



STSL Low-Power Tilt Sensor

Product Manual

V1.8-20260918

Technical Highlights

- ❖ Tri-axial measurement;
- ❖ Measurement range: $\pm 30^\circ$;
- ❖ Resolution: 0.001° ;
- ❖ Measurement accuracy: $< \pm 15^\circ$: 0.01° / $< \pm 30^\circ$: 0.05° ;
- ❖ IP67 protection rating.



Product Introduction

The STSL is a compact, low-power, multi-parameter dual-axis inclinometer launched by RUICOM Technology for industrial field control applications. Offering high measurement accuracy, it enables simultaneous monitoring of a target's tilt angle, oscillation frequency, oscillation amplitude, and center offset, making it suitable for industrial engineering scenarios such as the monitoring of towers and bridges. The product features excellent angular output stability, superior measurement precision, and outstanding thermal drift characteristics, with long-term stability rated at better than 0.02° . It employs a fifth-order filtering algorithm to process gravity signals and resolve the sensor's X- and Y-axis angles. Each unit undergoes cross-axis error calibration using a multi-faceted rotary table prior to shipment, helping to minimize inter-axis interference across a wide measurement range. Options include RS485 or TTL interfaces with integrated protection circuitry to reduce bus communication interference. The power supply system incorporates multi-stage voltage regulation alongside over-voltage and over-current protection. Designed for the industrial sector, the product offers robust operational stability and strong scalability, making it suitable for a variety of harsh industrial control environments. Please refer to the product datasheet for specific technical specifications.

Application Scope

Typical application scenarios for this product series include foundation pit inclinometer monitoring, industrial automation, leveling systems for construction machinery platforms, control systems for photovoltaic tracking mounts, aerospace attitude reference systems, precision instrument leveling devices, and other high-precision tilt measurement fields such as platform leveling.



Performance Parameters

STSL	Condition	Parameter	Unit
Measurement range		±30	°
Measurement axis		X Y Z	Axis
Resolution ¹⁾		0.001	°
Measurement accuracy ²⁾	@25°C	< ±15°: 0.01° / < ±30°: 0.05°	°(RMS)
Long-term stability ³⁾		<0.02	°
Zero-point temp coefficient	-40~85°C	0.0008	°/°C
Sensitivity temp coefficient	-40~85°C	≤0.01	%/°C
Oscillation frequency measurement range		<12.5Hz	
Oscillation frequency measurement accuracy		0.20Hz	
Center offset	Calculation based on installation height and the angle between the Z-axis and the perpendicular bisector		
Dynamic oscillation amplitude	Calculation based on installation height and the dynamic angle between the Z-axis and the perpendicular bisector		
Power-up startup time		2.5s	
Response time ⁴⁾		<0.1s	
Response frequency		>10HZ	
Output signal		RS485/TTL Optional	
Communication protocol		MODBUS RTU Protocol	
Electromagnetic compatibility (EMC)		In accordance with EN 61000 and GB/T 17626	
Insulation resistance		≥100 MΩ	
Shock resistance		100g @ 11ms, 3-axis (half-sine wave)	
Vibration resistance		10grms / 10~1000Hz	
Waterproof rating		IP67	
Cable		Standard 2m cable (4 x 0.2 mm ²)	
Weight		≤150g(including 2m cable)	

¹⁾ Resolution: Refers to the smallest change in the measured quantity that the sensor can detect and distinguish within its measurement range.

²⁾ Measurement Accuracy: Calculated as the root mean square (RMS) error between measured values and the actual angle, based on multiple measurements (more than 16) taken at room temperature.

³⁾ Long-term Stability: Refers to the deviation between the maximum and minimum values observed during one year of continuous operation at room temperature.

⁴⁾ Response Time: Refers to the time required for the sensor's output to reach the standard value following a change in the measured angle.

Ordering and Selection

Model	Encapsulation	Signal Output
STSL	3:PCBA Board	485: RS485
	6:Standard housing	TTL: TTL

E.g: STSL6-485: Standard housing / RS485 Output.

✓ Electrical Parameters

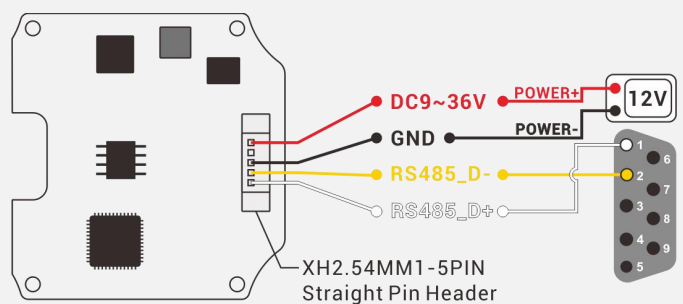
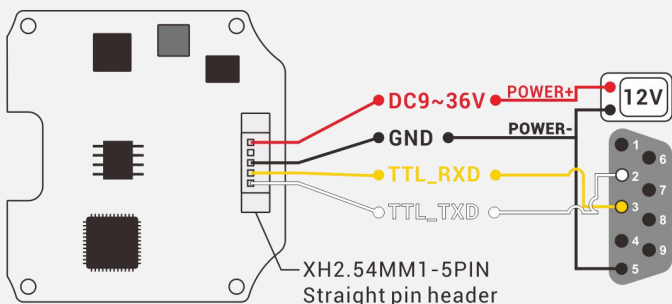
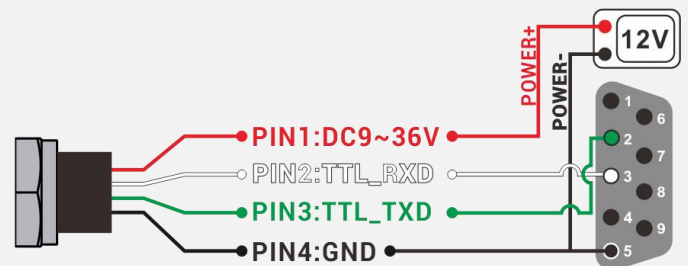
Parameter	Condition	MINI	TYP	MAX	Unit
Supply voltage	Standard	9	12	36	V
	Customized		5		V
Operating current	No load		6.5(12V)		mA
Operating temp		-40		+85	°C
Storage temp		-40		+85	°C

✓ Indicator Light Operation Instructions

After the device is powered on, the red indicator light remains steadily lit; it begins to flash once a connection is successfully established and data transmission is underway. If the red light remains steadily lit during the connection process, it indicates a connection failure, and a reconnection attempt is required.

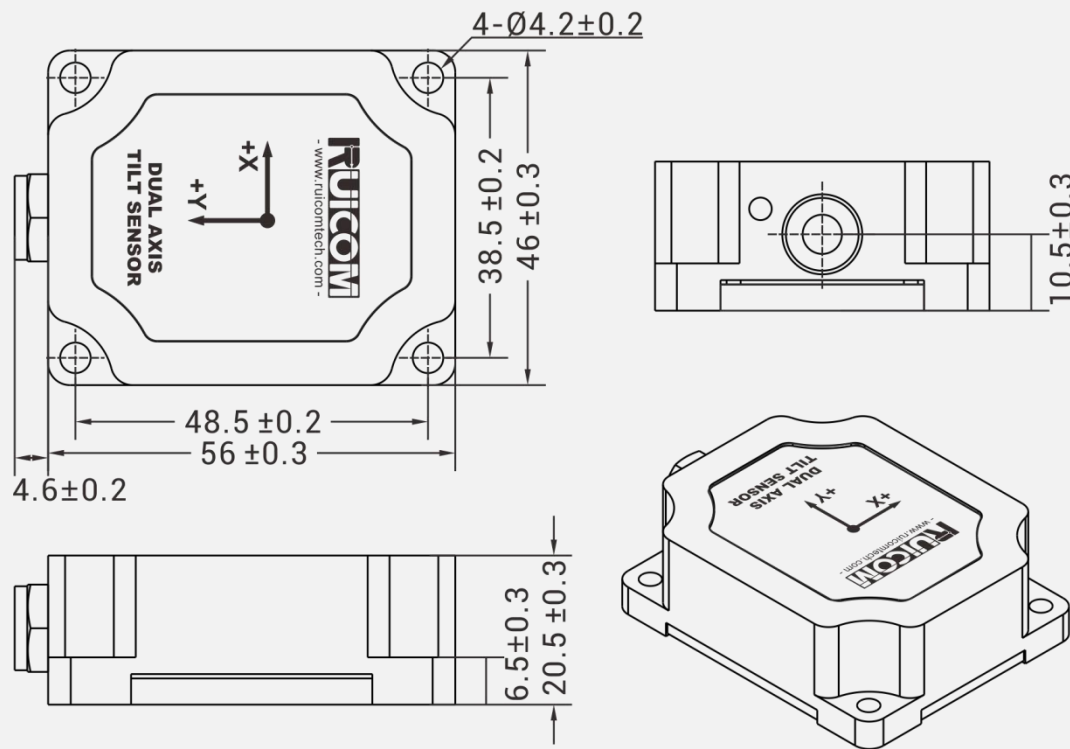
✓ Interface Definition

Housing Wire Colors	PCBA Wire Colors	Definition	Description
Red	Red	DC9~36V	Positive pole of power supply
White	White	TTL_RXD / RS485_D+	receive data
Green	Yellow	TTL_TXD / RS485_D-	Send data
Black	Black	GND	Negative pole of power supply



Product Dimensions

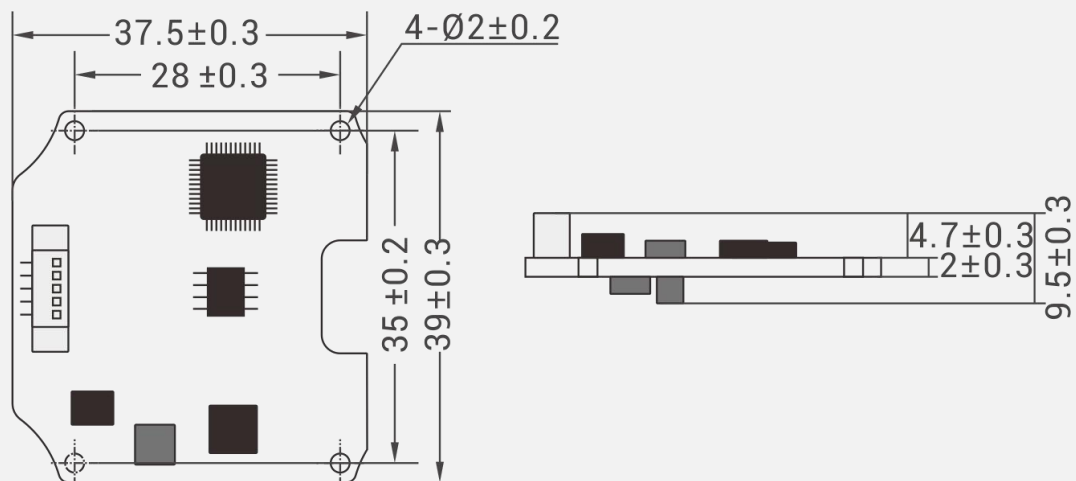
Package dimensions:



Housing dimensions: L60.6 × W46 × H20.5 mm

Recommended screws: 4 × M4 screws

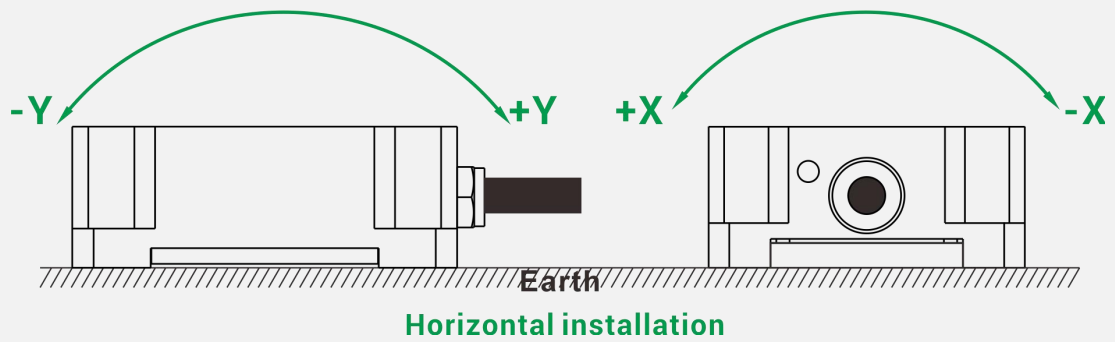
PCBA Dimensions:



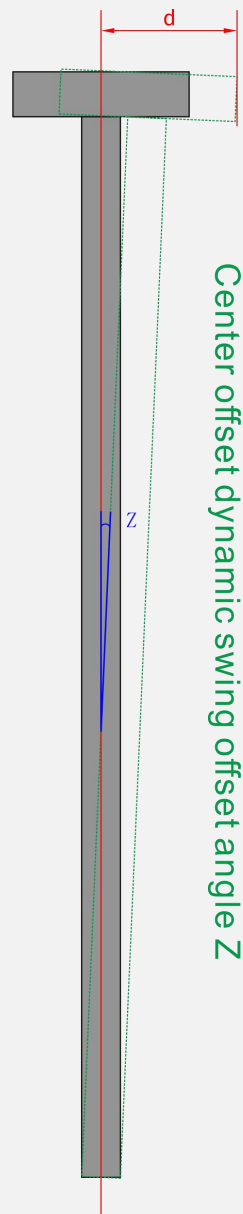
Housing dimensions: L37.5 × W39 × H9.5 mm

Recommended screws: 4 × M2 screws

✓ Measurement direction



✓ ILLUSTRATION Of Dynamic Offset Swing Measurement



MODBUS RTU Communication protocol

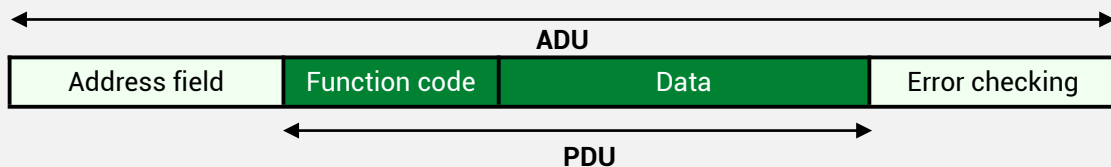
Default parameters: Parity: N (No parity), Baud rate: 9600 bps.

1. Protocol Type: MODBUS Protocol

Communication format: 1 start bit, 8 data bits, 1 stop bit.

Address range: 1–247; address 0 is the universal broadcast address. When a write command uses address 0 (broadcast address), all sensors respond to the command but do not send a reply.

Configuration commands (Function Code 06): Password authentication is required prior to executing any configuration command; authentication must be performed for each setting operation; otherwise, there will be no response.



2. Transmission format

(1) Command message format

Address	Function Code	Data Start Address (High Byte)	Data Start Address (Low Byte)	Number of Data Points (High Byte)	Number of Data Points (Low Byte)	CRC16 Bit Checksum
High-order bits first						

Return after sending the read data command:

Address	Function Code	Byte Length	Data 1 Input	Data 2 Input	...	CRC16 Bit Checksum
High-order bits first	High-order bits first					

Send write data command:

Address	Function Code	Data Start Address (High Byte)	Data Start Address (Low Byte)	High-order data bits	Low-order data bits	CRC16 Bit Checksum
High-order bits first						

Return after sending the write data command—Return the command as-is.:

Address	Function Code	Data Start Address (High Byte)	Data Start Address (Low Byte)	High-order data bits	Low-order data bits	CRC16 Bit Checksum
High-order bits first						

(2) Abnormal response returned

Illegal function:

Slave station address	Function Code	Exception Code	CRC16 Checksum
	80H+ Original Function Code	01	

Invalid data address:

Slave station address	Function Code	Exception Code	CRC16 Checksum
	80H+ Original Function Code	02	

Invalid data address:

Slave station address	Function Code	Exception Code	CRC16 Checksum
	80H+ Original Function Code	03	

The request data value is empty (no value provided):

Slave station address	Function Code	Exception Code	CRC16 Checksum
	80H+ Original Function Code	0F	

3. Module RTU Register Address Instructions and Explanatory Information

Register Add.	Register Name	Data Type	Word Count	Register Type	Supported Function Codes and Content
0x0002-0x0003	Password Authentication	UINT32	2	WO	0x10(Only supports writing two registers.) 0x10 (Supports writing 2 registers only) Password authentication is required before configuring registers; authentication must be performed for each configuration operation.
0x0004	Baud Rate	UINT16	1	WO	0x06(Write single register) 0x00A0: 2400bps 0x00A1: 4800bps 0x00A2: 9600bps(Default) 0x00A3: 19200bps 0x00A4: 38400bps 0x00A5: 115200bps Settings take effect after power-up.
0x0006	Parity Bit Setting Register	UINT16	1	WO	0x06 (Write single register); write the following values to configure the parity bit. 1) 0x0000: Even parity; 2) 0x0001: No parity (Default); 3) 0x0002: Odd parity;
0x0007	Software Reset Register	UINT16	1	WO	0x06 (Write single register); write the following value to perform a software reset. 1) 0xFFFF: Perform software reset;
0x0008-0x0009	Installation Height Register (cm)	UINT32	2	R/W	0x03/0x10 (Read/Write 2 registers); Default value: 3000 cm
0x000D	Tilt Angle Alarm Threshold	Uint16	1	R/W	0x03/0x06 (Read/Write single register); Angle=uint16*0.01(°) Default value: 0.50°
0x000E	Oscillation Amplitude Alarm Threshold (mm)	Uint16	1	R/W	0x03/0x06 (Read/Write single register); Default value: 500 mm
0x000F	Oscillation Frequency Alarm Threshold	Uint16	1	R/W	0x03/0x06 (Read/Write single register); Hz = uint16*0.01(hz) Default value: 0.50 Hz

0x0010	Horizontal Offset Alarm Threshold (mm)	Uint16	1	R/W	0x03/0x06 (Read/Write single register); Default value: 500 mm
0x0011	Z-Axis Angle Relative/ Absolute Zero Point Setting	Uint16	1	R/W	0x03/0x06 (Read/Write single register); 0x0000: Absolute zero point (default); 0x0001: Relative zero point;
0x0012	XY-Axis Angle Relative/ Absolute Zero Point Setting	Uint16	1	R/W	0x03/0x06 (Read/Write single register); 0x0000: Absolute zero point (default); 0x0001: Relative zero point;
0x0016-0x0017	X-Axis Tilt Angle	float	2	RO	0x03 (Read output register)
0x0018-0x0019	Y-Axis Tilt Angle	float	2	RO	0x03 (Read output register)
0x001A-0x001B	Z-Axis Angle (Angle between Z-axis and vertical)	float	2	RO	0x03 (Read output register)
0x001C-0x001D	Temperature (Tc)	float	2	RO	0x03 (Read output register)
0x001E	AM Oscillation Amplitude	Uint16 (mm)	1	RO	0x03 (Read output register)
0x001F	FR Oscillation Frequency	Uint16 (*0.01Hz)	1	RO	0x03 (Read output register) Note: *0.01 scaling factor (Hz)
0x0020	Center Offset	Uint16 (mm)		RO	0x03 (Read output register)
0x0021	Alarm Status	Uint16	1	RO	0x03 (Read output register) Bit 0: Tilt angle alarm; Bit 1: Oscillation frequency alarm; Bit 2: Oscillation amplitude alarm; Bit 3: Center offset alarm; Bit = 1: Alarm active; Status automatically clears after reading;
0x0030	Sensor Address (Modbus Address)	UINT16	1	W/R	0x03 (Supports reading single register); 0x06 (Supports writing single register); Default address: 0x01

3.1 Read Current Measurement Value Command

Example of a command to read data using Modbus function code 03H, with a starting register address of 0x0016 and a register count of 0x000C:

Host query command:		Slave response:		
Sensor Address	01H	Sensor Address	01H	
Function Code	03H	Function Code	03H	
Access the starting address of the register.	00H	Data field length	18H	
	16H	Data field	BFH	X-axis data (ft32)
Number of registers	00H		3AH	

	0CH		B0H	
CRC	A40BH		AEH	
			BFH	
			45H	Y-axis data (ft32)
			66H	
			0AH	
			3FH	
			87H	Z-axis data (ft32)
			D8H	
			00H	
			41H	
			F3H	Temperature T (ft32)
			16H	
			14H	
			16H	Amplitude Am (uint16)
			94H	
			02H	Vibration frequency freq (uint16)
			36H	
			16H	Horizontal offset (uint16)
			94H	
			00H	Alarm flag (uint16)
			01H	
		CRC	DDF1H	

Note: The data field of the slave device's response frame is BF 3A B0 AE BF 45 66 0A 3F 87 D8 00 41 F3 16 14 16 94 02 36 16 94 00 01

The X-axis corresponds to bytes 1–4 of the data field, the Y-axis to bytes 5–8, the Z-axis to bytes 9–12, temperature (T) to bytes 13–16, Am to bytes 17–18, freq to bytes 19–20, and offset to bytes 21–22. All data follows big-endian format (most significant byte first). The XYZ angles and temperature (T) are stored as float32 values; Am, freq, and offset are stored as uint16 values. Am and offset are in millimeters (mm), and the frequency (Hz) is obtained by dividing freq by 100.

Taking the aforementioned data frame as an example, the angle conversion process is as follows:

- 1) X-axis: BF 3A B0 AEH = -0.729°;
- 2) Y-axis: BF 45 66 0AH = -0.771°;
- 3) Z-axis: 3F 87 D8 00H = 1.601°;
- 4) Temperature (T): 41 F3 16 14H = 30.386°;
- 5) Amplitude (Am): 16 94H = 5780 mm;
- 6) Vibration frequency (Freq): 0236H = 566; 566/100 = 5.66 Hz;
- 7) Horizontal offset: 1694H = 5780 mm;
- 8) Alarm flag: 0001H (tilt angle alarm present);

3.2 Address-code instruction

3.2.1 Read Address Instruction Register Address: 0030H Number of Registers: 0001H

Host query command:		Slave response:		
Sensor address	00H	Sensor address	01H	
Function code	03H	Function code	03H	
Register access address	00H	Data field length	02H	
	30H	Data field	00H	Sensor address
Number of registers	00H		01H	
	01H	CRC	7984H	
CRC	85D4H			

Command frame sent by the host:

Addr	03h	0h	30h	0h	1h	CRC16L	CRC16H
------	-----	----	-----	----	----	--------	--------

Data frame returned by the module:

Addr	03h	Length=2	Data1~Data2	CRC16L	CRC16H
------	-----	----------	-------------	--------	--------

The data returned by the command is a 2-byte integer; the module address ranges from 1 to 247, with 0x00 serving as the broadcast address.

It can be used as the starting address for the command to read the module's address code.

When sending, use 0x00 for Addr; upon return, both Addr and Data1~Data2 reflect the module's current address code.

Send: 00 03 00 30 00 01 85 D4

Recv: 01 03 02 00 01 79 84

3.2.2 Address code setting command

Command to set the sensor address code:		Slave response:	
Sensor address	01H	Sensor address	01H
Function code	06H	Function code	06H
Address	00H	Register address	00H
	30H		30H
New address of the sensor	00H	New address of the sensor	00H
	04H		04H
CRC	8806	CRC	8806

Command frame sent by the host:

Addr	06h	0h	30h	0h	NewAddr	CRC16L	CRC16H
------	-----	----	-----	----	---------	--------	--------

Data frame returned by the module:

NewAddr	06h	0h	30h	0h	NewAddr	CRC16L	CRC16H
---------	-----	----	-----	----	---------	--------	--------

NewAddr: Module address encoding 1~247. When Addr is 0x00, there is no response after the address is successfully modified.

Password authentication is required before the operation; upon success, the original command is returned; otherwise, there is no response.

Example:

Set ID: Function Code 06, Register Address 30, Byte Count 2.

Send: 01 06 00 30 00 04 88 06

Recv: 01 06 00 30 00 04 88 06

3.3 MBRTU Mode Password Authentication Command

Register Address: 0002H Number of Registers: 0002H

MBRTU mode password authentication command:		Slave response:	
Sensor address	01H	Sensor address	01H

Function code	10H	Function code	10H
Register address	00H	Register address	00H
	02H		02H
Data word length	00H	Data word length	00 H
	02H		02H
Number of data bytes	04H	CRC	E008
Data	50H		
	53H		
	57H		
	44H		
CRC	AD64		

Command frame sent by the host:

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

Data frame returned by the module:

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

Data1~Data4: 50 53 57 44。

If the setting is successful, the above command is returned; otherwise, an error code is returned.

Note: In MBRTU mode, all write and calibration 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 - Successful!

Function: Modify address (Address 1 changed to 2)

Send: 01 06 00 30 00 02 08 04

Receive: 02 06 00 30 00 02 08 37

Address modified to: [2]

3.4 Installation height

3.4.1 Set installation height (default height: 3000 cm)

Register address: 0008H Number of registers: 0002H

Command to set installation height:		Slave response:	
Sensor Address	01H	Sensor Address	01H
Function Code	10H	Function Code	10H
Register address	00H	Register address	00H
	08H		08H
Data word length	00H	Data word length	00H
	02H		02H
Number of data bytes	04H	CRC	C00A
Height data (cm)	00H		
	00H		
	0BH		

	B8H		
CRC	F54B		

The height data (cm) is a uint32: 00 00 0B B8 = 3000 cm.

Command frame sent by the host:

Addr	10h	0h	08h	00h	02h	04h	Data1~Data4	CRC16L	CRC16H
------	-----	----	-----	-----	-----	-----	-------------	--------	--------

Data frame returned by the module:

Addr	10h	0h	08h	0h	02h	CRC16L	CRC16H
------	-----	----	-----	----	-----	--------	--------

Send: 01 10 00 08 00 02 04 00 00 0B B8 F5 4B

Receive: 01 10 00 08 00 02 C0 0A

The height data (cm)—Data1 through Data4—is of type uint32: 00 00 0B B8 = 3000 cm.

If the setting is successful, the command above is returned; otherwise, there is no response.

3.4.2 Read installation height

Host query command:		Slave response:	
Sensor address	01H	Sensor address	01H
Function code	03H	Function code	03H
Register access address	00H	Data field byte count	04H
	08H		00H
Number of registers	00H	Height data field (cm)	00H
	02H		0BH
CRC	45C9H		B8H
		CRC	FD71H

Note: The altitude data is a 4-byte uint32; the high byte comes first, followed by the low byte.

Command frame sent by the host:

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

Data frame returned by the module:

Addr	03h	Length=4	Data1~Data4	CRC16L	CRC16H
------	-----	----------	-------------	--------	--------

The data returned by the command is a 4-byte uint32, with the high-order byte first and the low-order byte last.

3.5 XY-axis relative/absolute zero point command

3.5.1 Command to query the zero-point status of the current XY angles

Register address: 0012H Number of registers: 0001H

Host query command:		Slave response:	
Sensor address	01H	Sensor address	01H
Function code	03H	Function code	03H
Register access address	00H	Data field length	02H
	12H		00H
Number of registers	00H	Data field	01H
	01H		7984H
CRC	240FH	CRC	

Note: Data fields: 0000H: Absolute zero point; 0001H: Relative zero point;

Example: Command frame sent by the host:

Addr	03h	00h	12h	00h	01h	CRC16L	CRC16H
------	-----	-----	-----	-----	-----	--------	--------

Data frame returned by the module:

Addr	03h	Length=2	Data1~Data2	CRC16L	CRC16H
------	-----	----------	-------------	--------	--------

Send: 01 03 00 12 00 01 24 0F

Recv: 01 03 02 00 01 79 84

3.5.2 Command to set absolute XY angles and the zero point

Set XY angle absolute and zero command:		Slave response:	
Sensor Address	01H	Sensor Address	01H
Function Code	06H	Function Code	06H
Address	00H	Register Address	00H
	12H		12H
Data Content	00H	Data Content	00H
	01H		01H
CRC	E80FH	CRC	E80FH

Note: Data content: 0x0000: Absolute zero point; 0x0001: Relative zero point

Host sends command frame to set relative zero point:

Addr	06h	0h	12h	00h	01h	CRC16L	CRC16H
------	-----	----	-----	-----	-----	--------	--------

Data frame returned by the module:

Addr	06h	0h	12h	00h	01h	CRC16L	CRC16H
------	-----	----	-----	-----	-----	--------	--------

Password authentication is required before the operation; upon success, the original command is returned, otherwise there is no response.

Example:

Send: 01 06 00 12 00 01 E8 0F

Recv: 01 06 00 12 00 01 E8 0F

3.6 Z-axis relative/absolute zero point command

3.6.1 Command to query the current zero-point status of the Z-axis angle.

Register address: 0011H Number of registers: 0001H

Host query command:		Slave response:	
Sensor address	01H	Sensor address	01H
Function code	03H	Function code	03H
Register access address	00H	Data field length	02H
	11H		00H
Number of registers	00H	Data field	00H
	01H		00H
		CRC	B844H
CRC	D40FH		

Note: Data fields: 0000H: Absolute zero point; 0001H: Relative zero point;

Command frame sent by the host:

Addr	03h	00h	11h	00h	01h	CRC16L	CRC16H
------	-----	-----	-----	-----	-----	--------	--------

Data frame returned by the module:

Addr	03h	Length=2	Data1~Data2	CRC16L	CRC16H
------	-----	----------	-------------	--------	--------

Send: 01 03 00 11 00 01 D4 0F

Recv: 01 03 02 00 00 B8 44

3.6.2 Command to set the current Z-axis zero point

Command to set current Z-axis zero point:		Slave response:	
Sensor Address	01H	Sensor Address	01H
Function Code	06H	Function Code	06H
Address	00H	Register address	00H
	11H		11H
Data Content	00H	Data Content	00 H
	01H		01H

CRC	180FH	CRC	180FH
-----	-------	-----	-------

Note: Data content: 0x0000: Absolute zero point 0x0001: Relative zero point

The host sends a command frame to set the relative zero point:

Addr	06h	0h	11h	00h	01h	CRC16L	CRC16H
------	-----	----	-----	-----	-----	--------	--------

Data frame returned by the module:

Addr	06h	0h	11h	00h	01h	CRC16L	CRC16H
------	-----	----	-----	-----	-----	--------	--------

Password authentication is required before the operation; upon success, the original command is returned; otherwise, there is no response.

Example:

Send: 01 06 00 11 00 01 18 0F

Recv: 01 06 00 11 00 01 18 0F

3.7 Command to set communication baud rate

Register Address: 0004h Register Length: 0001H

Command to set communication baud rate:		Slave response:	
Sensor Address	01H	Sensor Address	01H
Function Code	06H	Function Code	06H
Address	00H	Register address	00H
	04H		04H
Data Content	00H	Data Content	00 H
	01H		01H
CRC	09CBH	CRC	09CBH

Data Content: 0x00A0: 2400bps 0x00A1: 4800bps 0x00A2: 9600bps
 0x00A3: 19200bps 0x00A4: 38400bps 0x00A5: 115200bps

Note: Takes effect after power cycling.

The host sends a command frame to set the baud rate:

Addr	06h	0h	04h	00h	A1h	CRC16L	CRC16H
------	-----	----	-----	-----	-----	--------	--------

Data frame returned by the module:

Addr	06h	0h	04h	00h	A1h	CRC16L	CRC16H
------	-----	----	-----	-----	-----	--------	--------

Password authentication is required before the operation; upon success, the original command is returned; otherwise, there is no response.

Example:

Tx: 01 06 00 04 00 A1 09 CB

Rx: 01 06 00 04 00 A1 09 CB

3.8 Instruction for setting the communication parity bit

Register Address: 0006h Register Length: 0001H

Command to set communication parity bit:		Slave response:	
Sensor Address	01H	Sensor Address	01H
Function Code	06H	Function Code	06H
Address	00H	Register address	00H
	06H		06H
Data Content	00H	Data Content	00H
	00H		00H
CRC	69CBH	CRC	69CBH

Data content: 0x0000: Even parity 0x0001: No parity 0x0002: Odd parity

Note: Takes effect after power cycling.

The host sends a command frame to set the parity bit:

Addr	06h	0h	06h	00h	00h	CRC16L	CRC16H
------	-----	----	-----	-----	-----	--------	--------

Data frame returned by the module:

Addr	06h	0h	06h	00h	00h	CRC16L	CRC16H
------	-----	----	-----	-----	-----	--------	--------

Password authentication is required before the operation; upon success, the original command is returned; otherwise, there is no response.

Example:

Send: 01 06 00 06 00 00 69 CB

Recv: 01 06 00 06 00 00 69 CB

3.9 Command to set the tilt alarm threshold

3.9.1 Command to query the current tilt alarm threshold

Register address: 000DH Number of registers: 0001H

Host query command:		Slave response:	
Sensor address	01H	Sensor address	01H
Function code	03H	Function code	03H
Register access address	00H	Data field length	02H
	0DH	Data field	00H
Number of registers	00H		14H
	01H	CRC	B84BH
CRC	15C9H		

Note: Data field: 0014H = 20/100 = 0.20°;

Default alarm angle: 0.5°.

Command frame sent by the host:

Addr	03h	00h	0Dh	00h	01h	CRC16L	CRC16H
------	-----	-----	-----	-----	-----	--------	--------

Data frame returned by the module:

Addr	03h	Length=2	Data1~Data2	CRC16L	CRC16H
------	-----	----------	-------------	--------	--------

TX: 01 03 00 0D 00 01 15 C9

RX: 01 03 02 00 14 B8 4B

3.9.2 Command to set the tilt alarm threshold

Command to set tilt alarm threshold:		Slave response:	
Sensor Address	01H	Sensor Address	01H
Function Code	06H	Function Code	06H
Address	00H	Register address	00H
	0DH		0DH
Data Content	00H	Data Content	00H
	32H		32H
CRC	99DCH	CRC	99DCH

Note: Data content: 0x0032 = 50/100 = 0.50°

Host sends command frame:

Addr	06h	0h	0Dh	00h	32h	CRC16L	CRC16H
------	-----	----	-----	-----	-----	--------	--------

Data frame returned by the module:

Addr	06h	0h	0Dh	0Dh	32h	CRC16L	CRC16H
------	-----	----	-----	-----	-----	--------	--------

Password authentication is required before operation. If the operation is successful, the original command will be returned. Otherwise, there will be no response.

Example:

TX: 01 06 00 0D 00 32 99 DC

RX: 01 06 00 0D 00 32 99 DC

3.10 Command to set the swing amplitude alarm threshold

3.10.1 Command to query the current oscillation amplitude alarm threshold

Register Address: 000EH Number of Registers: 0001H

Host query command:		Slave response:	
Sensor address	01H	Sensor address	01H
Function code	03H	Function code	03H
Register access address	00H	Data field length	02H
	0EH	Data field	00H
Number of registers	00H		64H
	01H	CRC	B9AFH
CRC	E5C9H		

Note: Data field: 0064H = 100 mm;

Default oscillation amplitude alarm threshold: 500 mm

Command frame sent by the host:

Addr	03h	00h	0Eh	00h	01h	CRC16L	CRC16H
------	-----	-----	-----	-----	-----	--------	--------

Data frame returned by the module:

Addr	03h	Length=2	Data1~Data2	CRC16L	CRC16H
------	-----	----------	-------------	--------	--------

TX: 01 03 00 0E 00 01 E5 C9

RX: 01 03 02 00 64 B9 AF

3.10.2 Command to set the swing amplitude alarm threshold

Command to set swing alarm threshold:		Slave response:	
Sensor Address	01H	Sensor Address	01H
Function Code	06H	Function Code	06H
Address	00H	Register address	00H
	0EH		0EH
Data Content	01H	Data Content	01H
	F4H		F4H
CRC	E81EH	CRC	E81EH

Note: Data content: 0x01F4 = 500mm

Host sends command frame:

Addr	06h	0h	0Eh	01h	F4h	CRC16L	CRC16H
------	-----	----	-----	-----	-----	--------	--------

Data frame returned by the module:

Addr	06h	0h	0Eh	01h	F4h	CRC16L	CRC16H
------	-----	----	-----	-----	-----	--------	--------

Password authentication is required before the operation; upon success, the original command is returned, otherwise there is no response.

Example:

TX: 01 06 00 0E 01 F4 E8 1E

RX: 01 06 00 0E 01 F4 E8 1E

3.11 Command to set the oscillation frequency alarm threshold

3.11.1 Command to query the current oscillation frequency alarm threshold

Register address: 000FH Number of Registers: 0001H

Host query command:		Slave response:	
Sensor address	01H	Sensor address	01H
Function code	03H	Function code	03H
Register access address	00H	Data field length	02H
	0FH	Data field	00H

Number of registers	00H		14H
	01H	CRC	B84BH
CRC	B409H		

Note: Data field: 0014H = 20/100 = 0.20 Hz;
 Default oscillation frequency alarm threshold: 0.5 Hz;
 Command frame sent by the host:

Addr	03h	00h	0Fh	00h	01h	CRC16L	CRC16H
------	-----	-----	-----	-----	-----	--------	--------

Data frame returned by the module:

Addr	03h	Length=2	Data1~Data2	CRC16L	CRC16H
------	-----	----------	-------------	--------	--------

TX: 01 03 00 0F 00 01 B4 09

RX: 01 03 02 00 14 B8 4B

3.11.2 Command to set the oscillation frequency alarm threshold

Command to set oscillation frequency alarm threshold:		Slave response:	
Sensor Address	01H	Sensor Address	01H
Function Code	06H	Function Code	06H
Address	00H	Register address	00H
	0FH		0FH
Data Content	00H	Data Content	00H
	32H		32H
CRC	381CH	CRC	381CH

Note: Data content: 0x0032 = 50 = 0.50 Hz

Host sends command frame:

Addr	06h	0h	0Fh	00h	32h	CRC16L	CRC16H
------	-----	----	-----	-----	-----	--------	--------

Data frame returned by the module:

Addr	06h	0h	0Fh	00h	32h	CRC16L	CRC16H
------	-----	----	-----	-----	-----	--------	--------

Password authentication is required before the operation; upon success, the original command is returned; otherwise, there is no response.

Example:

TX: 01 06 00 0F 00 32 38 1C

RX: 01 06 00 0F 00 32 38 1C

3.12 Command to set the horizontal offset alarm threshold

3.12.1 Command to query the current offset alarm threshold

Register address: 0010H Number of registers: 0001H

Host query command:		Slave response:	
Sensor Address	01H	Sensor Address	01H
Function Code	03H	Function Code	03H
Register access address	00H	Data field length	02H
	10H		01H
Number of registers	00H	Data field	F4H
	01H		B853H
CRC	85CFH	CRC	

Note: Data field: 01F4H = 500mm;

Default horizontal offset alarm threshold: 500mm

Command frame sent by the host unit:

Addr	03h	00h	10h	01h	01h	CRC16L	CRC16H
------	-----	-----	-----	-----	-----	--------	--------

Data frame returned by the module:

Addr	03h	Length=2	Data1~Data2	CRC16L	CRC16H
------	-----	----------	-------------	--------	--------

TX: 01 03 00 10 00 01 85 CF

RX: 01 03 02 01 F4 B8 53

3.12.2 Command to set the offset alarm threshold

Command to set offset alarm threshold:		Slave response:	
Sensor Address	01H	Sensor Address	01H
Function Code	06H	Function Code	06H
Address	00H	Register address	00H
	10H		10H
Data Content	00H	Data Content	00H
	64H		64H
CRC	89E4H	CRC	89E4H

Note: Data content: 0x01F4 = 500mm

Host sends command frame:

Addr	06h	0h	10h	00h	64h	CRC16L	CRC16H
------	-----	----	-----	-----	-----	--------	--------

Data frame returned by the module:

Addr	06h	0h	10h	00h	64h	CRC16L	CRC16H
------	-----	----	-----	-----	-----	--------	--------

Password authentication is required before the operation; upon success, the original command is returned; otherwise, there is no response.

Example:

TX: 01 06 00 10 00 64 89 E4

RX: 01 06 00 10 00 64 89 E4

● The method for calculating the CRC code is:

1. Initialize a 16-bit register to hexadecimal FFFF (i.e., all bits set to 1); refer to this as the CRC register;
2. Perform an XOR operation between the first 8-bit binary data (the first byte of the communication message frame) and the lower 8 bits of the 16-bit CRC register, storing the result in the CRC register;
3. Shift the contents of the CRC register one bit to the right (towards the least significant bit), filling the most significant bit with 0, and check the bit shifted out;
4. If the shifted-out bit is 0: Repeat step 3 (shift right by one bit again);
If the shifted-out bit is 1: Perform an XOR operation between the CRC register and the polynomial A001 (1010 0000 0000 0001);
5. Repeat steps 3 and 4 until 8 right shifts have been performed, thereby processing the entire 8-bit data;
6. Repeat steps 2 through 5 to process the next byte of the communication message frame;
7. After completing the calculation for all bytes of the communication message frame using the steps above, swap the high and low bytes of the resulting 16-bit CRC register;
8. The final content of the CRC register 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 (uchCRCHi<<8|uchCRCLo);
}
```