart sensor

Rx received data from the smart sensor

SendModbusRequest (0x04, 1, IO_OPERATIONAL_MODE, 8);

Tx 01 04 00 02 00 08 50 0C

Rx 01 04 10 00 01 00 00 00 00 03 E5 03 E5 00 00 00 F8 00 EA 66 12

Explanation of the received bytes:

01 = MODBUS address

04 = read input registers

10 = number of received data bytes

00 01 = operational mode (mode 1)

00 00 = status flags (none)

00 00 = scale factor = 0 = 1x

03 E5 = 997 decimal = sensor 1 data in W/m2

03 E5 = 997 decimal = raw sensor 1 data in W/m2

00 00 = 0 = standard deviation sensor 1

00 F8 = 248 = 24.8 ˚C.

00 EA = 234 = 23.4 Volt

66 12 = MODBUS checksum (CRC16)

Overview discrete inputs

A discrete input can be true or false. A discrete input is read only; a coil can be read or written.

STATUS INDICATORS					
INPUT	PARAMETER	R/W	DEF.	MODE	DESCRIPTION
0	IO_FALSE	R	0	ALL	ALWAYS FALSE (FOR TESTING ONLY)
1	IO_TRUE	R	1	ALL	ALWAYS TRUE (FOR TESTING ONLY)
2	IO_VOID_DATA_FLAG	R	*	ALL	VOID SIGNAL,1=UNSTABLE SIGNAL, TEMPERATURE TOO LOW OR TOO HIGH

STATUS INDICATORS					
3	IO_OVERFLOW_ERROR	R	*	ALL	OVERFLOW, SIGNAL OUT OF RANGE
4	IO_UNDEFLOW_ERROR	R	*	ALL	UNDERFLOW SIGNAL OUT OF RANGE
5	IO_ERROR_FLAG	R	*	ALL	GENERAL HARDWARE ERROR (SET IF ONE OF THE H/W ERROR FLAGS IS SET)
6	IO_ADC_ERROR	R	*	ALL	HARDWARE ERROR A/D CONVERTER
7	IO_DAC_ERROR	R	*	ALL	HARDWARE ERROR D/A CONVERTER
8	IO_CALIBRATION_ERROR	R	*	ALL	CALIBRATION CHECKSUM ERROR
9	IO_UPDATE_FAILED	R	*	ALL	UPDATE CALIBRATION PARAMETERS FAILED.

Legend

Input	Discrete input	Modbus discrete input 0 is the first discrete input	
Coil	Modbus Coil	A coil can be read or written.	
Parameter	Name	Name of the register	
R/W	Read write	R	Read only
		R/W	Read/write
Def	Default value	default value at power on (0, 1or *) * = undefined	
Mode	operation mode	N	available in normal mode
		S	available in service mode
		C	available in calibration mode (not for users)
		F	available in factory mode (not for users)
		All	available in all modes

Inputs can be read in all modes but some coils can't be written in normal mode or service mode.

Overview coils

DEVICE CONTROL					
COIL	PARAMETER	R/W	DEF.	MODE	DESCRIPTION
10	IO_CLEAR_ERROR	R/W	0	ALL	SELECT NORMAL OPERATION AND CLEAR ERROR (1= CLEAR ERROR)
11 - 17	FACTORY USE ONLY				
18	IO_RESTART_MODBUS	R/W	0	ALL	RESTART THE DEVICE WITH MODBUS PROTOCOL
19	FACTORY USE ONLY				
20	IO_ROUNDOFF	R/W	1	S,N	ENABLE ROUNDING OF SENSOR DATA
21	IO_AUTO_RANGE	R/W	0	S,N	ENABLE AUTO RANGE MODE (0=NO AUTO RANGE)
22	IO_FASTRESPONSE	R/W	0	S,N	ENABLE FAST RESPONSE FILTER (0=NO FILTER)
23	IO_TRACKING_FILTER	R/W	1	S,N	ENABLE TRACKING FILTER (0=NO FILTER)

Note: The default value of the device options are stored non-volatile memory. The default values can be overruled during the operation. However, at power-on the default values are restored and the smart sensor will start up with the default values stored in the non-volatile memory.

ADC CONTROL					
COIL	PARAMETER	R/W	DEF.	MODE	DESCRIPTION
24 - 34	Factory use only				

Read write holding register overview

Register 34 **IO_DEF_SCALE_FACTOR**

The default scale factor is set in the factory mode or service mode and is stored in non-volatile memory. The default scale factor stored in non-volatile memory is always set after a power-on. However it is possible to change the default setting during operation by writing a value to the register 34. Note this value is not stored in non-volatile memory and is overwritten with the default value at power on.

The following values are valid:

Scale factor = 2
Scale factor = 1
Scale factor = 0
Scale factor = -1

Scale factor 0 is the default value. See also input register 4 IO_SCALE_FACTOR.

Read discrete inputs overview

Discrete input 0 IO_FALSE This discrete input is always false
Discrete input 1 IO_TRUE This discrete input is always true

Discrete input 2 **IO_VOID_DATA_FLAG**

The void data flag is raised when the data in register IO_SENSOR1_DATA or IO_RAW_SENSOR1_DATA is not valid, because the body temperature of the sensor is too low or too high, when there is an internal overflow condition, because a calculation is out of range or a division by zero occurred, the reference voltage of the ADC is not stable or the digital filter is not stable. When the IO_VOID_DATA_FLAG is set, bit 0 in the IO_STATUS_FLAGS is also set.

The IO_VOID_DATA_FLAG and bit 0 of the IO_STATUS_FLAGS are cleared when the IO_VOID_DATA_FLAG is read by the computer.

Discrete input 3 IO_OVERFLOW_ERROR

This discrete input is raised when an out of range condition occurs and the sensor data (see IO_SENSOR1_DATA) is above the maximum value specified by the calibration program or above 29,999. The typical maximum value is 4000 W/m².

When the IO_OVERFLOW_ERROR is set, bit 1 in the IO_STATUS_FLAGS is also set.

The IO_OVERFLOW_ERROR and bit 1 of the IO_STATUS_FLAGS are cleared when the IO_OVERFLOW_ERROR is read by the computer.

Discrete input 4 IO_UNDERFLOW_ERROR

This discrete input is raised when an underflow condition occurs and the sensor data (see IO_SENSOR1_DATA) is below the minimum value specified by the calibration program or below -29,999. The typical minimum value is- 400 W/m².

When the IO_UNDERFLOW_ERROR is set, bit 2 in the IO_STATUS_FLAGS is also set.

The IO_UNDERFLOW_ERROR and bit 2 of the IO_STATUS_FLAGS are cleared when the IO_UNDERFLOW_ERROR is read by the computer.

Discrete input 5 IO_ERROR_FLAG

The error flag is raised when there is a (fatal or correctable) hardware error or software error such as: ADC error, DAC error, calibration error or when the update of the calibration data failed. When the IO_ERROR_FLAG is raised the error code is copied to the register IO_ERROR_CODE (see register 26).

The error flag is cleared when a true condition is written to the coil: “IO_CLEAR_ERROR”. This has no effect when the error is fatal or not resolvable such as a calibration error.

The error flag is always set after a power up, this is to indicate the power went off, or a restart occurred. The computer should raise the IO_CLEAR_ERROR in order to reset the error flag.

Discrete input 6 IO_ADC_ERROR

This flag is raised when the A/D converter responsible for the conversion of the analog signals to digital signals detected a failure (hard or software).

The ADC error flag is cleared when a true condition is written to the coil: “IO_CLEAR_ERROR” and the error produced by the ADC, is not fatal.

Discrete input 7 IO_DAC_ERROR

This flag is raised when the D/A converter responsible for the conversion of the digital signal to the analogue output signal detected a failure (hard or software).

The DAC error flag is cleared when a true condition is written to the coil: “IO_CLEAR_ERROR” and the error produced by the DAC, is not fatal.

Discrete input 8 IO_CALIBRATION_ERROR

The calibration error flag is raised when the sensor was not calibrated or a checksum error was detected in the calibration data. This flag can’t be cleared unless the sensor is sent back to the manufacturer or dealer for a re-calibration.

Discrete input 9 IO_UPDATE_FAILED

The update failed is raised when data is written to the non-volatile memory and the update failed. This can happen in calibration mode when calibration data in written to non-volatile memory or in the service mode when device options are written to the non-volatile memory.

If this error is set you should retry the last update action. If the error does not disappear then there could be a hardware problem with the non-volatile memory (EEPROM).

Read write discrete coils overview

Coil 10 IO_CLEAR_ERROR

Setting this coil will clear the error only when the error is a non-fatal error. Reading this coil will always return a 0. The coil IO_CLEAR_ERROR can be used to select the normal mode (see IO_OPERATIONAL_MODE).

The smart sensors will always start-up in the normal mode. Note Use IO_CLEAR_ERROR to return to the normal mode.

Coil 20 IO_ROUNDOFF

Setting this coil enables rounding of the data presented in IO_SENSOR1_DATA and IO_RAW_SENSOR1_DATA.

If not set then the customer should round off the received data before processing the data.

The default value after power on is ON.

If IO_ROUNDOFF is cleared, then the sensor is not calibrated and could produce more digits, than there are significant digits.

Coil 21 IO_AUTO_RANGE

Setting this coil enables the auto-range feature. The auto-range feature increases the number of digits for small signals

The default value after power on is OFF.

If IO_AUTO_RANGE is set then the sensor is not calibrated and could produce more digits, than there are significant digits.

Coil 22 IO_FASTRESPONSE

Setting this coil enables the fast response filter. This filter increases the step response of the sensor. Disabling the fast response give the SMP pyranometers the same response time as the CMP equivalents.

The default value after power on is ON.

Coil 23 IO_TRACKING_FILTER

Setting to this coil enables the tracking filter. The tracking filter reduces the noise of the signal. However, when the filter is on, the step response on a sudden signal change is decreased. The smart sensor uses variable filter constants to minimize the effect on the step response.

The default value after power on is OFF.

Requesting serial number

Register 41 IO_BATCH_NUMBER

The batch number defines the production year of the smart sensor, 11 = 2011, 12=2012 etc.

Register 42 IO_SERIAL_NUMBER

Register 42 defines the 4 digits serial number of the smart sensor. Only the combination of the batch number and serial number is unique.

4.2. Simple demonstration program

The simple “C” program below will show how to read the sensor data and how to deal with errors. The program will read the registers: “operational mode, status flags, scale factor, and sensor data” from Modbus device with address 2 into registers uOperationMode, uStatusFlags, iScaleFactor and iSensorData. Then the program will check the operation mode (must be “normal”) and if there are no errors flags set in iStatusFlags. If there is an error then set the IO_ERROR_FLAG.

```
UInt16    uOperationalMode = 0;
UInt16    uStatusFlags = 0;
Int16     iScaleFactor = 0;
Int16     iSensorData = 0;
float     fSensorData = 0;

int main (void)
{
          while (true)
          {
                    // Send MODBUS request 0x04 Read input registers to slave 2
                    // Get modus data will wait for the answer and copies the data to registers
                    // uOperationalMode, uStatusFlags, iScaleFactor and iSensorData

                    SendModbusRequest (0x04, 2, IO_OPERATIONAL_MODE, 4);
                    WaitModbusReply ();
                    GetModbusData ();

                    If (uOperationalMode != 1)
                    {
                              // Send MODBUS request 0x05 write single coil to slave 2
                              SendModbusRequest (0x05, 2, IO_CLEAR_ERRROR, true);
                              WaitModbusReply ();
                    }
                    else if (uStatusFlags != 0)
```

```
            {
                SendModbusRequest (0x05, 2, IO_CLEAR_ERRROR, true);
                WaitModbusReply ();
            }
            switch (iScaleFactor)
            {
                case 2: fSensorData = (float)(iSensorData) / 100.0;
                case 1: fSensorData = (float)(iSensorData) / 10.0;
                case 0: fSensorData = (float)(iSensorData);
                case -1: fSensorData = (float)(iSensorData) * 10.0;
                default: fSensorData = 0.0;
            }
            // wait 1 second.
            Delay (1000);
        }
    }
```