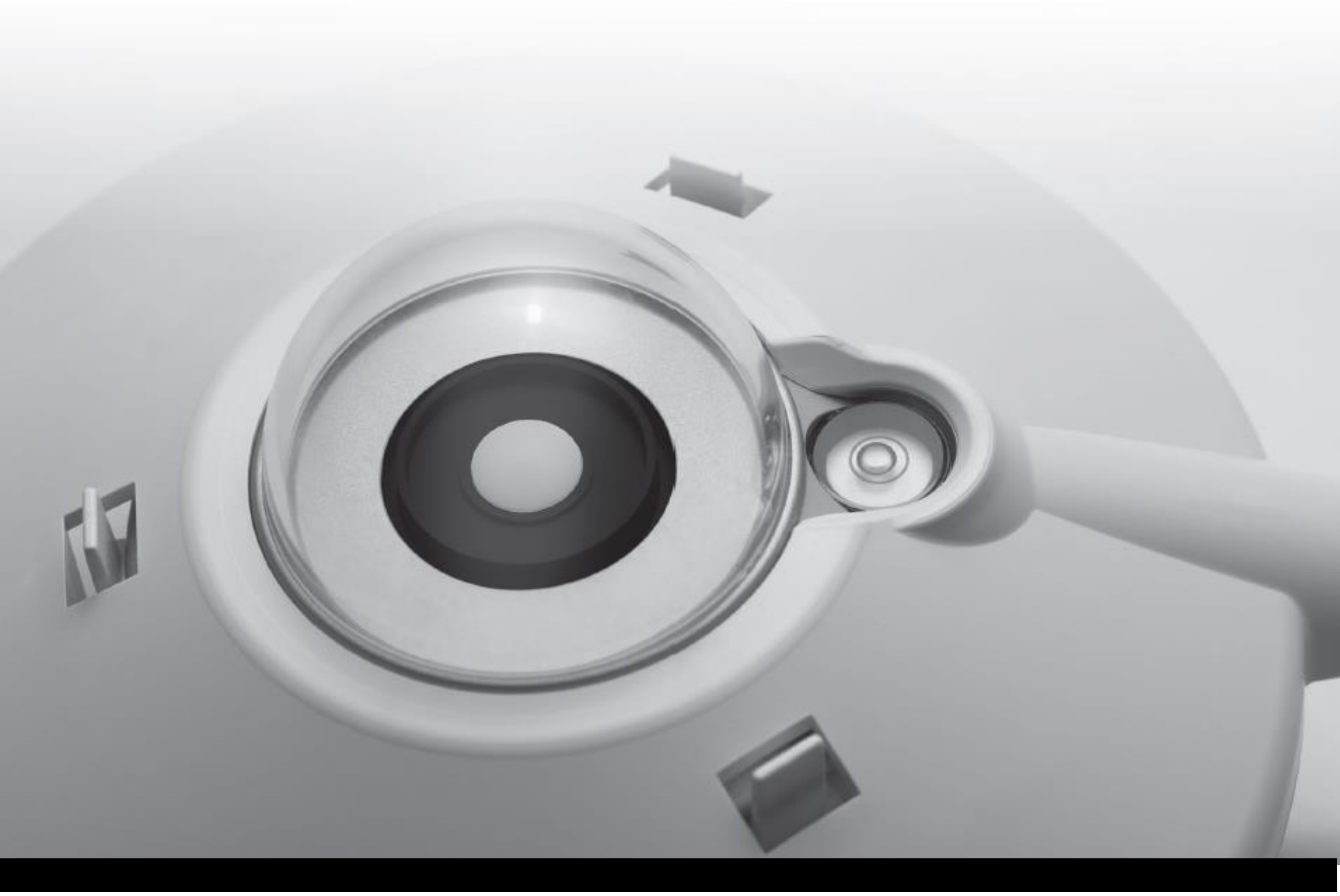
Smart Pyranometer

Modbus® Communication Manual



Modbus®

This document describes the interfacing of Kipp & Zonen Smart Pyranometers (SMP series).

Commonly Used Modbus® Commands

The commands are all 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, such as 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 series devices do not make a difference between a "coil" and a "discrete input". The only difference is that a discrete input is read only. The SMP series devices do not make a difference between a holding register and an input register. The only difference is that an input register is read only.

Input Registers Overview

Input registers are read only.

Real-time Processed Data

PDU ADDRESS	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
2	IO_OPERATIONAL_MODE	R	U16	All	Operational mode: normal, service, calibration, and factory
3	IO_STATUS_FLAGS	R	U16	All	Device Status flags
4	IO_SCALE_FACTOR	R	S16	All	Scale factor for sensor data (determines number of decimal places)
5	IO_SENSOR1_DATA	R	S16	N, S	Temperature compensated radiation in W/m^2 (Net radiation for SGR) (1)
6	IO_RAW_SENSOR1_DATA	R	S16	N, S	Raw, non-linearized and non-temperature compensated radiation(1)
7	IO_STDEV_SENSOR1	R	S16	N, S	Standard deviation IO_SENSOR1_DATA
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
10	IO_SENSOR2_DATA	R	S16	N, S	Temperature compensated long wave down radiation in W/m^2 (only for SGR) (1)
11	IO_RAW_SENSOR2_DATA	R	S16	N, S	Long wave down radiation in W/m^2 (only for SGR) (1)
12	IO_STDEV_SENSOR2	R	S16	N, S	Not used, always 0
13	IO_TEMP_SENSOR_1_K	R	U16	N, S	Body temperature in 0.01 °K (only for SGR)
14	IO_TEMP_SENSOR_2_K	R	U16	N, S	Panel temperature in 0.01 °K (only for DustIQ and RT1)
15	IO_TILT	R	U16	N, S	Tilt of the sensor in the horizontal plane in 0.1° (only for SMP12)
16	IO_RH	R	U16	N, S	Internal relative humidity of the sensor in 0.1% (only for SMP12)
60	IO_DAC_OUTPUT_VOLTAGE	R	U16	N, S	DAC output voltage or current (actual voltage or current)
61	IO_SELECTED_DAC_INPUT	R	U16	N, S	DAC selected input voltage

(1) The scale factor defines the format and number of decimal places

R = Read only | U16 = 16-bit unsigned integer | S16 = 16-bit signed integer | N = available in normal mode | S = available in service mode

Real-time Data A/D Counts

PDU ADDRESS	PARAMETER	R/W	TYPE	MODE	DESCRIPTION
18	IO_ADC1_COUNTS	R	S32	All	Input voltage sensor 1 in 0.01 µV
19					(R18=MSB, R19=LSB)
20	IO_ADC2_COUNTS	R	S32	All	Not supported, always 0
21					
22	IO_ADC3_COUNTS	R	S32	All	Input voltage body temperature sensor in 0.01 µV
23					(R22=MSB, R23=LSB)

PDU ADDRESS	PARAMETER	R/W	TYPE	MODE	DESCRIPTION
24	IO_ADC4_COUNTS	R	S32	All	Input voltage power sensor in 0.01 µV
25					(R24=MSB, R25=LSB)

R = Read only | S32 = 32-bit signed integer | All = available in normal and service mode

Error Reports

PDU ADDRESS	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	Priority 1 error count
29	IO_ERROR_COUNT_PRIO2	R	U16	All	Priority 2 error count
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_TIMEH	R	U16	All	On time in seconds (MSB word)
33	IO_SENSOR_ON_TIMEL	R	U16	All	On time in seconds (LSB word)
41	IO_BATCH_NUMBER	R	U16	All	Production batch number = year in YY
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®) device address RS-485

[2] Writing any value to input registers 26-33 will reset the contents of the registers

R = Read only | U16 = 16-bit unsigned integer | All = available in normal and service mode

Real Time Floating Data Points

PDU ADDRESS	PARAMETER	R/W	TYPE	MODE	DESCRIPTION
10000	U_DEVICE_TYPE	R	U16	All	Device type of the sensor (see register IO_DEVICE_TYPE)
10001	U_OPERATIONAL_MODE	R	U16	All	Operational mode (see register IO_OPERATIONAL_MODE)
10002	U_ERROR_CODE	R	U16	All	Most recent/ actual error code (see register IO_ERROR_CODE)
10003	U_STATUS_FLAGS	R	U16	All	Device Status flags (see register U_STATUS_FLAGS)
10004	U_BATCH_NR	R	U16	All	Production Batch number (see register IO_BATCH_NUMBER)
10005	U_SERIAL_NR	R	U16	All	Serial number (see register IO_SERIAL_NUMBER)
10006	FL_SENSOR1_DATA	R	F32	All	Temperature compensated radiation 1 in W/m^2 with decimal point set by scale factor
10008	FL_STDEV_SENSOR1	R	F32	All	Standard deviation Sensor 1 with decimal point
10010	FL_ SENSOR2_DATA	R	F32	All	Temperature compensated radiation Sensor 2 or Long wave down with decimal point.
10012	FL_STDEV_SENSOR2	R	F32	All	Not used. Always 0
10014	FL_BODY_TEMPERATURE	R	F32	All	Body temperature in ° Kelvin with decimal point.
10016	FL_EXT_POWER_SENSOR	R	F32	All	External power voltage with decimal point
10018	F_PVPANEL_TEMP_K	R	F32	All	PV panel temperature in ° Kelvin
10020	FL_TILT	R	F32	All	Tilt of the sensor in the horizontal plane in °
10022	FL_RH	R	F32	All	Internal relative humidity of the sensor in %

R = Read only | U16 = 16-bit unsigned integer | F32 = 32-bit floating point | All = available in normal and service mode

Real Time Floating Data Points

PDU ADDRESS	PARAMETER	R/W	TYPE	MODE	DESCRIPTION
20000	FL_TILT_ROLL[1]	R	F32	All	Roll of the sensor in °
20002	FL_TILT_PITCH[1]	R	F32	All	Pitch of the sensor in °

[1] Registers only available for SMP12 pyranometer

Legend

PDU Address	PDU Address + 1 = Modbus® register number		
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
		F32	32-bit floating point
Mode	Operation mode	N	Available in normal mode
		S	Available in service mode
		ALL	Available in all modes

Holding Registers Overview

PDU ADDRESS	PARAMETER	R/W	TYPE	MODE	DESCRIPTION
34	IO_DEF_SCALE_FACTOR	R/W	S16	All	Default scale factor
35 ~ 40	Factory use only				

R/W = Read/Write | S16 = 16-bit signed integer | All = available in normal and service mode

Discrete Inputs Overview

Status Indicators

INPUT	PARAMETER	R/W	DEFAULT	MODE	DESCRIPTION
0	IO_SENSOR1_DISCONNECTED	R	0	All	Sensor 1 disconnected
1	IO_SENSOR2_DISCONNECTED	R	0	All	Sensor 2 disconnected
2	IO_VOID_DATA_FLAG	R	0	All	Void signal, 1=unstable signal, temperature too low or too high
3	IO_OVERFLOW_ERROR	R	1[1]	All	Overflow, signal out of range
4	IO_UNDEFLOW_ERROR	R	1[1]	All	Underflow signal out of range
5	IO_ERROR_FLAG	R	1[1]	All	General hardware error (set if one of the H/W error flags is set)
6	IO_ADC_ERROR	R	1[1]	All	Hardware error A/D converter
7	IO_DAC_ERROR	R	1[1]	All	Hardware error D/A converter
8	IO_CALIBRATION_ERROR	R	1[1]	All	Calibration checksum error
9	IO_UPDATE_FAILED	R	1[1]	All	Update parameters stored in nonvolatile memory failed

A discrete input can be true or false. A discrete input is read only and can be read in all modes.

[2] Set if an error occurred at power on, otherwise cleared.

Discrete Coils Overview

Device Control

COIL	PARAMETER	R/W	DEFAULT	MODE	DESCRIPTION
10	IO_CLEAR_ERROR	R/W	0	All	Select normal operation and clear error (1=clear error)
11 TO 17	Factory use only				
18	IO_RESTART_MODBUS	R/W	0	All	Restart the device with Modbus® protocol
19	Factory use only				
20	IO_ROUND	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)

R/W = Read/Write | N = available in normal mode | S = available in service mode | All = available in normal and service mode

A coil can be read, but some can't be written in normal mode or service mode.

Note: The default values of the device options are stored in non-volatile memory. The default values can be overruled during 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.

Legend

Input	PDU Address + 1 = Modbus® register number		
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 register	
R/W	Read Write	R	Read only
		R/W	Read/write
Def	Default Value		Default value at power on (0,1, or undefined)
Mode	Operation mode	N	Available in normal mode
		S	Available in service mode
		ALL	Available in all modes

Inputs can be read in all modes but some coils cannot be written to in normal or service mode

Input Register Details

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

REGISTER	PARAMETER	DESCRIPTION
0	IO_DEVICE_TYPE	The device type defines which device is connected. This register can be used to check the type of the connected device. IO_datamodel_version 107 supports the following type of sensors:

Sensor Type	*Value*	*# of sensors*
SMP3 (volt version)	601	1
SMP3 (current loop version)	602	1
SMP6 (volt version)	619	1
SMP6 (current version)	620	1
SMP10 (volt version)	617	1

SMP10 (current version)	618	1
SMP11 (volt version)	603	1
SMP11 (current loop version)	604	1
SMP12 (volt version)	633	1
SMP21 (volt version)	605	1
SMP21 (current loop version)	606	1
SMP22 (volt version)	607	1
SMP22 (current loop version)	608	1
SGR3 (volt version)	609	2*
SGR3 (current loop version)	610	2*
SGR4 (volt version)	611	2*
SGR4 (current loop version)	612	2*
SHP1 (volt version)	613	1
SHP1 (current loop version)	614	1
SUV5 (volt version)	615	1
SUV5 (current loop version)	616	1

1 IO_DATAMODEL_VERSION

The data-model describes the functions supported by the smart sensor. This document is valid for data-model version: 107. 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 >=107. If you receive another value, then you should read an older or newer version of this document and check the differences.

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
5	Error mode

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

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 #	*Individual bit representation*	*Remark*
0	Quality of the signal	see IO_VOID_DATA_FLAG
1	Overflow	see IO_OVERFLOW_ERROR
2	Underflow	see IO_UNDERFLOW_ERROR
3	Error flag	see IO_ERROR_FLAG
4	ADC Error	see IO_ADC_ERROR
5	DAC Error	see IO_DAC_ERROR
6	Calibration Error	see IO_CALIBRATION_ERROR
7	Update EEPROM error	see IO_UPDATE_FAILED
8	Power failure error	see POWER_FAILED_FLAG
9	Tilt sensor error	see IO_TILT_ERROR
10	Relative humidity sensor error	see IO_RH_ERROR
11	Relative humidity threshold warning	see IO_RH_THRESHOLD
12	Body temperature sensor error	see IO_BODY_TEMP_ERROR

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 a copy of register 34 IO_DEF_SCALE_FACTOR, made during power up.
If the register IO_SCALE_FACTOR is not set to 0 then you must multiply or divide the data of the above mentioned four IO_SENSOR registers.

Scale Factor	*Calculation*
2	floating point result = integer register X / 100
1	floating point result = integer register X / 10
0	floating point result = integer register X
-1	floating point result = integer register X * 10

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).

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.

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	PARAMETER	DESCRIPTION
7	IO_STDEV_SENSOR1	This register 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*
8	IO_BODY_TEMPERATURE	The body temperature sensor measures the temperature of the body in 0.1°C. To convert the data to a floating-point number, make the following calculation: *floating point result = integer register (IO_BODY_TEMPERATURE) / 10*
9	IO_EXT_POWER_SENSOR	The external power sensor measures the external voltage applied to the chassis socket in 0.1 Volt. To convert the data to a floating-point number, make the following calculation: *floating point result = integer register (IO_EXT_POWER_SENSOR) / 10*
15	IO_TILT	The tilt sensor measures the tilt of the sensor in the horizontal plane in 0.1°. To convert the data to a floating-point number, make the following calculation: *floating point result = integer register (IO_TILT) / 10*
16	IO_RH	The RH sensor measures the Internal relative humidity of the sensor in 0.1%. To convert the data to a floating-point number, make the following calculation: *floating point result = integer register (IO_RH) / 10*

Holding Register Details

REGISTER	PARAMETER	DESCRIPTION
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

Discrete Inputs Details

INPUT	PARAMETER	DESCRIPTION
0	IO_SENSOR1_DISCONNECTED	0 = true, 1 = false
1	IO_SENSOR2_DISCONNECTED	0 = true, 1 = false
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.
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.
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.
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

INPUT	PARAMETER	DESCRIPTION
		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 to reset the error flag.
6	IO_ADC_ERROR	This flag is raised when the A/D converter responsible for the conversion of the analogue 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.
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.
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.
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).

Discrete Coils Details

COIL	PARAMETER	DESCRIPTION
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.
18	IO_RESTART_MODBUS	
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.
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.
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.
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	PARAMETER	DESCRIPTION
41	IO_BATCH_NUMBER	The batch number defines the production year of the smart sensor, 20=2020, 21=2021 etc.
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.

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);
            }
}
```

OTT HydroMet B.V. | Delftechpark 36 | 2628 XH Delft | The Netherlands | +31 15 2755 210 | solar.info@otthydromet.com | www.otthydromet.com

Document Number: 0426210 | 01-0324

Find the extended product manuals for all our Smart Pyranometers on OTTHydromet.com.