Soil Moisture & Temperature & EC Sensor
User Manual

Product Model: S-Temp&VWC&EC-02
Version: V1.1
## Contents

1 Introduction .................................................................................................................................................. 3  
2 Specifications .............................................................................................................................................. 4  
3 Wiring ....................................................................................................................................................... 5  
4 Installation ................................................................................................................................................ 6  
5 RS485 Modbus Protocol ............................................................................................................................ 7  
   5.1 Modbus Protocol ............................................................................................................................... 7  
   5.2 Modbus Register ............................................................................................................................... 8  
   5.3 Detail of Modbus Register ................................................................................................................ 10  
   5.4 Communication Sample ..................................................................................................................... 15  
      5.4.1 Function number 3 communication sample ............................................................................. 15  
      5.4.2 Function number 4 communication sample ............................................................................. 16  
      5.4.3 Function number 6 communication sample ............................................................................. 17  
      5.4.4 Function number 16 communication sample .......................................................................... 18  
   5.5 CRC16 Check Algorithm ..................................................................................................................... 21  
6 Document version ..................................................................................................................................... 24
1 Introduction

This S-Temp&VWC&EC-02 soil moisture & EC & temperature sensor is provided with high accurate and high sensitive. It is an important tool to observe and study the occurrence, evolution, improvement and the dynamics water of saline soil. By measuring the dielectric constant of the reaction of soil, soil direct stable real moisture content. This S-Temp&VWC&EC-02 sensor can measure the volume of soil moisture. The soil moisture measurement method is in line with international standards at present. Apply to the soil moisture monitoring, scientific experiment, water-saving irrigation, greenhouse vegetables, flowers, grass, soil, plant cultivation, measured speed of sewage treatment, grain storage, greenhouse control, precision agriculture.

Features:
- Soil moisture content, electrical conductivity and temperature all in one
- One solution can also be used for fertilizer, and other nutrient solution conductivity matrix
- Electrode using special treatment of the alloy material, can withstand a strong external impact, not easy to damage
- Completely sealed, acid and alkali corrosion, can be buried in the soil or directly into the water for long-term dynamic testing
- High precision, fast response, good compatibility, the probe insert design to ensure accurate measurement, reliable performance
- Perfect protection circuit
# 2 Specifications

## Soil Temperature

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>-40 °C ~ 80 °C</td>
</tr>
<tr>
<td>Accuracy</td>
<td>± 0.5°C</td>
</tr>
<tr>
<td>Resolution</td>
<td>0.1 °C</td>
</tr>
</tbody>
</table>

## Soil Moisture

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>From completely dry to fully saturated (0% - 100%)</td>
</tr>
<tr>
<td>Accuracy</td>
<td>± 3% (0<del>50%); ± 5% (50</del>100%)</td>
</tr>
<tr>
<td>Resolution</td>
<td>0.03% (0<del>50%); 1% (50</del>100%)</td>
</tr>
</tbody>
</table>

## Electrical Conductivity

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>0 ~ 10000 μs/cm</td>
</tr>
<tr>
<td>Accuracy</td>
<td>± 3%</td>
</tr>
<tr>
<td>Resolution</td>
<td>10 μs/cm</td>
</tr>
</tbody>
</table>

## Temperature Compensation

- Built in temperature compensation sensor, range 0-50°C

## General Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Model</td>
<td>S-Temp&amp;VWC&amp;EC-02</td>
</tr>
<tr>
<td>Interface</td>
<td>RS-485</td>
</tr>
<tr>
<td>Protocol</td>
<td>MODBUS-RTU RS485</td>
</tr>
<tr>
<td>Power Supply</td>
<td>3.6 ~ 30V DC</td>
</tr>
<tr>
<td>Current Consumption</td>
<td>6mA@24V DC (quiescent dissipation)</td>
</tr>
<tr>
<td>IP Rating</td>
<td>IP68</td>
</tr>
<tr>
<td>Cable Length</td>
<td>5 meters</td>
</tr>
<tr>
<td>Operating Temperature</td>
<td>-40 ~ 85°C</td>
</tr>
<tr>
<td>The material of the probe</td>
<td>Anti-corrosion special electrode</td>
</tr>
<tr>
<td>Sealing material</td>
<td>The black flame retardant epoxy resin</td>
</tr>
<tr>
<td>Installation</td>
<td>All embedded or probe inserted into the measured medium</td>
</tr>
<tr>
<td>Device Weight</td>
<td>210g</td>
</tr>
</tbody>
</table>
3 Wiring

<table>
<thead>
<tr>
<th>Color</th>
<th>Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yellow</td>
<td>RS485+/A/T+</td>
</tr>
<tr>
<td>White</td>
<td>RS485-/B/T-</td>
</tr>
<tr>
<td>Red</td>
<td>VCC+, power supply</td>
</tr>
<tr>
<td>Black</td>
<td>VCC-, power ground</td>
</tr>
<tr>
<td>Green (SET)</td>
<td>SETTING mode. When sensor power-up with the SET wire connected to Power Supply +, then sensor using setting mode communication parameters for RS485. When sensor power-up with the SET wire connected to Power Supply – or unconnected, the sensor using communication parameters in register for RS485. Please refer to the usage below. All RS485 communication parameters (Mosbus Slave Address, baudrate, parity, databits, stopbits) are set in internal register and can be saved when power down, the factory setting is ADDRESS=1, BAUDRATE=9600bps, PARITY=NONE, DATABITS=8bits, STOPBITS=1bit; Sometimes you may FORGET the communication settings, in this case, you can connect the GREEN &amp; RED wire together to PowerSupply+, black wire to PowerSupply-, then re-power up the sensor, then the sensor start-up with a fixed communication settings (we call it setting mode) ADDRESS=0, BAUDRATE=9600bps, PARITY=NONE, DATABITS=8bits, STOPBITS=1bit; Communicate with the sensor using this parameters and then set your desired communication parameters, then disconnect the green wire from PowerSupply+, then re-power up the sensor, and the sensor will communicate with your settings.</td>
</tr>
</tbody>
</table>
4 Installation

Surface installation
- Clear away any stones. Pre-form holes in very hard soils before insertion.
- Push the sensor into the soil until the rods are fully inserted. Ensure good soil contact.
- If you feel strong resistance when inserting the sensor, you have probably hit a stone. Stop, and re-insert at a new location.

Note: The sensor is suitable for soil surface temperature measurements.

Installing at depth
- Make a 45mm diameter hole, preferably at about 10° to the vertical using a auger.
- Push the sensor into the soil until rods are fully inserted. Ensure good soil contact.
- Fill and repack the hole with soil.

Alternatively
- Dig a trench, and install horizontally.

Because of the direct determination of the soluble salt ions in the soil, the water content of the soil can be higher than about 20%, and the soluble ions in the soil can correctly reflect the electrical conductivity of the soil. In the long-term observation, after irrigation or rainfall measured values are close to the true level. If the velocity measurement, first in the tested soil watering, to be full of water permeability were measured.

(1) Rapid measurement method: selected measurement locations, avoid the rocks, to ensure that the needle will not touch the stones like hard object, according to the required depth of cut open the surface soil, maintain the tightness degree of the original soil below the sensor body, clenched vertically inserted into the soil, can not be inserted before and after shaking, ensure the close contact with the soil. A measuring point within a small range test should repeatedly averaging.

(2) Buried in the underground measurement method: vertical drilling diameter greater than 20 cm depth of pit, according to the measurement needs, then the sensor wire inserted into the pit wall in a given level of depth, the pit landfill compaction, ensure the close contact with the soil. Stable after a period of time, can be last for days, months or even longer to measure and record.

If the surface measurement is hard, should first hole (diameter should be less than the diameter of the probe), and then inserted into the soil and the soil compaction and measurement; sensor should prevent violent vibration and impact, but not with a hard object percussion. Because the sensor for black package, in the strong sunlight will make the sensor to make sharp warming (up to over 50 °C), in order to prevent the temperature measurement of high temperature impact sensor, please pay attention to sun protection in the field or fields.
5 RS485 Modbus Protocol

5.1 Modbus Protocol

Modbus Protocol is widely used to establish master-slave communication between intelligent devices or sensors. A MODBUS message sent from a master to a slave contains the address of the slave, the function code (e.g. 'read register' or 'write register'), the data, and a check sum (LRC or CRC).

S-Temp&VWC&EC-02 sensor with RS485 interface, support Modbus protocol. The communication parameters to factory default values for: baud rate 9600 bps, one start bit, 8 data bits, no parity, one stop bit. Communication protocol is Modbus RTU protocol. Communication parameters can be changed by the setup program or MODBUS command, after the communication parameters are changed, the sensor is required to re-enter the sensor to be effective.

Following modbus function code are supported by sensor.

- Modbus Function Code 0x03: used for reading holding register.
- Modbus Function Code 0x04: used for reading input register.
- Modbus Function Code 0x06: used for writing single holding register.
- Modbus Function Code 0x10: used for writing multiple holding register.
### 5.2 Modbus Register

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Register address (HEX / DEC)</th>
<th>Parameter type</th>
<th>Modbus function number</th>
<th>Parameter range and description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEMPERATURE</td>
<td>0x0000 /0</td>
<td>INT16, read</td>
<td>3/4</td>
<td>-4000-8000 corresponds to -40.00 ~ 80.00°C.</td>
</tr>
<tr>
<td>VWC - Volumetric Water Content</td>
<td>0x0001 /1</td>
<td>UINT16, read</td>
<td>3/4</td>
<td>0-10000 corresponds to 0-100%</td>
</tr>
<tr>
<td>EC - Electrical Conductivity</td>
<td>0x0002 /2</td>
<td>UINT16, read</td>
<td>3/4</td>
<td>0-20000 corresponds to 0-20000us/cm</td>
</tr>
<tr>
<td>SALINITY</td>
<td>0x0003 /3</td>
<td>UINT16, read</td>
<td>3/4</td>
<td>0-20000 corresponds to 0-20000mg/L</td>
</tr>
<tr>
<td>TDS</td>
<td>0x0004 /4</td>
<td>UINT16, read</td>
<td>3/4</td>
<td>0-20000 corresponds to 0-20000mg/L</td>
</tr>
<tr>
<td>EPSILON</td>
<td>0x0005 /5</td>
<td>UINT16, read</td>
<td>3/4</td>
<td>0-8200 corresponds to 0.00~82.00</td>
</tr>
<tr>
<td>SOIL TYPE</td>
<td>0x0020 /32</td>
<td>UINT16, read-write</td>
<td>3/6/16</td>
<td>0-3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0: Mineral soil</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1: Sandy soil</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2: Clay</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3: Organic soil</td>
</tr>
<tr>
<td>TEMP UNIT</td>
<td>0x0021 /33</td>
<td>UINT16, read-write</td>
<td>3/6/16</td>
<td>0: °C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1: °F</td>
</tr>
<tr>
<td>EC &amp; TEMP COFF</td>
<td>0x0022 /34</td>
<td>UINT16, read-write</td>
<td>3/6/16</td>
<td>0-100 corresponds to 0.00%-10.0%</td>
</tr>
<tr>
<td>SALINITY COFF</td>
<td>0x0023 /35</td>
<td>UINT16, read-write</td>
<td>3/6/16</td>
<td>0-100 corresponds to 0.00-1.00</td>
</tr>
<tr>
<td>TDS COFF</td>
<td>0x0024 /36</td>
<td>UINT16, read-write</td>
<td>3/6/16</td>
<td>0-100 corresponds to 0.00-1.00</td>
</tr>
<tr>
<td>Slave ADDRESS</td>
<td>0x0200 /512</td>
<td>UINT16, read-write</td>
<td>3/6/16</td>
<td>0-255</td>
</tr>
<tr>
<td>BAUDRATE</td>
<td>0x0201 /513</td>
<td>UINT16, read-write</td>
<td>3/6/16</td>
<td>0-6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0: 1200bps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1: 2400bps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2: 4800bps</td>
</tr>
<tr>
<td></td>
<td>Value</td>
<td>Type</td>
<td>Access</td>
<td>Value Range</td>
</tr>
<tr>
<td>----------------</td>
<td>----------</td>
<td>-----------------</td>
<td>--------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| PROTOCOL       | 0x0202 /514 | UINT16, read-write | 3/6/16 | 0~1         | 0: Modbus RTU  
|                |          |                 |        |             | 1: Modbus ASCII                                                             |
| PARITY         | 0x0203 /515 | UINT16, read-write | 3/6/16 | 0~2         | 0: No parity bit  
|                |          |                 |        |             | 1: even parity check  
|                |          |                 |        |             | 2: Odd Parity bit                                                          |
| DATABITS       | 0x0204 /516 | UINT16, read-write | 3/6/16 | 1           | 1:8 data bits                                                              |
| STOPBITS       | 0x0205 /517 | UINT16, read-write | 3/6/16 | 0~2         | 0:1 Stop bit  
|                |          |                 |        |             | 1:2 Stop bit                                                               |
| RESPONSE DELAY | 0x0206 /518 | UINT16, read-write | 3/6/16 | 0~255       | 0-255 corresponds to the 0-2550 milliseconds sensor to receive the host request for a period of time and then the delay response. The time delay for setting the value of *10 milliseconds. Set to 0 when no delay. |
| ACTIVE OUTPUT INTERVAL | 0x0207 /519 | UINT16, read-write | 3/6/16 | 0~255       | 0-255 corresponds to 0-255 seconds does not require the host to request, the sensor to send data at a fixed time interval. The time interval is set value * 1 second. Set to 0 when the active output function is prohibited. |
5.3 Detail of Modbus Register

**TEMPERATURE**

<table>
<thead>
<tr>
<th>Parameter range</th>
<th>-4000-8000 corresponds to -40.00~80.00 ℃</th>
<th>Default: none</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter storage</td>
<td>none</td>
<td></td>
</tr>
</tbody>
</table>

Meaning: the measured value of the temperature, negative for complement representation.

For example: if the return value is 0702H (Hexadecimal, source code), the first byte is 07H, the second byte and the low byte is 02H, then the temperature for the measured value (07H*256+02H) /100=17.94 ℃.

If the return value is FF05H (Hex, the complement), the first byte is FFH, low second byte is 05H, then temperature measurement value ((FFH*256+05H) -FFFFH-1H) / 100 = (FF05H-FFFFH-1H) Celsius /100=-2.51 ℃.

**VWC --- volumetric water content**

<table>
<thead>
<tr>
<th>Parameter range</th>
<th>0-10000 corresponds to 0-100%</th>
<th>Default: none</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter storage</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

Significance: volumetric water content measurements.

For example: if the return value is 071DH (Hexadecimal), the first byte of the high byte is 07H, second bytes of low byte is 1DH, then the measured value is (07H*256+1DH) /100= (7*256+29)/100 =18.21. representative volume water content is 18.21%.

**EC --- electrical conductivity**

<table>
<thead>
<tr>
<th>Parameter range</th>
<th>0-20000 corresponds to 0-20000us/cm</th>
<th>Default: none</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter storage</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

Significance: electrical conductivity measurement.

For example: if the return value is 071DH (Hexadecimal), the first byte is 07H, the second byte and the low byte is 1DH, then conductivity measurement value (07H*256+1DH) = (7*256+29) =1821.

representative soil conductivity is 1821us/cm

**SALINITY**

<table>
<thead>
<tr>
<th>Parameter range</th>
<th>0-20000 corresponds to 0-20000mg/L</th>
<th>Default: none</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Storage</td>
<td>Significance</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>Parameter</td>
<td>None</td>
<td>Salinity Measurement. For example: if the value returned is 071DH (Hexadecimal), the first byte of the high byte is 07H, the second byte low byte is 1DH, then the salinity measurement value (07H<em>256+1DH) = (7</em>256+29) =1821. representative the soil salinity is 1821mg/L.</td>
</tr>
</tbody>
</table>

**TDS--- total dissolved solids**

<table>
<thead>
<tr>
<th>Parameter range</th>
<th>0-20000 corresponds to 0-20000mg/L</th>
<th>Default: none</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter storage</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td><strong>Significance:</strong> TDS measurement value. For example: if the value returned is 071DH (Hexadecimal), the first byte of the high byte is 07H, second bytes of low byte is 1DH, then the TDS measurement value (1DH+07H<em>256)= (7</em>256+29) =1821. representative TDS is 1821mg/L.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**EPSILON--- dielectric constant**

<table>
<thead>
<tr>
<th>Parameter range</th>
<th>0-8200 corresponds to 0.00-82.00</th>
<th>Default: none</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter storage</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td><strong>Meaning:</strong> dielectric constant. For example: if the value returned is 071DH (Hexadecimal), the first byte is 07H, the second byte low byte is 1DH, then the measured value is (1DH+07H<em>256) /100= (7</em>256+29)/100 =18.21. to represent the dielectric constant of 18.21.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TEMP UNIT--- degree unit**

| Parameter range | 0: ºC  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter storage</td>
<td>Immediate storage</td>
</tr>
<tr>
<td><strong>Significance:</strong> unit of temperature.</td>
<td></td>
</tr>
</tbody>
</table>

**EC TEMP COFF**

| Parameter range | 0-100 corresponds to 0.0%-10.0% | Default: 20 (2%)
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter storage</td>
<td>Immediate storage</td>
<td></td>
</tr>
<tr>
<td><strong>Significance:</strong> the temperature compensation coefficient of electrical conductivity</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### SALINITY COFF

<table>
<thead>
<tr>
<th>Parameter range</th>
<th>0-100 corresponds to 0.00-1.00</th>
<th>Default: 55 (0.55)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter storage</td>
<td>Immediate storage</td>
<td></td>
</tr>
<tr>
<td>Significance:</td>
<td>Salinity / conductivity compensation coefficient</td>
<td></td>
</tr>
</tbody>
</table>

### TDS COFF

<table>
<thead>
<tr>
<th>Parameter range</th>
<th>0-100 corresponds to 0.00-1.00</th>
<th>Default: 50 (0.50)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter storage</td>
<td>Immediate storage</td>
<td></td>
</tr>
<tr>
<td>Significance:</td>
<td>TDS/ conductivity compensation coefficient</td>
<td></td>
</tr>
</tbody>
</table>

### MEASURE METHOD --- Measure Method

<table>
<thead>
<tr>
<th>Data Range</th>
<th>Default: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0: Continuous</td>
<td></td>
</tr>
<tr>
<td>1: Request</td>
<td></td>
</tr>
<tr>
<td>Power Down Save</td>
<td>Immediate storage</td>
</tr>
</tbody>
</table>

Note: In Continuous mode, the sensor always convert the VWC, EC and Temperature and update the internal data and ready for reading. In Request mode, the sensor convert the VWC, EC and Temperature data by receiving the data request command from master device.

### SLAVE ADDR --- Modbus address

<table>
<thead>
<tr>
<th>Parameter range</th>
<th>0-255</th>
<th>Default: 1 or 18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter storage</td>
<td>Immediate storage</td>
<td></td>
</tr>
</tbody>
</table>

Note: Please re-power on the sensor to take effective after set.
The default slave address for sensors with a waterproof aviation connector is 18.
The default slave address for sensors with hookup wires is 1.

### BAUDRATE

<table>
<thead>
<tr>
<th>Parameter range</th>
<th>0-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0:1200bps</td>
<td></td>
</tr>
<tr>
<td>1:2400bps</td>
<td></td>
</tr>
<tr>
<td>2:4800bps</td>
<td></td>
</tr>
<tr>
<td>Default: 3</td>
<td></td>
</tr>
</tbody>
</table>
### Protocol --- Serial communication Protocol

<table>
<thead>
<tr>
<th>Parameter Range</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>0~1</td>
<td>0</td>
</tr>
<tr>
<td>0: Modbus RTU</td>
<td></td>
</tr>
<tr>
<td>1: Modbus ASCII</td>
<td></td>
</tr>
</tbody>
</table>

**Parameter storage**: Immediate storage

### Parity --- Serial communication Check bit

<table>
<thead>
<tr>
<th>Parameter Range</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>0~2</td>
<td>0</td>
</tr>
<tr>
<td>0: none</td>
<td></td>
</tr>
<tr>
<td>1: even parity check</td>
<td></td>
</tr>
<tr>
<td>2: Odd parity check</td>
<td></td>
</tr>
</tbody>
</table>

**Parameter storage**: Immediate storage

### Data Bits

<table>
<thead>
<tr>
<th>Parameter Range</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1: 8 data bits</td>
<td></td>
</tr>
</tbody>
</table>

**Parameter storage**: Immediate storage

### Stop Bits

<table>
<thead>
<tr>
<th>Parameter Range</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>0~1</td>
<td>0</td>
</tr>
<tr>
<td>0: 1 stop bit</td>
<td></td>
</tr>
<tr>
<td>1: 2 stop bits</td>
<td></td>
</tr>
</tbody>
</table>

**Parameter storage**: Immediate storage
### RESPONSE DELAY

<table>
<thead>
<tr>
<th>Parameter range</th>
<th>0-255</th>
<th>Default: 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter storage</td>
<td>Immediate storage</td>
<td></td>
</tr>
</tbody>
</table>

Serial communication delay response used in the following circumstances: when the host sends a request command, delay module (RESPONSEDELAY * 10 milliseconds), then the response data is returned to the host. For example, to set up RESPONSEDELAY = 5, so delay module 5 * 10 = 50 millisecond response requesting host. Set to 0 for no delay an immediate response. This command is mainly used to host from RS485 transmission switch state to the receiving state relatively slow speed of occasions.

### ACTIVE OUTPUT INTERVAL

<table>
<thead>
<tr>
<th>Parameter range</th>
<th>0-255</th>
<th>Default: 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter storage</td>
<td>Immediate storage</td>
<td></td>
</tr>
</tbody>
</table>

Serial communication active output time interval used in the following circumstances: hosts that do not need to send a request command module active output response data and output interval for ACTIVEOUTPUTINTERVAL second, such as setting ACTIVEOUTPUTINTERVAL = 5. So module every 5 seconds according to set up the communication protocol of a debate output data. Set to 0 when the active output is invalid, the main request before response. This command is mainly used in GPRS wireless transmission, terminal active node data transmission occasions.

Note: when the active output data is set, only one module can be connected on the RS485 bus.
5.4 Communication Sample

In the following instructions, the data at the beginning of the 0x or the ending of the H is a 16 - band data. Modbus protocol with two common types of registers:
(1) To maintain the register, storage data is not lost, it is read and write. Usually with function number 3 (0x03) read, use function number 6 (0x06) or 16 (0x10) write.
(2) The input registers are used to store a number of read - only physical variables, such as temperature values, that are read - only and usually read with a function number 4 (0x04).

5.4.1 Function number 3 communication sample

Common request format: AA 03 RRRR NNNN CCCC

<table>
<thead>
<tr>
<th>AA</th>
<th>1 byte</th>
<th>Address, 0-255</th>
</tr>
</thead>
<tbody>
<tr>
<td>03</td>
<td>1 byte</td>
<td>Function number 3</td>
</tr>
<tr>
<td>RRRR</td>
<td>2 byte</td>
<td>Start register address, high byte in front</td>
</tr>
<tr>
<td>NNNN</td>
<td>2 byte</td>
<td>read the number of registers N, high byte in the front</td>
</tr>
<tr>
<td>CCCC</td>
<td>2 byte</td>
<td>CRC CHECK</td>
</tr>
</tbody>
</table>

Common request format: AA 03 MM VV0 VV1 VV2 VV3… CCCC

<table>
<thead>
<tr>
<th>AA</th>
<th>1 byte</th>
<th>Address, 0-255</th>
</tr>
</thead>
<tbody>
<tr>
<td>03</td>
<td>1 byte</td>
<td>Function number 3</td>
</tr>
<tr>
<td>MM</td>
<td>1 byte</td>
<td>Returns the number of data byte in the register value</td>
</tr>
<tr>
<td>VV0,VV1</td>
<td>2 byte</td>
<td>Returns the first register value</td>
</tr>
<tr>
<td>VV2,VV3</td>
<td>2 byte</td>
<td>Returns the second register value</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>Returns the &quot;N&quot; register value (N=MM/2)</td>
</tr>
<tr>
<td>CCCC</td>
<td>2 byte</td>
<td>CRC CHECK</td>
</tr>
</tbody>
</table>

For example: to read register 0x0200-0x0201, namely from the machine address and baud rate for example

Ask: 01 03 0200 0002 C5B3

<table>
<thead>
<tr>
<th>Address</th>
<th>1 byte</th>
<th>0x01</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function number</td>
<td>1 byte</td>
<td>0x03</td>
</tr>
</tbody>
</table>
Start register address | 2byte | 0x0200
---|---|---
Register number | 2byte | 0x0002
Check | 2byte | 0xC5B3

Respond: 01 03 04 00 01 00 03 EB F2

| Address | 1byte | 0x01 |
| Function number | 1byte | 0x03 |
| Effective byte number | 1byte | 0x04 |
| Slave address register value | 2byte | 0x00 (From machine address high byte) |
| | | 0x01 (From machine address low byte) |
| The baud rate register value | 2byte | 0x00 (High baud rate byte) |
| | | 0x03 (low baud rate byte) |
| Check | 2byte | 0xEBF2 |

### 5.4.2 Function number 4 communication sample

Common request format: AA 04 RRRR NNNN CCCC

| AA | 1byte | Address, 0-255 |
| 04 | 1byte | Function number 4 |
| RRRR | 2byte | Start register address, High byte in front |
| NNNN | 2byte | o read the number N Register, high byte in the front |
| CCCC | 2byte | CRC CHECK |

Common request format: AA 04 MM VV0 VV1 VV2 VV3… CCCC

| AA | 1byte | Address, 0-255 |
| 04 | 1byte | Function number 4 |
| MM | 1byte | Returns the number of data byte in the register value |
| VV0, VV1 | 2byte | Returns the first register value |
| VV2, VV3 | 2byte | Returns the second register value |
| … | … | Returns the "N" register value (N=MM/2) |
For example: to read the register 0x0000-0x0003, that reads the temperature, water content, electrical conductivity value

Ask: 01 04 0000 0003 B00B

<table>
<thead>
<tr>
<th>Address</th>
<th>1byte</th>
<th>0x01</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function number</td>
<td>1byte</td>
<td>0x04</td>
</tr>
<tr>
<td>Start register address</td>
<td>2byte</td>
<td>0x0000</td>
</tr>
<tr>
<td>Register number</td>
<td>2byte</td>
<td>0x0003</td>
</tr>
<tr>
<td>Check</td>
<td>2byte</td>
<td>0xB00B</td>
</tr>
</tbody>
</table>

Respond: 01 04 06 08 90 0E 93 02 4E D2 57

<table>
<thead>
<tr>
<th>Address</th>
<th>1byte</th>
<th>0x01</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function number</td>
<td>1byte</td>
<td>0x04</td>
</tr>
<tr>
<td>Effective byte number</td>
<td>1byte</td>
<td>0x06</td>
</tr>
<tr>
<td>Temperature register value</td>
<td>2byte</td>
<td>0x08 0x90</td>
</tr>
<tr>
<td>Volume water content register value</td>
<td>2byte</td>
<td>0x0E 0x93</td>
</tr>
<tr>
<td>Conductivity register value</td>
<td>2byte</td>
<td>0x02 0x4E</td>
</tr>
<tr>
<td>Check</td>
<td>2byte</td>
<td>0xD257</td>
</tr>
</tbody>
</table>

### 5.4.3 Function number 6 communication sample

Common request format: AA 06 RRRR VVVV CCCC

<table>
<thead>
<tr>
<th>AA</th>
<th>1byte</th>
<th>Address, 0-255</th>
</tr>
</thead>
<tbody>
<tr>
<td>06</td>
<td>1byte</td>
<td>Function number6</td>
</tr>
<tr>
<td>RRRR</td>
<td>2byte</td>
<td>Register address, high byte in front</td>
</tr>
<tr>
<td>VVVV</td>
<td>2byte</td>
<td>To write the value of the register, the high byte is in the front</td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>CCCC</td>
<td>2byte</td>
<td>CRC CHECK</td>
</tr>
</tbody>
</table>

Common request format: AA 06 RRRR VVVV CCCC

<table>
<thead>
<tr>
<th>AA</th>
<th>1byte</th>
<th>Address, 0-255</th>
</tr>
</thead>
<tbody>
<tr>
<td>06</td>
<td>1byte</td>
<td>Function number 6</td>
</tr>
<tr>
<td>RRRR</td>
<td>2byte</td>
<td>Register address, high byte in front</td>
</tr>
<tr>
<td>VVVV</td>
<td>2byte</td>
<td>To write the value of the register, the high byte is in the front</td>
</tr>
<tr>
<td>CCCC</td>
<td>2byte</td>
<td>CRC CHECK</td>
</tr>
</tbody>
</table>

For example: to write register 0x0021, namely the temperature unit for Fahrenheit cases

Ask: 01 06 0021 0001 1800

<table>
<thead>
<tr>
<th>Address</th>
<th>1byte</th>
<th>0x01</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function number</td>
<td>1byte</td>
<td>0x06</td>
</tr>
<tr>
<td>Start register address</td>
<td>2byte</td>
<td>0x0021</td>
</tr>
<tr>
<td>Register number</td>
<td>2byte</td>
<td>0x0001</td>
</tr>
<tr>
<td>Check</td>
<td>2byte</td>
<td>0x1800</td>
</tr>
</tbody>
</table>

Respond: 01 06 0021 0001 1800

<table>
<thead>
<tr>
<th>Address</th>
<th>1byte</th>
<th>0x01</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function number</td>
<td>1byte</td>
<td>0x06</td>
</tr>
<tr>
<td>Start register address</td>
<td>2byte</td>
<td>0x0021</td>
</tr>
<tr>
<td>Register number</td>
<td>2byte</td>
<td>0x0001</td>
</tr>
<tr>
<td>Check</td>
<td>2byte</td>
<td>0x1800</td>
</tr>
</tbody>
</table>

5.4.4 Function number 16 communication sample

Common request format: AA 10 RRRR NNNN MM VVVV1 VVVV2 …CCCC
<table>
<thead>
<tr>
<th>AA</th>
<th>1byte</th>
<th>Address, 0-255</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 (HEX)</td>
<td>1byte</td>
<td>Function number16 (10 binary system)</td>
</tr>
<tr>
<td>RRRR</td>
<td>2byte</td>
<td>Start register address, High byte in front</td>
</tr>
<tr>
<td>NNNN</td>
<td>2byte</td>
<td>To read the number N Register, high byte in the front</td>
</tr>
<tr>
<td>MM</td>
<td>1byte</td>
<td>The number of byte to write the value of the register</td>
</tr>
<tr>
<td>VVVV1</td>
<td>2byte</td>
<td>To write the value of the first register, the high byte is in the front.</td>
</tr>
<tr>
<td>VVVV2</td>
<td>2byte</td>
<td>To write the value of the second register, the high byte is in the front.</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>To write the value of the &quot;N&quot; register, the high byte is in the front. N=MM/2</td>
</tr>
<tr>
<td>CCCC</td>
<td>2byte</td>
<td>CRC CHECK</td>
</tr>
</tbody>
</table>

Common request format: AA 10 RRRR NNNN CCCC

<table>
<thead>
<tr>
<th>AA</th>
<th>1byte</th>
<th>Address, 0-255</th>
</tr>
</thead>
<tbody>
<tr>
<td>10(HEX)</td>
<td>1byte</td>
<td>Function number16 (10 binary system)</td>
</tr>
<tr>
<td>RRRR</td>
<td>2byte</td>
<td>Start register address, High byte in front</td>
</tr>
<tr>
<td>NNNN</td>
<td>2byte</td>
<td>To read the number N Register, high byte in the front</td>
</tr>
<tr>
<td>CCCC</td>
<td>2byte</td>
<td>CRC CHECK</td>
</tr>
</tbody>
</table>

For example: to write register 0x0200-0x0201 is set from the machine address is 1, the baud rate is 19200bps as an example

Ask: 01 10 0200 0002 04 0001 0004 BACC

<table>
<thead>
<tr>
<th>0x01</th>
<th>1byte</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x10 (HEX)</td>
<td>1byte</td>
<td>Function number 16 (DEC)</td>
</tr>
<tr>
<td>0x0200</td>
<td>2byte</td>
<td>Start register address, High byte in front</td>
</tr>
<tr>
<td>0x0002</td>
<td>2byte</td>
<td>To read the number N Register, high byte in the front</td>
</tr>
<tr>
<td>0x04</td>
<td>1byte</td>
<td>The number of byte to write the value of the register</td>
</tr>
<tr>
<td>0x0001</td>
<td>2byte</td>
<td>To write such as from the station address register</td>
</tr>
<tr>
<td>Address</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>0x0004</td>
<td>2byte</td>
<td>To write such as from the station address register</td>
</tr>
<tr>
<td></td>
<td></td>
<td>value is 4</td>
</tr>
<tr>
<td>0xBACC</td>
<td>2byte</td>
<td>CRC CHECK</td>
</tr>
</tbody>
</table>

Respond: 01 10 0200 0002 4070

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>1byte</td>
<td>Address</td>
</tr>
<tr>
<td>0x10 (HEX)</td>
<td>1byte</td>
<td>Function number16 (DEC)</td>
</tr>
<tr>
<td>0x0200</td>
<td>2byte</td>
<td>Start register address, high byte in the front</td>
</tr>
<tr>
<td>0x0002</td>
<td>2byte</td>
<td>To read the number N Register, high byte in the front</td>
</tr>
<tr>
<td>0x4070</td>
<td>2byte</td>
<td>CRC CHECK</td>
</tr>
</tbody>
</table>
5.5 CRC16 Check Algorithm

// -----------------------------------------------
// CRC calculation of C51 language function is as follows
// Enter the parameter 1: snd, to be the name of the byte Check array
// Input parameters 2: num, the total number of Check to be byte
// Function return value: Check and
// -----------------------------------------------
unsigned int calc_crc16 (unsigned char *snd, unsigned char num)
{
    unsigned char i, j;
    unsigned int c, crc=0xFFFF;
    for(i = 0; i < num; i++)
    {
        c = snd[i] & 0x00FF;
        crc ^= c;
        for(j = 0; j < 8; j++)
        {
            if (crc & 0x0001)
            {
                crc>>=1;
                crc^=0xA001;
            }
            else
            {
                crc>>=1;
            }
        }
    }
    return(crc);
For example: to read the register 0x0000-0x0002, that reads the temperature, water content, electrical conductivity value

<table>
<thead>
<tr>
<th>Host Ask:01 0400000003 B00B (8 byte)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
</tr>
<tr>
<td>Function number</td>
</tr>
<tr>
<td>Start register address</td>
</tr>
<tr>
<td>Register number</td>
</tr>
<tr>
<td>Check</td>
</tr>
</tbody>
</table>

When the host needs to send data to the sensor, it will need to send Check data stored in the snd array. (01 04 00 00 00 03 A total of 6 byte), Among them num=6

Pseudo code as follows:

```c
unsigned char request[8]={01,04,00,00,00,03,00,00}; // The last two 00,00 are CHECK CRC 
unsigned char num=6; // Calculate the array of the first 6 CRC CHECK byte 
unsigned int crc16=0; 
crc16= calc_crc16 (request, num); 
request[6]= crc16%256; // Store check CRC in an array to be sent 
request[7]= crc16/256; 
CommPort.Send(request, 8); // Send data through serial port
```

Sensor Respond:01 04 06 08 90 0E 93 02 4E D2 57 (11 byte)

<table>
<thead>
<tr>
<th>Address</th>
<th>1byte</th>
<th>0x01</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function number</td>
<td>1byte</td>
<td>0x04</td>
</tr>
<tr>
<td>Effective byte number</td>
<td>1byte</td>
<td>0x06</td>
</tr>
<tr>
<td>Temperature register value</td>
<td>2byte</td>
<td>0x08</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x90</td>
</tr>
<tr>
<td>Volume water content register value</td>
<td>2byte</td>
<td>0x0E</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x93</td>
</tr>
<tr>
<td>Conductivity register value</td>
<td>2byte</td>
<td>0x02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x4E</td>
</tr>
<tr>
<td>Check</td>
<td>2byte</td>
<td>0xD257</td>
</tr>
</tbody>
</table>
When the host receives the 11 byte data returned by the sensor, the following CRC calculation is performed, where num=11

Pseudo code as follows:

```c
unsigned char response[11]= { 01 04 06 08 90 0E 93 02 4E D2 57 }; // The last two byte are the CHECK CRC that the sensor returns
unsigned char num=11; // Calculate the entire return of the 11 CRC CHECK byte
unsigned int crc16=0;
crc16= calc_crc16 (response, num);
if(crc16==0)
{
    // Check CRC correctly, you can use the returned data
}
else
{
    // Check CRC error, can not be used to return the data
}
```

To get results back to 0 so the success of Check, if Check fails to return to a nonzero value. If the Check does not succeed, it shows that the transmission process is wrong, should give up the collected data, re collection.

The success of the Check, use the following formula to calculate the temperature (negative to complement representation) and conductivity of H at the end of the 16 hexadecimal data:

- Temperature = \( \frac{(08H*256+90H)}{100}=2192/100=21.92 °C \)
- Volumetric water content = \( \frac{(0EH*256+93H)}{100}=3731/100=37.31% \)
- Conductivity = \( 02H*256+4EH=2*256+78 =590 \ \mu s/cm \)
## 6 Document version

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
<th>Editor</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.0</td>
<td></td>
<td>First Version</td>
<td></td>
</tr>
<tr>
<td>V1.1</td>
<td>04/28/23</td>
<td>Modify entire document</td>
<td>Yvonne.Meng</td>
</tr>
</tbody>
</table>