SOLUZIONE SOLARE

# Modbus Configuration

## MODBUS PROTOCOL

Modbus is a Master-Slave protocol that is widely used as an industry standard. It is simple, efficient and reliable. It can be easily used to access and collect data or exchange information between digital systems over a serial line local bus (and with its TCP/IP extension through a LAN or World Wide Web).
Please refer to specific detailed documentation and implementations freely available at www.modbus.org

SM PRO is a Modbus RTU slave that implements the following standard access functions:

| Function code | Description |
|---|---|
| **0x03** | READ HOLDING REGISTERS |
| **0x04** | READ INPUT REGISTERS |
| **0x06** | WRITE SINGLE REGISTER |
| **0x10** | WRITE MULTIPLE REGISTERS |

Tab. 2

Please note that in the current implementation of SM PRO function codes 0x03 and 0x04 are equivalent and address the same data area.

Data is accessible through Modbus's functions by 16 bits units called "registers". In the current implementation of SM PRO these registers are available:

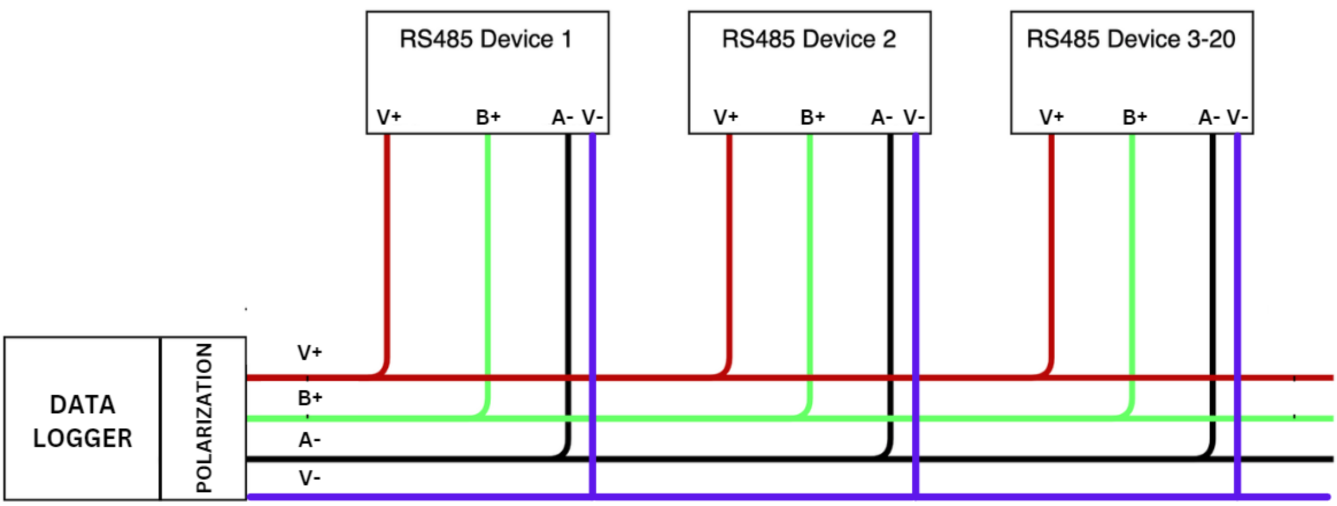
| Register # | Description | Access | NV save |
|---|---|---|---|
| **0x0101 or 0x0201** | **Current irradiance level** [W/m$^2$], | R | |
| **0x0102** | **Current PT100 temperature** [°C], 2-complement value, fixed point 14.2 format (14 bits integer, 2 bits fractional) | R | |
| **0x0202** | **Current PT100 temperature** [°C], format multipl. by 10 (to get value in °C divide by 10) | R | |
| **0x0103** | **Status**, bit coded<br>**Bit** – **Description**<br>0 – Factory calibration/configuration<br>1 = OK; 0 = need recalibration<br>1 – Not volatile parameters<br>1 = OK; 0 = default loaded, need to be changed/saved<br>2 – Digital input monitor<br>1 = not active (open); 0 = active (shorted to GND)<br>3 – PT100 RTD element<br>1 = OK; 0 = shorted or open circuit (not present/malfunctioning)<br>4 – Analog output<br>1 = OK; 0 = output current can't flow at desired level due to wire break/high load impedance/output voltage approaching positive supply<br>5 – Watchdog<br>1 = reset by watchdog timeout occurred; 0 = normal operation<br>all undefined bits read as 0 | R | |
| **0x8001** | **Serial number**, least significant word | R | |
| **0x8002** | **Serial number**, most significant word | R | |
| **0x8003** | **Firmware main version**, hexadecimal | R | |
| **0x8004** | **Firmware minor version**, hexadecimal | R | |
| **0x8005** | **Node address**, range 1 ÷ 247, decimal, **default 1** | R/W | Y |
| **0x8006** | **Bitrate**, coded, range 0 ÷ 4, decimal, **default 1**<br>0 – 9600 bps<br>1 – 19200 bps<br>2 – 38400 bps<br>3 – 57600 bps | R/W | Y |

| | | | |
|---|---|---|---|
| | 4 – 115200 bps | | |
| **0x8007** | **Serial configuration**, coded, range 0 ÷ 3, decimal, **default 0**<br>0 – 8N1 (8 bit / no parity / 1 stop bit)<br>1 – 8E1 (8 bit / even parity / 1 stop bit)<br>2 – 8O1 (8 bit / odd parity / 1 stop bit)<br>3 – 8N2 (8 bit / no parity / 2 stop bit) | R/W | Y |
| **0x8008** | **Serial reply delay** [ms], range 0 ÷ 100, decimal, **default 1** | R/W | Y |
| **0x8009** | **Analog output mode**, coded, range 0 ÷ 4, decimal, **default 2**<br>0 – output disabled<br>1 – 0 ÷ 10 V<br>2 – 0 ÷ 5 V<br>3 – 0 ÷ 20 mA current loop<br>4 – 4 ÷ 20 mA current loop | R/W | Y |
| **0x800A** | **Analog output select**, coded, range 0 ÷ 3, decimal, **default 2**<br>0 – irradiance<br>1 – PT100 temperature<br>2 – selected by digital input status: open = irradiance; close = PT100 temp.<br>3 – value setted by register 0x8201 | R/W | Y |
| **0x800B** | **PT100 RTD reading enable**, coded, range 0 ÷ 1, decimal, **default 1**<br>0 – disabled<br>1 – enabled | R/W | Y |
| **0x8101** | **Not volatile params save command**, write 1 to execute (then wait 1 s before to send next message) | W | |
| **0x8102** | **Software reset command**, write 1 to execute (then wait 6 s before to send next message) | W | |
| **0x8201** | **Analog output level** [], range 0 ÷ 65535, decimal, fixed point 0.16 format (16 bits fractional) | W | |

Please note that, conventionally, Modbus register's numbering starts from 1 but register's addressing start from 0 so, to obtain the register's address you had simply to subtract 1 from its number. That's meaningful depending on, as a master, you are using a high level Modbus utility/program (that normally refers to the registers' number) or a low level driver (that normally directly works with addresses).

## MODBUS – MORE INFORMATION

1. For troubleshooting and information about MODBUS protocol kindly visit https://modbus.org/ even for recommendations on polarization. Kindly check the general schema of the MODBUS RS485 protocol given below.



2. In case if you have a bad communication and in order to have a right polarization, follow this formula for calculating the voltages B+ and A- for finding out if the connections satisfy the condition

$$\mathbf{V_{B+} - V_{A-} \geq 0.2\ V}$$

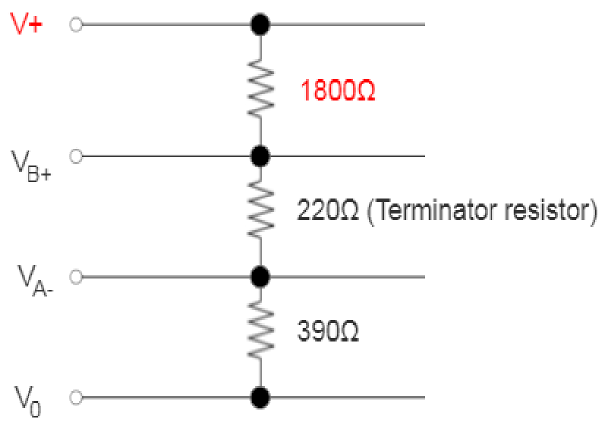
Voltage in B+ is equal to $\frac{(R_2 + R_3)}{(R_1 + R_2 + R_3)} * (V+)$

Voltage in A- is equal to $\frac{(R_3)}{(R_1 + R_2 + R_3)} * (V+)$

Where $R_1$, $R_2$ and $R_3$ are the three resistors.

V+ is the supply voltage

**Example for 12V**




**Example for 24V**