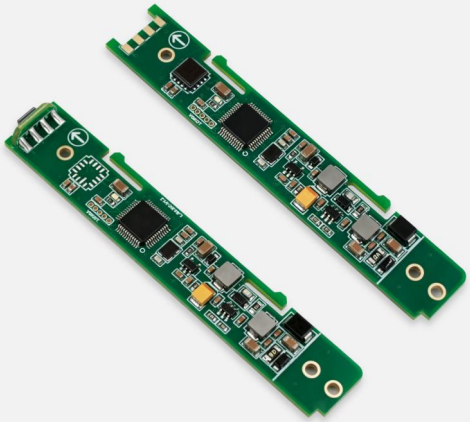
HTS3D High-Precision Tilt Sensor

Product Manual

V1.5-20260519

Technical Highlights



- ❖ Dual-axis tilt measurement;
- ❖ Measurement range: ±85°;
- ❖ Measurement accuracy: 0.01°;
- ❖ Resolution: 0.001°;
- ❖ DC 5~36V wide voltage input.

Product Introduction

The HTS3 is a biaxial tilt sensor launched by RUICOM Technology, primarily suitable for industrial engineering fields such as foundation pit tilt sensor monitoring and bridge inspection. This product inherits the stability of the HTS series while offering improved accuracy, superior temperature drift performance, and long-term stability controllable within 0.01 degrees Celsius. Through a built-in 5th-order filtering algorithm combined with biaxial gravity composite calculation, it can calculate the sensor's roll and pitch angles. Before leaving the factory, each product undergoes multi-faceted turntable calibration to correct biaxial cross-axis errors, effectively reducing mutual interference between measurement axes under large ranges. The product is equipped with an RS485 interface and integrated protection circuitry to prevent interference with the RS485 bus; the power supply section employs multi-stage voltage regulation and overvoltage/overcurrent protection design. This product meets industrial-grade application standards, possesses stable performance and good expandability, and is suitable for various harsh industrial control environments.

Application Scope

Typical applications of this product series include: foundation pit inclination measurement, railway locomotive monitoring, industrial automation, engineering machinery platform leveling systems, photovoltaic tracking bracket control systems, aerospace attitude reference systems, precision instrument leveling devices, and platform leveling in the field of high-precision tilt measurement.

Performance Parameters

HTS3D	Condition	Parameters	Unit
Measurement Range		±85	°
Measurement Axis		Dual Axis: X、Y	Axis
Resolution		0.001	°
Measurement Accuracy	@25℃	0.01	°
Long-Term Stability		0.01	°
Zero-Point Temperature Coefficient	-40 ~ 85℃	±0.0002	°/℃
Sensitivity Temperature Coefficient	-40 ~ 85℃	≤50	ppm/℃
Power-On Startup Time		1	S
Response Bandwidth		5	Hz
Output Signal	RS485		
Electromagnetic Compatibility	According to EN61000 and GBT17626		
Insulation Resistance	≥100 megohms		
Shock Resistance	100g@11ms, three-axis (half sine wave)		
Vibration Resistance	10grms、10 ~ 1000Hz		
Weight	≤10g(excluding cables)		

1) Resolution: The smallest identifiable angular change within the effective measurement range, reflecting its ability to monitor minute tilt fluctuations.
2) Measurement Accuracy: A comprehensive evaluation of the sensor's performance indicators under standard ambient temperature, including the overall performance of linearity error, repeatability error, hysteresis error, zero-point offset error, and cross-axis sensitivity error.
3) Long-Term Stability: The deviation between the maximum and minimum values of the sensor's output value over a year of continuous operation at room temperature.
4) Zero-Point Temperature Coefficient: The ratio of the sensor's output value to temperature changes under zero-input conditions, defined as the ratio of the zero-point offset within the rated operating temperature range to the room temperature reference value.
5) Sensitivity Temperature Coefficient: A stability index of the sensor's full-scale output value with temperature changes, characterizing the drift rate of sensitivity relative to the room temperature reference value within the rated temperature range.

Electrical Parameters

Parameter	Condition	MIN	TYP	MAX	Unit
Power supply voltage	Standard	5	12	36	V
	Customized		Other Voltage		
Operating current	No load		5		mA
Operating temperature		-40		+85	℃
Storage temperature		-40		+85	℃

Ordering and Selection

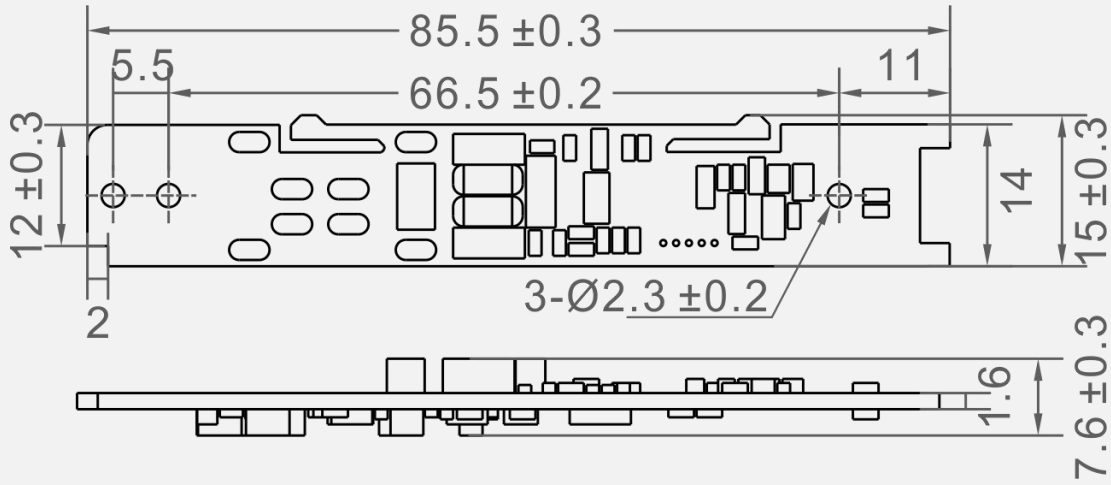
Model		Communication Protocol		Installation Orientation
HTS3D	-	77: 77 Protocol	-	H:Horizontal Installation
		MB: MODBUS Protocol		V:Vertical Installation
Example: HTS3D-77-H: 77 Protocol/Horizontal Installation.				

Product Dimensions

Vertical mounting plate dimension diagram

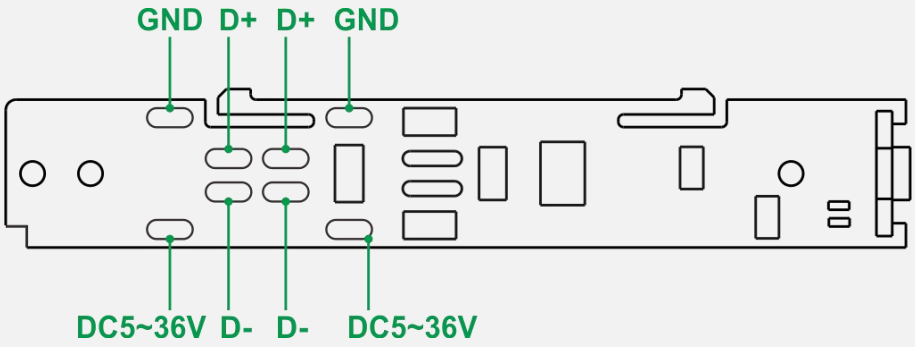
Product Dimensions: L86×W15×H9mm

Horizontal mounting plate dimension diagram



Product Dimensions: L85.5×W15×H7.6mm

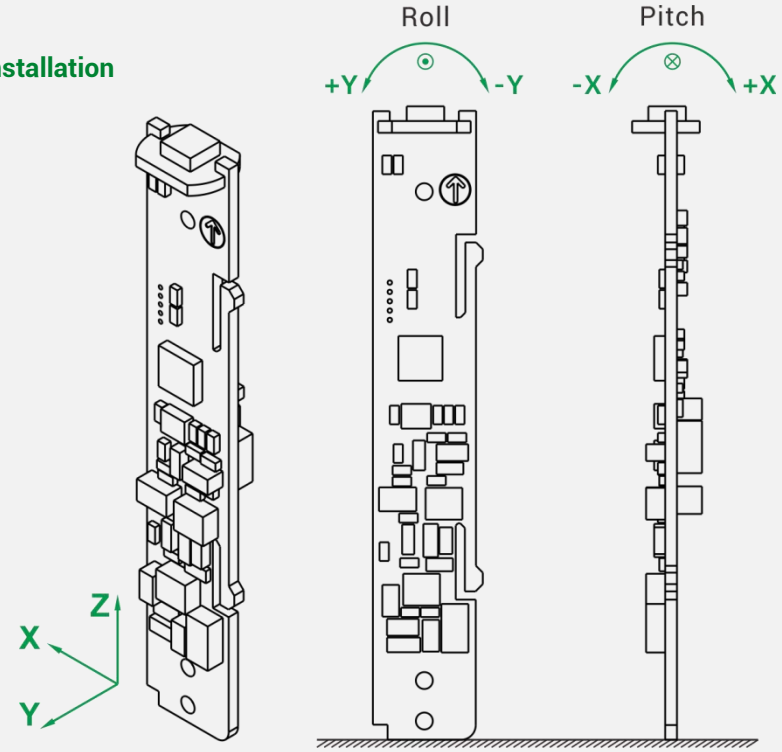
Interface Definition



Measurement Direction

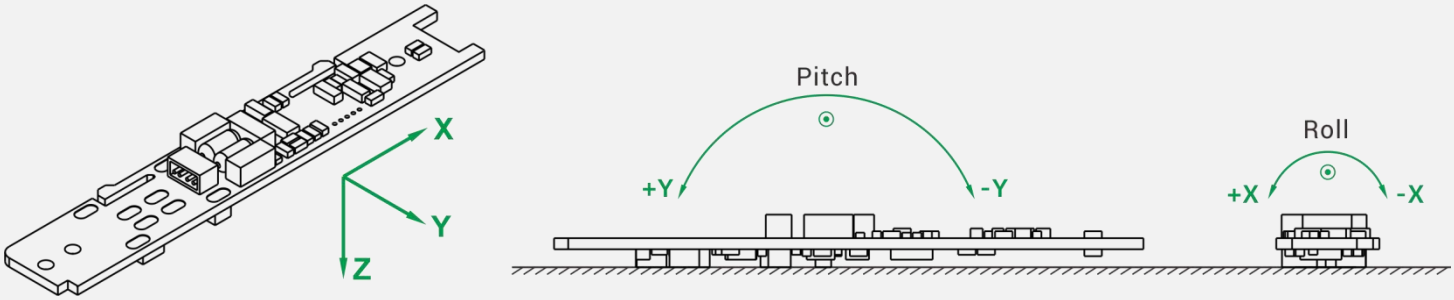
During installation, the sensor mounting surface should be kept parallel to the surface of the target being measured, and the effects of dynamics and acceleration on the sensor should be minimized. Please refer to the diagram below for installation instructions:

Vertical Installation



*Coordinate system: Right-Front-Top, corresponding to X-Y-Z.

Horizontal Installation



* Coordinate system: front-right-bottom, corresponding to X-Y-Z.

77 Communication protocol

1. Data frame format: (Default parameters: 8 data bits, 1 stop bit, no parity, address code 0, baud rate 2400.)

Identifier (1byte)	Data Length (1byte)	Address Code (4byte)	Command Word (1byte)	Data Field (n byte)	Checksum (1byte)
0x77	length	address	CMD	numbers	cksum

Data format: Hexadecimal;
Identifier: 0x77, start of frame flag;
Data length: Byte length from the data length field to the checksum, excluding the identifier;
Address code: Address of the acquisition module, 4 bytes, high byte first, low byte last (0x00000000 is a universal address);
Data field: Content and length vary depending on the command;
Checksum: Sum of data length, address code, command word, and data field, considering only the low byte.

2. Command format

Command word	Function	Note
0x01	Read X-axis angle	
0x02	Read Y-axis angle	
0x03	Query current horizontal roll angle (R angle)	Used to read the radius (R) angle during horizontal mounting.
0x04	Read X, Y, Z-axis angles and temperature	The Z-axis angle is meaningless.
0x05	Set relative/absolute zero point	
0x0A	Save settings	
0x0B	Set baud rate	
0x0D	Query relative/absolute zero point	
0x0F	Set address	
0x1F	Query current address	

2.1 Read the X-axis angle (command word 0x01)
Send command: 77 07 00 00 00 00 01 08

Identifier (1byte)	Data Length (1byte)	Address Code (4byte)	Command Word (1byte)	Data Field (0byte)	Checksum (1byte)
0x77	07	00 00 00 00	01	NULL	cksum

Response format:

Identifier (1byte)	Data Length (1byte)	Address Code (4byte)	Command Word (1byte)	Data Field (4byte)	Checksum (1byte)
0x77	0x0B	00 00 00 00	81	SX XX YY Y0	cksum

Note: The data field is 4 bytes. The returned angle is in compressed BCD code. S is the sign bit (0 for positive, 1 for negative), XXX is a 3-digit integer value, and YYY is a 3-digit decimal value. Other axes are the same, such as 10268110 representing -26.811 degrees.

2.2 Read the Y-axis angle (command word 0x02)
Send command: 77 07 00 00 00 00 02 09

Identifier (1byte)	Data Length (1byte)	Address Code (4byte)	Command Word (1byte)	Data Field (0byte)	Checksum (1byte)
0x77	07	00 00 00 00	02	NULL	cksum

Response format:

Identifier (1byte)	Data Length (1byte)	Address Code (4byte)	Command Word (1byte)	Data Field (4byte)	Checksum (1byte)
0x77	0x0B	00 00 00 00	82	SX XX YY Y0	cksum

Note: The data field is 4 bytes. The returned angle is in compressed BCD code. S is the sign bit (0 for positive, 1 for negative), XXX is a 3-digit integer value, and YYY is a 3-digit decimal value. Other axes are the same, such as 00268110 representing 26.811 degrees.

2.3 Query the current horizontal roll angle (R angle) (command word 0x03)
Send command: 77 07 00 00 00 00 03 0A

Identifier (1byte)	Data Length (1byte)	Address Code (4byte)	Command Word (1byte)	Data Field (0byte)	Checksum (1byte)
0x77	07	00 00 00 00	03	NULL	cksum

Response format:

Identifier (1byte)	Data Length (1byte)	Address Code (4byte)	Command Word (1byte)	Data Field (4byte)	Checksum (1byte)
0x77	0x0B	00 00 00 00	83	SX XX YY Y0	cksum

Note: Used for horizontal mounting alignment, to read the current mounting roll angle (R angle). The data field is 4 bytes, and the returned angle is in compressed BCD code. S is the sign bit (0 for positive, 1 for negative), XXX is a 3-bit integer value, and YYY is a 3-bit decimal value. Other axes are the same, such as 00268110 representing 26.811 degrees.

2.4 Read X, Y, Z axis angles and temperature (command word 0x04)
Send command: 77 07 00 00 00 00 04 0B

Identifier (1byte)	Data Length (1byte)	Address Code (4byte)	Command Word (1byte)	Data Field (0byte)	Checksum (1byte)
0x77	07	00 00 00 00	04	NULL	cksum

Response format:

Identifier (1byte)	Data Length (1byte)	Address Code (4byte)	Command Word (1byte)	Data Field (16byte)	Checksum (1byte)
0x77	0x17	00 00 00 00	0x84	SX XX xx x0 SY YY yy y0 SZ ZZ zz z0 ST TT tt t0	cksum

Note: The data field is 16 bytes, and the returned data is in compressed BCD code:
SX XX xx x0: X-angle value, S is the sign bit (0 for positive, 1 for negative), XXX is a 3-bit integer value, and xxx0 is a 3-bit decimal value. Other axes follow the same pattern, e.g., 00268110 represents 26.811 degrees.
SY YY yy y0: Y-angle value, S is the sign bit (0 for positive, 1 for negative), YYY is a 3-bit integer value, and yyy0 is a 3-bit decimal value. Other axes follow the same pattern, e.g.,

10255550 represents -25.555 degrees.

SZ ZZ zz z0: Y-angle value, S is the sign bit (0 for positive, 1 for negative), ZZZ is a 3-bit integer value, and zzz0 is a 3-bit decimal value. Other axes follow the same pattern, e.g., 10255550 represents -25.555 degrees.

ST TT tt t0: These represent temperature values. S is the sign bit (0 for positive, 1 for negative), TTT is a 3-digit integer value, and ttt0 is a 3-digit decimal value. Other axes follow the same pattern, such as 00233500 representing 23.35 degrees.

Note: Due to the internal initialization of the tilt angle, there will be no data response for the angle reading command within approximately 2 seconds after power-on.

2.5 Set relative/absolute zero point (command word 0x05)

Send command: 77 08 00 00 00 00 05 01 0E

Identifier (1byte)	Data Length (1byte)	Address Code (4byte)	Command Word (1byte)	Data Field (1byte)	Checksum (1byte)
0x77	08	00 00 00 00	05	00: Absolute Zero 01: Relative Zero	cksum

Response format:

Identifier (1byte)	Data Length (1byte)	Address Code (4byte)	Command Word (1byte)	Data Field (1byte)	Checksum (1byte)
0x77	0x08	00 00 00 00	0x85	00: Setup successful FF: Setup failed	cksum

Note: If set to absolute zero, the measurement angle will be based on the factory-set zero point; if set to relative zero, the measurement angle will be based on the current position as the zero point reference.

2.6 Set the baud rate (command word 0x0B)

Send command: 77 08 00 00 00 00 0B 00 13

Identifier (1byte)	Data Length (1byte)	Address Code (4byte)	Command Word (1byte)	Data Field (1byte)	Checksum (1byte)
0x77	08	00 00 00 00	0B	00: 2400 01: 4800 02: 9600 03: 19200 04: 115200	cksum

Note: Baud rate is bps;

00: 2400 (Default Value) 01: 4800 02: 9600 03: 19200 04: 115200。

Response format:

Identifier (1byte)	Data Length (1byte)	Address Code (4byte)	Command Word (1byte)	Data Field (1byte)	Checksum (1byte)
0x77	0x08	00 00 00 00	0x8B	00: Setup successful FF: Setup failed	cksum

2.7 Set module address (command word 0x0F)

Send command: 77 0B 00 00 00 00 0F 12 34 56 78 XX

Identifier (1byte)	Data Length (1byte)	Address Code (4byte)	Command Word (1byte)	Data Field (4byte)	Checksum (1byte)
0x77	0x0B	00 00 00 00	0F	12 34 56 78	cksum

Response format:

Identifier (1byte)	Data Length (1byte)	Address Code (4byte)	Command Word (1byte)	Data Field (1byte)	Checksum (1byte)
0x77	0x08	12 34 56 78	0x8F	00: Setup successful FF: Setup failed	cksum

Note: The address cannot be set to 0x00000000; 0x00000000 is a universal address.

2.8 Query relative/absolute zero point (command word 0x0D)
Send command: 77 07 00 00 00 00 0D 14

Identifier (1byte)	Data Length (1byte)	Address Code (4byte)	Command Word (1byte)	Data Field (0byte)	Checksum (1byte)
0x77	07	00 00 00 00	0D	NULL	cksum

Response format:

Identifier (1byte)	Data Length (1byte)	Address Code (4byte)	Command Word (1byte)	Data Field (1byte)	Checksum (1byte)
0x77	0x08	00 00 00 00	0x8D	00: Setup successful FF: Setup failed	cksum

2.9 Save settings (command word 0x0A)
Send command: 77 07 00 00 00 00 0A 11

Identifier (1byte)	Data Length (1byte)	Address Code (4byte)	Command Word (1byte)	Data Field (0byte)	Checksum (1byte)
0x77	07	00 00 00 00	0A	NULL	cksum

Response format:

Identifier (1byte)	Data Length (1byte)	Address Code (4byte)	Command Word (1byte)	Data Field (1byte)	Checksum (1byte)
0x77	0x08	00 00 00 00	0x8A	00: Setup successful FF: Setup failed	cksum

Note: This command is used to save the current settings. The power-off save function is only available after executing this command. Otherwise, the changed parameters will not be saved after power failure, and the baud rate will only take effect after a power failure.

3.0 Query current address (command word 0x1F) (the factory default address code is 0)
Send command: 77 07 00 00 00 00 1F 26

Identifier (1byte)	Data Length (1byte)	Address Code (4byte)	Command Word (1byte)	Data Field (0byte)	Checksum (1byte)
0x77	07	00 00 00 00	1F	NULL	cksum

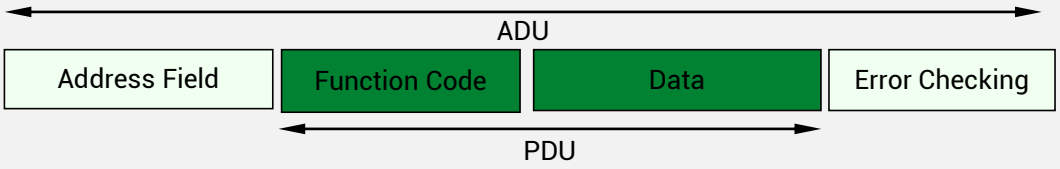
Response format:

Identifier (1byte)	Data Length (1byte)	Address Code (4byte)	Command Word (1byte)	Data Field (4byte)	Checksum (1byte)
0x77	0x0B	12 34 56 78	0x9F	12 34 56 78	cksum

MODBUS Communication Protocol

Communication format: 1 start bit, 8 data bits, no parity bit, 1 stop bit; Default parameters: Address code 1, baud rate 9600.

The MODBUS protocol stipulates that there should be at least 3.5 bytes of time between two data frames, and this sensor extends this time to 15ms.



Frame format:

1. Transmission format

(1)Command message format

Send read data command:

Address	Function Code	Data start address high bits	Data start address low bits	Number of data points (high byte)	Number of data points (lower digit)	CRC16 bit check
High bit First						

Return of the read data command:

Address	Function Code	Byte length	Data 1 input	Data 2 input	...	CRC16 bit check
High bit First	High bit First					

Send write data command:

Address	Function Code	Data start address high bits	Data start address low bits	Data high bit	Data low bit	bit check CRC16
High bit First						

Sending a write command returned the same result as the original command.:

Address	Function Code	Data start address high bits	Data start address low bits	Data high bit	Data low bit	bit check CRC16
High bit First						

(2) Abnormal response return

illegal functions:

Slave address	Function code	Error code	CRC16 Check
	80H+ Original Function Code	01	

illegal data address:

Slave address	Function code	Error code	CRC16 Check
	80H+ Original Function Code	02	

illegal data values:

Slave address	Function code	Error code	CRC16 Check
	80H+ Original Function Code	03	

The requested data value is empty (no value exists):

Slave address	Function code	Error code	CRC16 Check
	80H+ Original Function Code	0F	

2. Module RTU instructions and descriptions

Function Code	Data Start Address	Number of Data Words	Number of Data Bytes	Instruction Description
04H	0016H	8	16	Read current measured values XYZ and temperature value -- floating-point data
04H	001EH	2	4	Read roll angle when horizontal -- floating-point data
06H	0004H	1	2	0x00A0: 2400 0x00A1: 4800 0x00A2: 9600 0x00A3: 19200 0x00A4: 38400 0x00A5: 115200 (Change baud rate and power cycle effective).
06H	0006H	1	2	Write 0x00FF to clear the GAMMA angle (γ). Write 0x0001 to generate a GAMMA angle (γ). The generated GAMMA angle is stored in FLASH.
03H	0008H	2	4	Read the GAMMA (γ) angle (this value is stored in FLASH). If no GAMMA angle (γ) is generated, a data null exception is returned.
03H	0030H	1	2	Read address code -- integer data
06H	0030H	1	2	Modify address code -- integer data
10H	0002H	2	4	ms MBRTU mode password authentication

2.1Command to read current measurement and temperature values

Command frames sent by the microcomputer:

Addr	04h	0h	16h	0h	08h	CRC16H	CRC16L

Data frames returned by the module:

Addr	04h	Length=16	Data1 ~ Data16	CRC16H	CRC16L

The instruction returns a 16-byte floating-point number; the high byte comes first, followed by the low byte.

Length: 16 Return data length.

Data1~Data16: Return data.

Note: If Addr uses the 0xFA general address when sending the instruction, Addr will return to the module's current address. The instruction processing is the same.

Example:

a.Read X and Y azimuth angles and temperature values. Function code 04. Register address: 16 bytes. Number of bytes: 16.

Send: 01 04 00 16 00 08 10 08

Recv: 01 04 10 C0 8F 12 6F 3E FA E1 48 43 16 19 9A 42 04 00 00 F9 31

ID<1> X: -4.471°

ID<1> Y: +0.490°

ID<1> Azimuth Angle: 150.10°

ID<1> Temperature: 33℃

Note: Due to sensor initialization, the current measurement value and temperature command will be read and responded to within approximately 2 seconds after power-on.: Addr 84 0F CR16H CR16L。

2.2 Address code instruction

a. Read address command (the factory default address code is 1).
Command frames sent by the microcomputer:

Addr	03h	00h	30h	00h	01h	CRC16H	CRC16L

Data frames returned by the module:

Addr	03h	Length=2	Data1 ~ Data2	CRC16H	CRC16L

The instruction returns 2 bytes of integer data, with module address encoding 1~255; 0xfa is a universal address code.

It can be used as the starting address before the read address code instruction, and the read module reads the address code. When sending, Addr uses 0xFA, and when returning, Addr and Data1~Data2 are all the current address codes of the module.

b.Broadcast read ID function code 03 register address 30 bytes 1 universal code fa.
Send: FA 03 00 30 00 01 91 8E
Recv: 01 03 02 00 01 79 84

c.Modify address code command
Command frames sent by the microcomputer:

Addr	06h	0h	30h	0h	New Addr	CRC16H	CRC16L

Data frames returned by the module:

New Addr	06h	0h	30h	0h	New Addr	CRC16H	CRC16L

New Addr: Module address encoding is 1~255. When sending a command, regardless of whether Addr uses the general address 0xfa or the actual address code, the return value is always the current address code.
Password authentication is required before operation. If the operation is successful, the original instruction will be returned; otherwise, an error code will be displayed.
Example:
Set ID, function code 06, register address 30 bytes, number of bytes 2.
Send: 01 06 00 30 00 02 08 04
Recv: 02 06 00 30 00 02 08 37

2.3 MBRTU Mode Cryptographic Authentication Commands

Command frames sent by the microcomputer:

Addr	10h	0h	02h	0h	2h	04h	Data1~Data4	CRC16H	CRC16L

Data frames returned by the module:

Addr	10h	0h	02h	0h	02h	CRC16H	CRC16L

Data1~Data4: 50 53 57 44。

The above command will be returned if the setup is successful; otherwise, an error code will be displayed.

Note: In MBRTU mode, all write and tuning commands require password authentication.

Example:
Function:Password authentication
Send:01 10 00 02 00 02 04 50 53 57 44 AD 64

Receive: 01 10 00 02 00 02 E0 08
Password authentication - Success!

Function: Modify address Address 1 Modify 2
Send: 01 06 00 30 00 02 08 04
Receive: 02 06 00 30 00 02 08 37
The address has been changed:【 2 】

2.4 Generate GAMMA(γ) angle

Condition: Place the device horizontally, align all the tubes into a straight line, and try to straighten them as much as possible. Keep the tubes still. First, execute the command to clear the GAMMA(γ) angle, and then execute the command to generate the GAMMA(γ)angle. The successfully generated GAMMA(γ) angle will be saved in FLASH, and you can read and use it later by executing the command to read the GAMMA(γ) angle.

Note that the tubes for generating GAMMA(γ) angles must be placed horizontally to ensure all tubes are straight. Password authentication is required before executing any commands related to clearing or generating GAMMA(γ) angles.

a. Clear GAMMA(γ) angle

Command frame sent by microcomputer:

Addr	06h	00h	06h	00h	FFh	CRC16H	CRC16L

Data frames returned by the module:

Addr	06h	00h	06h	00h	FFh	CRC16H	CRC16L

b. Generate GAMMA(γ) angle

Command frames sent by the microcomputer:

Addr	06h	00h	06h	00h	01h	CRC16H	CRC16L

Data frames returned by the module:

Addr	06h	00h	06h	00h	01h	CRC16H	CRC16L

c. Read GAMMA(γ) angle

Command frames sent by the microcomputer:

Addr	03h	0h	08h	0h	02h	CRC16H	CRC16L

Data frames returned by the module:

Addr	03h	Length=4	Data1 ~ Data4	CRC16H	CRC16L

The instruction returns a 4-byte floating-point number; the high byte first, the low byte last.

If the GAMMA(γ) angle is empty, an exception response is returned:

Addr	83h	0Fh	CRC16H	CRC16L

Note: If no GAMMA angle is generated, the response when reading the GAMMA angle is: Addr 83 0F CRC16H CRC16L.

●The CRC code calculation method is as follows:

1) Preset a 16-bit register to hexadecimal FFFF (i.e., all 1s); this register is called the CRC register;
2) XOR the first 8-bit binary data (i.e., the first byte of the communication information frame) with the lower 8 bits of the 16-bit CRC register and put the result into the CRC register;
3) Shift the contents of the CRC register one bit to the right (towards the least significant bit), fill the most significant bit with 0, and check the shifted-out bit;
4) If the shifted-out bit is 0: repeat step 3 (shift right one bit again);
If the shifted-out bit is 1: XOR the CRC register with the polynomial A001 (1010 0000 0000 0001);
5) Repeat steps 3 and 4 until the right shift is performed 8 times, thus processing all 8 bits of data;
6) Repeat steps 2 to 5 to process the next byte of the communication information frame;

7) After calculating all bytes of the communication information frame according to the above steps, the high and low bytes of the resulting 16-bit CRC register are swapped;
8) The final CRC register content is the CRC code.

```
unsigned char const auchCRCHi[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40
} ;

unsigned char const auchCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7,
0x05, 0xC5, 0xC4,
0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB,
0x0B, 0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE,
0xDF, 0x1F, 0xDD,
0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2,
0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32,
0x36, 0xF6, 0xF7,
0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E,
0xFE, 0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B,
0x2A, 0xEA, 0xEE,
0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27,
0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1,
0x63, 0xA3, 0xA2,
0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD,
0x6D, 0xAF, 0x6F,
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8,
0xB9, 0x79, 0xBB,
0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4,
0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0,
0x50, 0x90, 0x91,
0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94,
0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59,
```

```
0x58, 0x98, 0x88,
0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D,
0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83,
0x41, 0x81, 0x80,
0x40
};

unsigned short MB_Make_CRC16( unsigned char *puchMsg, unsigned short usDataLen )
{
	unsigned char uchCRCHi = 0xFF;
	unsigned char uchCRCLo = 0xFF;
	unsigned uIndex;

	while (usDataLen--)
	{
		uIndex = uchCRCLo^*puchMsg++ ;
		uchCRCLo = uchCRCHi^auchCRCHi[uIndex];
		uchCRCHi = auchCRCLo[uIndex];
	}

	return (uchCRCLo<<8|uchCRCHi);
}
```