9427-S series

INDUCTANCE PROBE DISPLAY OPERATION MANUAL

www.insize.com

https://m.insize.com/9427-S.html

EN -- Please scan the QR code or visit the website for operation manual.

IT --- Scansiona il codice QR oppure visita il sito web per il manuale d'uso.

CZ -- Pro návod prosím naskenujte QR kód nebo navštivte webovou stránku.

ES -- Por favor, escanee el código QR o visite la página web para ver el manual de instrucciones.

FR -- Veuillez scanner le QR Code ou visiter notre site web pour accéder aux manuels d'utilisation.

DE -- Bitte scannen Sie den QR-Code oder besuchen Sie die Website für die Bedienungsanleitung.

PT -- Para aceder ao manual de instruções, por favor, faça a leitura do código QR ou visite o nosso site.

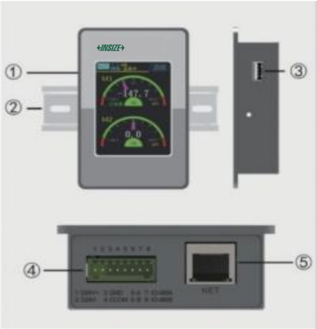
MN-947-S SERIES-E

V0

INSTALLATION AND INSTRUCTIONS

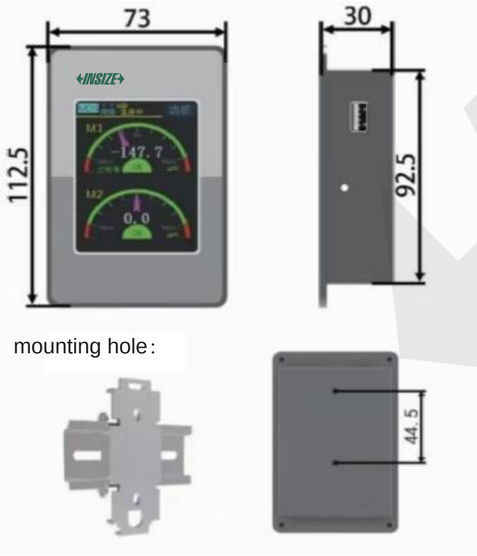
The 9427-S Inductive Probe Display is a digital measurement system that combines a high-precision AD chip, high-performance processor, and a touch LCD screen with high-definition display effects. A single unit can simultaneously display data from 4 LVDT sensors, and it also has functions such as tolerance judgment, IO control, feedback output, data transmission, multiple scheme selection, data storage, etc.

1.1 instruction



① 3.5" screen
② 35mm mounting rail
③ USB port
④ power and RS485 port
⑤ network port
(needs to be customised)

·Installation and dimensions（mm）



The holes match the holes on the rail

SPECIFICATION

					Sensor type*
Code	9427-S01A	9427-S02A	9427-S03A	9427-S04A	full bridge
	9427-S01B	9427-S02B	9427-S03B	9427-S04B	half bridge
Channel number	1	2	3	4	
Display	3.5" touchable LCD screen				
Display mode	banner display, dial display				
Unit	µm, mm				
Range (mm)	±1, ±2, ±5, ±10 (adjustable)				
Resolution (µm)	0.1, 1				
Communication port**	RS485 (Modbus RTU)				
Sampling time (ms)	2, 4, 8, 16, 32 (adjustable)				
Operation environment	0~40°C, <80% (non-condensing)				
Storage environment	-20~50°C, <80% (non-condensing)				
Power supply	DC 24V				
Dimension (L×W×H)	113×73×30mm				
Weight	252g				

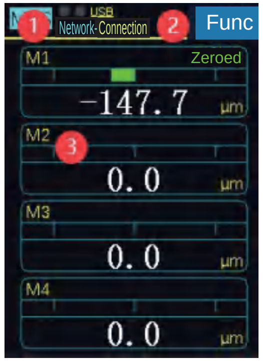
* Applicable inductance probes: INSIZE 9420 series, Mahr, Marposs, Solartron, TESA, DONG-DO, ACCRETECH, etc.

** Optional RS232 (Modbus RTU), network port (Modbus TCP and EtherNet/IP), WIFI (Modbus TCP and EtherNet/IP) or I/O

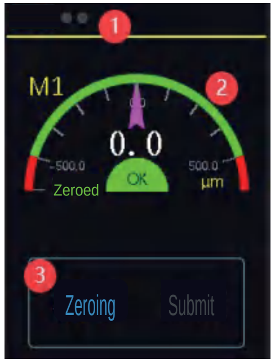
OPERATION GUIDE

2.1 Main measurement interface

After the instrument is powered on, it will automatically enter the measurement interface, presenting all the measured data as well as the status indication of whether it is qualified or not. Two display modes are provided, as follows



banner display



dial display

Banner display method (banner display is selected for more than 2 measurement items, and up to 8 items are displayed):

- Identification 1: Status information, including: transmission and reception status of communication, connection status of network, USB stick status.
- Identification 2: Function buttons, including: calibrate zero, submit.
- Identification 3: Measurement item value and status. m1\m2\m3\m4 indicates 4 measurement items.

Dial display method (1~2 measurement items can select dial):

- Identification 1: Status information, including: sending and receiving status of communication, connection status of network, and USB stick status.
- Identification 2: Measurement item value and status.
- Identification 3: Function buttons, including: calibrate zero, submit.

2.2 Tolerance and Zero

2.2.1 tolerance

Can change tolerances quickly, proceed as follows：

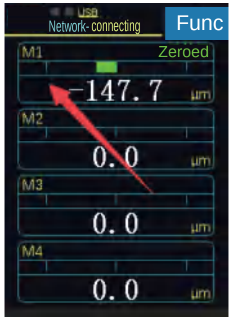


figure 3

Click anywhere in the M1 area

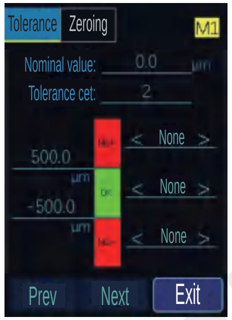


figure 4

Figure 4 shows the tolerance parameters of M1, the values are reduced from top to bottom, i.e., the top tolerance is the upper tolerance, the bottom tolerance is the lower tolerance, the interface can also change the nominal value. The tolerance is entered as a relative value and the nominal value as an absolute value in the following format:

- ◆ If the measured dimension is 30.500±0.010mm, then
  Nominal value: 30.500mm
  Upper tolerance: +0.010mm
  Lower tolerance: -0.010mm
- ◆ If the measured dimension is $30.500^{0}_{-0.01}$ mm, then
  Nominal value: 30.500mm
  Upper tolerance: +0mm
  Lower tolerance: -0.010mm

* Note that the units shown in Figure 4 are in microns (μm). If the units in mm are required, they can be changed in the system setup interface.

* In Fig. 4, the [Prev] [Next] button can be used to switch between different measurement items.

2.2.2 Zeroing

Zero calibration can be done in the main measurement interface by directly clicking the [Zeroing] button in the main measurement interface. If you want to set the parameters of calibration zero, such as change of calibration zero standard, change of calibration zero allowable range, you need to enter the secondary menu, the operation is as follows:

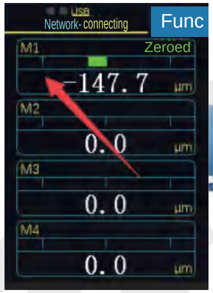


figure 5

点击 M1 区域的任意位置

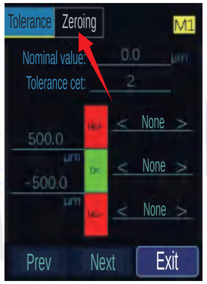


figure 6

点击【校零】按钮

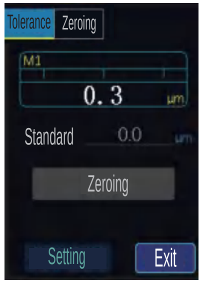


figure 7

Figure 7 shows the M1 calibration interface, the current state is not enabled calibration, you need to enter the [Calibration Settings], enable calibration, and set the appropriate range of calibration RMS value. During the measurement, the position of the sensor tooling is shifted and exceeds the range of valid values, and then the calibration will prompt the failure of calibration to remind the operator to pay attention to the problem.

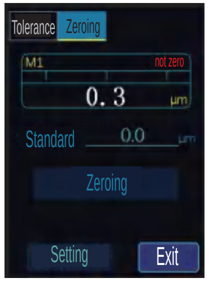


figure 8

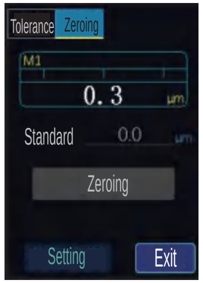


figure 8

Figure 8 shows the state after the calibration is turned on. After inputting the appropriate RMS range and standard value, you can click [Zeroing] or go back to the main measurement interface to calibrate the zero.

A common problem with zero calibration is that it fails. At this time, we should pay attention to the installation position of the sensor, as well as set the effective value range, to ensure that the normal installation position of the value of the pressure on the workpiece, within the effective value range. For example, after the sensor pressure on the workpiece value is: 0.5mm, the effective value range can be set to -0.8 ~ 0.8mm.

In addition, the gauges are also available with external IO triggered calibration and modbus command triggered calibration.

2.3 Sensors

2.3.1 Zeroing, Inverse

If you need to [Zeroing] the sensor, i.e. set the value of the current position of the sensor to 0, you can follow the steps below:

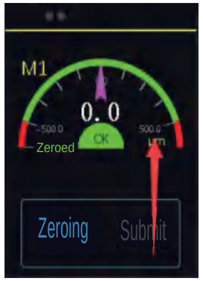
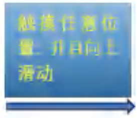


figure 9



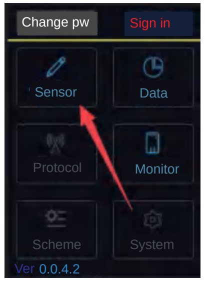
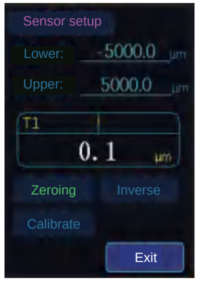


figure 10



Lower: -5000.0 µm
Upper: 5000.0 µm
T1 0. 1 µm

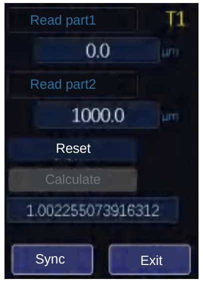
figure 11

After entering the sensor interface, you can see that the sensor T1 has been zeroed, if you want to cancel, click [Zeroing] again. The inverse operation is the same as the zero setting.

To switch to sensor T2, touch anywhere on T1 and swipe left.

2.3.2 Calculate

If the linearity of the sensor is found to be deviated during the actual measurement, the sensor can be re-calibrated. Firstly, enter the sensor interface by the steps in Fig. 9~Fig. 11, and then click the [Calibration] button (if there is no such button, you need to log in permission) to enter the calibration interface:



0.0 µm
1000.0 µm
1.002255073916312

figure 12

The calibration steps are as follows:

- Need 2 different sizes of standards, for example: standard block 1 is 30.000mm, standard block 2 is 31.000mm, the difference between the two is preferably more than 1mm, but not greater than the range of the sensor.
- Firstly, input 2 standard block size values, here input 0um,1000um.
- Place the standard block 1 into the corresponding position of the sensor, and click the [Read part 1] button in the screen. Then take out the standard block 1.
- Place the standard block 2 into the position just now, and then click the [Read part 2] button in the screen.
- Finally, click the [Calculate] button.

* Note the units used when entering standard block values. If the unit is in mm, enter 0mm,1mm; if the unit is in µm, enter 0um,1000µm.

[Sync] The currently corrected coefficients can be applied to other sensors.

2.4 communication status

When a measuring instrument needs to communicate with external devices such as PLCs and host computers, the byte stream of the current communication can be viewed on the monitoring interface of the measuring instrument to facilitate debugging. Please follow the steps below to enter the monitoring interface:

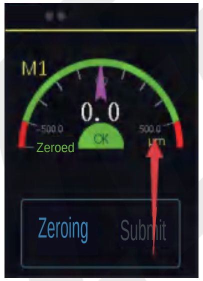


figure 13

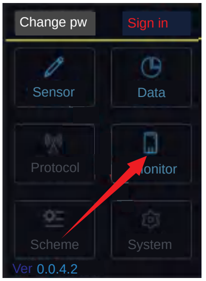
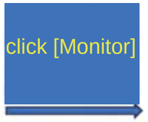
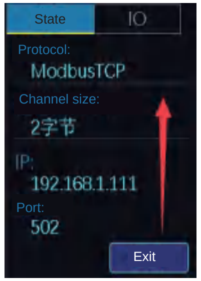


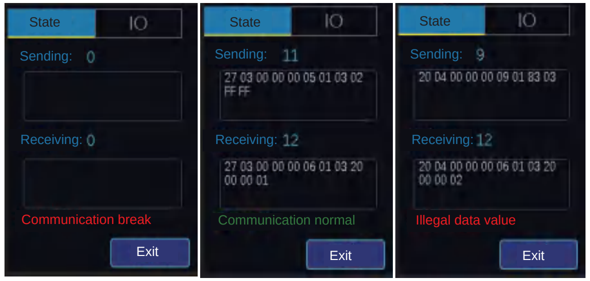
figure 14





Protocol: ModbusTCP
Channel size: 2字节
IP: 192.168.1.111
Port: 502

figure 15



Sending: 0

Receiving: 0

Communication break

Sending: 11
27 03 00 00 00 05 01 03 02 FF FF

Receiving: 12
27 03 00 00 00 06 01 03 20 00 00 01

Communication normal

Sending: 9
20 04 00 00 00 09 01 83 03

Receiving: 12
20 04 00 00 00 06 01 03 20 00 00 02

Illegal data value

figure 16

You can view the current protocol, channel value size, IP, port and other information in the monitoring interface.

The byte data of the current communication can be seen in the send and receive areas, and by comparing the normal bytes, erroneous communication can be corrected.

2.5 Programme

When several products of different sizes need to be measured, the tolerances as well as the nominal values of different sizes need to be entered frequently, which can cause a lot of trouble for the operation. Gauge provides a convenient way to do this. Gauge allows users to create multiple scenarios (up to 10), each of which can be set up with different tolerances, formulas, measurements, presentations etc.

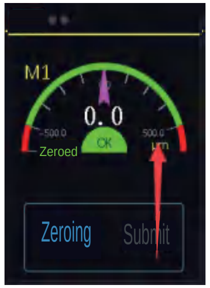


figure 17

Touch any position and swipe upward

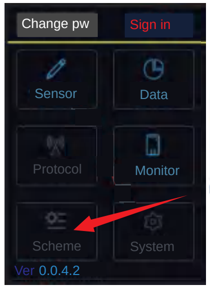


figure 18

click [Scheme]

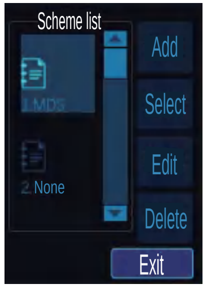
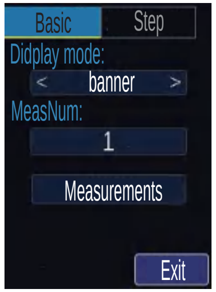


figure 19

The following describes the [Add], [Select], [Edit], and [Delete] programme operations respectively. [Add], [Edit], and [Delete] require login privileges to operate.

2.5.1 Add programme

Click the [Add] button, in the pop-up dialogue box, enter the name of the added scheme, be careful not to rename it with the previous one. Click to confirm, click [Edit] will appear in the settings interface, as shown in the figure below:



Didplay mode: < banner >

MeasNum: 1

figure 20

2.5.1.1 Basic parameters include: display mode, measured value display, number of measurement items

[Display mode]: There are 2 options: dial, banner. 2 measurements or less can be selected for dial display.

[Measured value display]: Relative values, absolute values

[Number of measurement items]: Up to 8 measurements can be entered

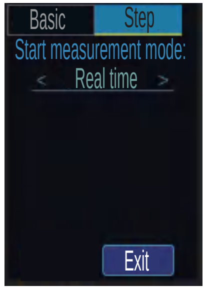
Meaning of measurement items: Measurable items, 1 measurement item can correspond to 1 sensor or multiple sensors. For example, the value of M1 can be directly equal to T1, or can be set to find the maximum value of T1, or to find the sum of two sensors T1+T2. The formulas correspond to: m1 = t1, m1 = max(t1), m1 = t1+t2, so that it can be easily configured for different measurement requirements.

2.5.1.2 Measurement parameters include: Measurement mode, Measurement end mode

Measurement parameters are mainly used to set up different measurement modes. The gauge supports four modes: real-time measurement, sensor-triggered measurement, external IO-triggered measurement, and command-triggered measurement. The following are the specific meanings of these measurement methods:

2.5.1.2.1 real-time measurement

Measuring in real time, i.e. at all times, whether the sensor is installed or not, or whether the workpiece is in place or not. The advantage of this approach is that the current value can be viewed at any time, making it easier to install and commission. The disadvantages are that it is not possible to know when the measured value is real and it cannot be used for dynamic measurements.



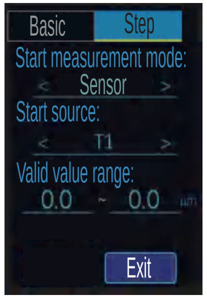
Start measurement mode: < Real time >

figure 21

2.5.1.2.2 Sensor Triggered Measurement

The sensor is triggered and the gauge can start measuring by automatically recognising whether the workpiece is pressed on or not.

It is necessary to set the effective range of the sensor used, which is used to recognise whether the workpiece is pressed on or not.



Start measurement mode: < Sensor >

Start source: < T1 >

Valid value range: 0.0 ~ 0.0 um

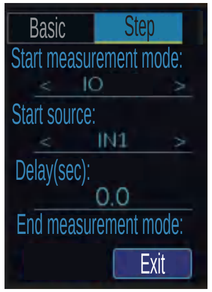
figure 22

For example, select T1 for trigger measurement and set the RMS range -0.8mm to 0.8mm. When the workpiece is not pressed on, the sensor value has to be outside the range, and when the workpiece is pressed on, the sensor value will be within the RMS range so that the gauge will automatically start measuring.

To ensure accurate measurements, the start-up delay must be set, depending on how much time is set.

2.5.1.2.3 External IO Trigger Measurement

Triggered by external IO, the gauge measurement can be initiated by external devices such as PLC, pushbuttons, proximity switches, and so on. This is a human-controlled method and is more suitable for use on manual measuring stations.


Basic | Step

Start measurement mode: < IO >

Start source: < IN1 >

Delay(sec): 0.0

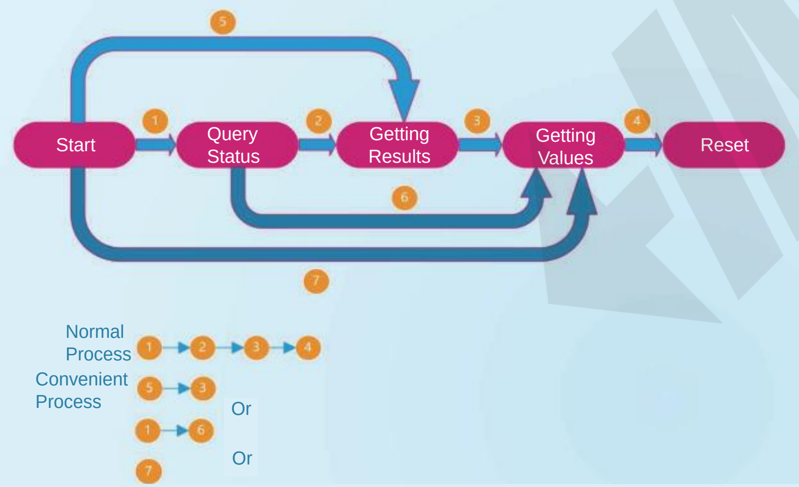
End measurement mode:

Exit

figure 23

2.5.1.2.4 Command Trigger Measurement

Command triggering, i.e. the way the host computer or PLC sends Modbus commands for measurement. The gauge system provides a complete set of measurement-related commands, including: start measurement, measurement status acquisition, measurement result acquisition, reset measurement and so on.



The simplest way to get the measured value is 7 steps, after starting the measurement, delay a specific time, directly read the measured value, provided that you must know precisely how long the gauge can be executed after starting the measurement. For example, if the current setup is 2 seconds, then after he PLC or host computer sends out the startup command, it will delay at least 2.1 seconds to read the measured value, and the value read at this time is the real current measured value.

2.5.1.2.5 Measurement end mode

Once the measurement is started, you have to face how to end it, and the gauge provides 3 modes to end it: time mode, IO mode, and command mode.

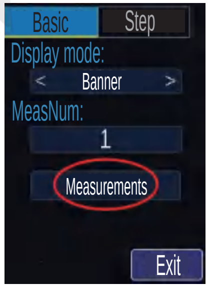
Time mode: Set a specific time, such as 1 second, then after starting the measurement, the system will measure for 1 second, then end the measurement and output the result.

IO mode: As with the IO start method, the measurement is terminated by an external signal. Generally used in conjunction with IO start.

Command mode: Ends the measurement with a command, as with command start. Generally used in conjunction with command start.

2.5.1.3 Detailed settings for measurement items

The tolerances, formulas and coded outputs of the measurement items are set here in the measurement items of the programme.


Basic | Step

Display mode: < Banner >

MeasNum: 1

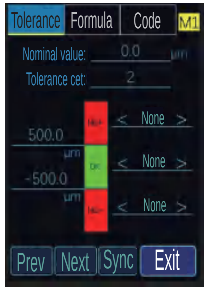
Measurements

Exit

figure 24

Click the [Measurements] button to enter the measurement item detailed configuration menu.

2.5.1.3.1 Tolerance


Tolerance | Formula | Code | M1

Nominal value: 0.0 μm

Tolerance cet: 2

NG+ 500.0 μm < None >

OK -500.0 μm < None >

NG- < None >

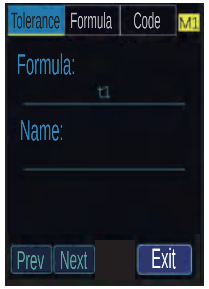
Prev | Next | Sync | Exit

figure 25

Tolerance setting is basically the same as chapter 2.2, you can set the output IO corresponding to each region, for example, [NG+] corresponds to [OUT1], [OK] corresponds to [OUT2], [NG-] corresponds to [OUT3].

2.5.1.3.2 Formula

Switch to the Formula screen to modify the formula of the current measurement item. You can also enter the name of the measurement item in this screen.



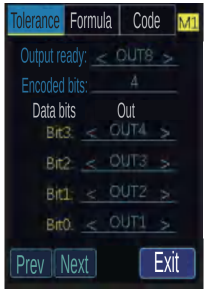
Formula: t1

Name:

figure 26

2.5.1.3.3 Code

Gauges can output values via IO encoding for easy direct access to the machine tool.



Output ready: < OUT8 >

Encoded bits: 4

Data bits Out

Bit3: < OUT4 >

Bit2: < OUT3 >

Bit1: < OUT2 >

Bit0: < OUT1 >

figure 27

The number of encoding bits is set to 4 in Figure 27. and BIT0 to BIT3. the format is defined as follows:

Symbol Bits	Data Bits	Data Bits	Data Bits
BIT3	BIT2	BIT1	BIT0
OUT4	OUT3	OUT2	OUT1

Regardless of the number of encoding bits set, the highest bit remains as the sign bit. So with 4-bit encoding, the maximum data that can be represented is: 2^(4 - 1) = ±8 in microns.

The process of parsing is explained below with two examples:

Example 1：

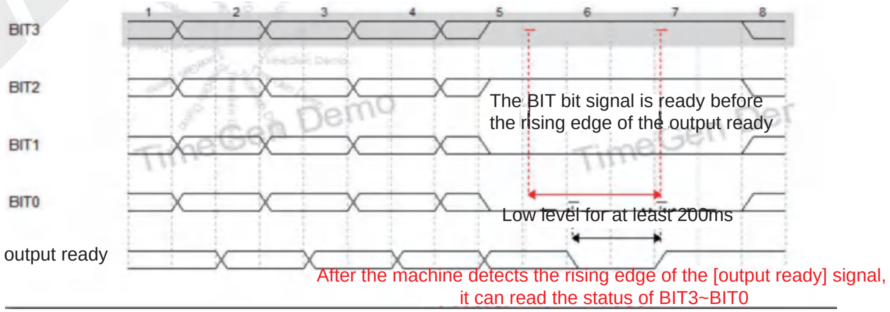
Symbol Bits	Data Bits	Data Bits	Data Bits
BIT3	BIT2	BIT1	BIT0
OUT4	OUT3	OUT2	OUT1
0	1	0	1

It can be seen that the sign bit is 0, indicating that the value is positive; the data bits are 101 from high to low, converted to decimal 5; the overall result is 5μm.

Example 2：

Symbol Bits	Data Bits	Data Bits	Data Bits
BIT3	BIT2	BIT1	BIT0
OUT4	OUT3	OUT2	OUT1
1	0	1	1

It can be seen that the sign bit is 1, indicating that the value is negative; the data bits from high to bottom are 011, which is converted to 3 in decimal; the overall result is -3μm.



The BIT bit signal is ready before the rising edge of the output ready

Low level for at least 200ms

After the machine detects the rising edge of the [output ready] signal, it can read the status of BIT3~BIT0

2.5.2 Options

In the scheme list, click the icon of the scheme you want to select, and then click the [Select] button on the right side to select the scheme. In addition the gauge supports switching schemes by means of modbus commands.

2.5.3 Editorial programme

Only the selected programme can be edited, and the editing is the same as adding a programme.

2.5.4 Delete programme

In the list of schemes, click the icon of the scheme you want to select, and then click the [Delete] button on the right to delete the scheme. The currently selected programme cannot be deleted.

2.6 Communication settings

The instrument can interact with external devices through serial port or network port, the protocols of serial port include: modbus-rtu, char-link protocol; the protocol of network port is mainly modbus-tcp. Through this interface, you can switch the protocol suitable for accessing your own device.

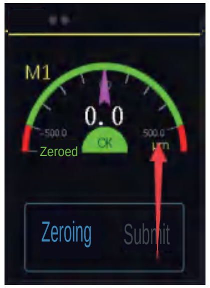


figure 28

Touch any position and swipe upward

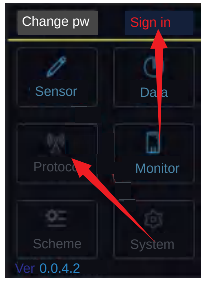
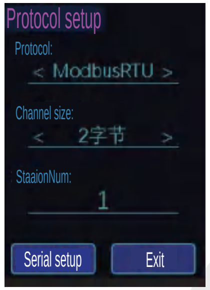


figure 29

Click on the newsletter to sign in


Protocol setup

Protocol: < ModbusRTU >

Channel size: < 2字节 >

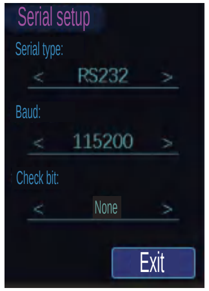
StaaionNum: 1

figure 30

When using Modbus protocol (both RTU and TCP), it is necessary to confirm the size of the channel value, that is, how many registers are required to represent the channel value. Typically, one Modbus register is 16 bits (2 bytes), and the maximum value it can represent is ±2^15, ranging rom -32767 to 32767. When converted to a representation format with 0.1μm precision, the maximum displacement values that can be represented are -3276.7μm and 3276.7μm. Therefore, if the total measurement range of the sensor exceeds this limit, two registers (4 bytes) should be used to represent one channel value. Users need to select the appropriate size according to the measurement range of their own sensors.

After selecting the protocol, it is necessary to configure the interface information, including serial port or network port.

2.6.1 Serial port settings


Serial setup

Serial type: < RS232 >

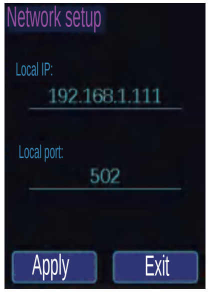
Baud: < 115200 >

Check bit: < None >

figure 31

The serial port settings include the serial port type, baud rate, and parity bit, which can be configured according to the user's own requirements.

2.6.2 Network Port Setting


Network setup

Local IP: 192.168.1.111

Local port: 502

figure 32

Gauge in wired network communication, if the protocol is ModbusTCP, the gauge is as a TCP server, need to connect to the gauge of the device as a TCP client; if the protocol is json, the gauge is as a TCP client, the device to receive json data to be a TCP server.

Static IP is generally used when the gauge is directly connected to a switch, computer, or PLC, the default configuration is as follows:

Gauge IP	Gauge Port
192.168.1.111	502

After changing the gauge IP and port, you have to click the [Apply] button.

2.7 Measurement data

The Gauge system provides measurement data storage, although storage to an external USB stick is only supported.

First of all, provide a USB memory stick that supports FAT32, FAT16, FAT12, exFAT and other formats. Insert the USB flash drive into the USB port of the gauge, at this time, the main interface will have a prompt for USB insertion.

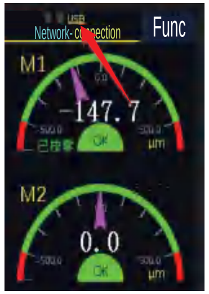


figure 33

At this point, you can click the [Submit] button on the main screen to store the data, or submit the data automatically after the measurement is completed. The specific submission method can be changed in [System Settings].

You can view the currently submitted data in the [Data] interface as shown below:

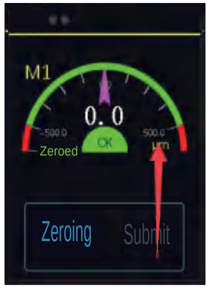


figure 34

Touch any position and swipe upward

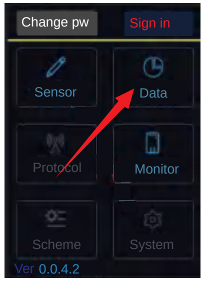
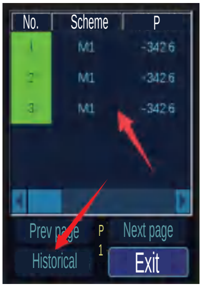


figure 35

Click [data]



No.	Scheme	P
1	M1	-342.6
2	M1	-342.6
3	M1	-342.6

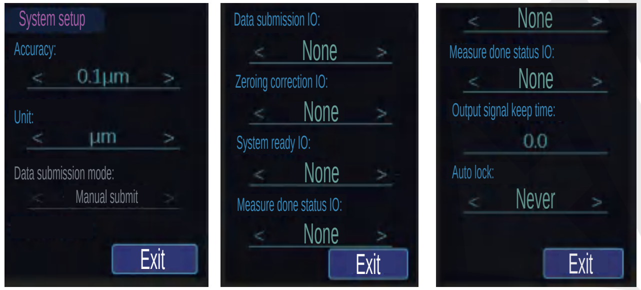
figure 36

Figure 36 shows the data list, you can click on the bottom [historical data] to view the data stored in the USB memory stick, you can only view the latest 100 data in the gauge, if you want to view all the stored data, you need to open the storage file on the computer to view.

The data in the list can also be deleted artificially by simply clicking on the row to be deleted and waiting for the [Delete] button to be displayed and then clicking on that button.

2.8 System settings

The system settings mainly include the following settable items:



System setup
Accuracy: < 0.1μm >
Unit: < μm >
Data submission mode: < Manual submit >

Data submission IO: < None >
Zeroing correction IO: < None >
System ready IO: < None >
Measure done status IO: < None >

< None >
Measure done status IO: < None >
Output signal keep time: 0.0
Auto lock: < Never >

figure 37

The interface can be swiped up and down to view the different settings, the following are described in these configurations

-- Resolution
The resolution of the gauge can be switched： 0.1μm and 1μm.

-- Units
There are 3 units available: μm, mm, si

-- Submit data
There are 3 units available: μm, mm, si

-- Submit data
To submit measurement data, you can choose between manual submission and automatic submission. Manual submission can be done by clicking the [Submit] button on the main measurement interface and external IO. Automatic submission] can only be selected if the measurement mode is non-real-time mode. When the measurement is completed, the system will automatically submit the current measurement data.

-- Submit data IO
This IO can only be set when [Manual Submission] data is selected, and a signal is provided through this IO to trigger the submission of data.

-- Calibration zero IO
Configuration of the calibration IO allows the calibration to be triggered by an external push button or proximity switch.

-- System Ready IO
This IO is used to identify whether the system is ready or not. If the system is ready, it will output a signal, otherwise it will cut off the signal. This IO is mainly used to access automated machines. (This output signal is not affected by the output hold time and will always be output)

-- Measurement completion IO
When the measurement is completed, this IO will output a signal, which can be used by the external device to determine whether to start reading the measurement data or to start the next measurement cycle.

-- Output Hold Time
The hold time of the output signal, for example, if it is set to 3 seconds, the output signal will automatically cut off the signal after 3 seconds after there is an output. A setting of 0 means that the signal will be held until the next cycle.

-- rest time
Depending on the set time, the gauge screen will automatically go out, and when it does, the screen can be made to light up again by touching the screen.

2.9 Multi-tolerance settings

Gauges offer several ways of using tolerance judgements, such as the rendering below:

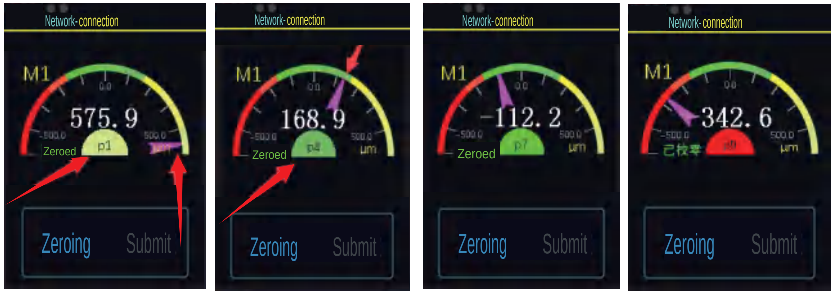


figure 38

You can see in the picture that the dial pointer is pointing to a different position, and the corresponding colour and text will change simultaneously, so that you can easily understand the current size range of the workpiece.

The specific setup steps are as follows: first enter into the tolerance interface.

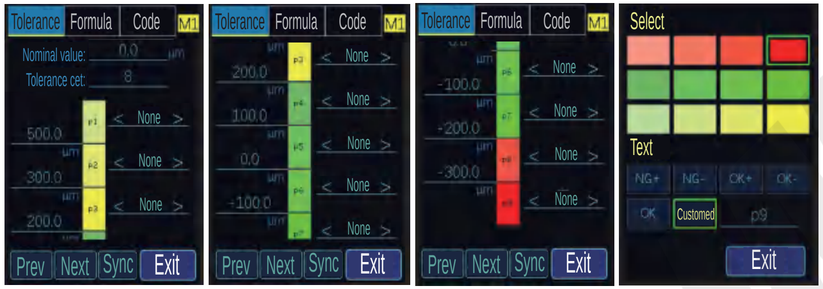


figure 39

First set the number of points to judge, say 8, then there are 9 judgeable intervals. You can see all the intervals by scrolling up and down.

Left edit field: You can set the limit value for each judgement point, from top to bottom, the value decreases in order.

Right edit field: You can select the output channel of the corresponding area in the same way as the normal tolerance setting.

Middle Colour Bar: By clicking the button area of P1~P9 in the picture, you can enter the colour and text selection interface.
You can choose the corresponding colour and text, the text can be customized, custom text length of up to 5 characters.

3. COMMUNICATION DESCRIPTION

3.1 MODBUS Communication

The gauge supports MODBUS-RUT and MODBUS-TCP, where RTU is with serial port and TCP with network port.

MODBUS- TCP is very similar to the serial link MODBUS- RTU protocol, the two are the same in that the application data unit is the same, the difference is that MODBUS TCP is transported on a TCP/IP network, with one more message header (MBAP), less CRC checksums, and the use of TCP's port 502; RTU has more device addresses and CRC checksums.

The gauge supports 3 function codes: 0×03, 0×06 and 0×10.

0×03	0×06	0×10
Read Holding Register	Write Single Holding Register	Write Multiple Holding Registers

Note that in the following text, [master] all denotes modbus master, and [slave] all denotes modbus slave.

3.1.1 MODBUS communication

Description	Register address	Contents	Privileges	Function code
Sensor data	0x2000	Current sensor value, values in descending order from 0x2000	Read-only	0x03
Measurement item data	0x4000	The value of the current measurement, the values are listed in descending order from 0x4000	Read-only	0x03
Sensor related	0xC000	Zeroing address, currently supports 4 sensors for zeroing	Read-write	0x03, 0x06, 0x10
	0x0C40	Zero cancellation address	Read-write	0x03, 0x06, 0x10
	0x0C80	Inverse address, currently supports 4 sensors for inversion	Read-write	0x03, 0x06, 0x10
	0x0C20	Sensor address calibration, currently supports 4 sensors for calibration Write 0x0001 to read standard 1 value Write 0x0002 to read standard 2 value Write 0x0003 to confirm calibration Write 0x0004 to reset calibration Write 0x0005 to synchronise calibration coefficients to other sensors	Write-only	0x06, 0x10
Measurement item related	0x0B00	Setting the measurement mode, currently supports 8 measurement items Write 0x0001 to start measurement Write 0x0002 to end measurement Write 0x0003 to reset measurement status	Write-only	0x06, 0x10
	0x0B20	Measurement status acquisition, currently supports 8 measurement items	Read-only	0x03
	0x0B40	Results acquisition for measurement items, currently 8 measurement items are supported.	Read-only	0x03
	0x0B20	Calibration, currently supports 8 measurements	Read-write	0x03, 0x06, 0x10
Programme options	0x0B80	Selection options, writable values between 1 and 10	Read-write	0x03, 0x06, 0x10

3.1.2 MODBUS Error Protocol Package

MODBUS-RTU			
Number	Exception function code	Error code	CRC
1	function code+0x80	0x01~0x08	2 bytes

MODBUS-TCP					
Transaction identification	Protocol identifier	Length	Unit identifier	Exception function code	Error code
2 bytes	2 bytes	2 bytes	1 byte	function code+0x80	0x01~0x08

Error Code Definition:

Error code	Meaning
0x01	Illegal function codes, i.e. the system does not support the function code
0x02	Illegal data address, such as the current read 4 channel data, the legal address is 0x2000~0x2004, beyond the address, then return data address error
0x03	Illegal data values, such as M1 zeroing, return an 03 error code, which may mean that the current zeroing has failed.
0x08	CRC Checksum Error

3.1.3 MODBUS Command

· RTU data acquisition command——0×2000、0×4000

MODBUS-RTU instruction 0x03 Function code request format Fixed 8 bytes									
	Descriptions		Code	Register address		Number of registers		CRC	
	Byte position	1	2	3	4	5	6	7	8
	Note	No.	Read holding register	High byte	Low byte	High byte	Low byte	Low byte	High byte
Channel value size 2 bytes	Get T1 sensor value	1	03	20	00	00	01	8F	CA
	Get M1 measurement item value	1	03	40	00	00	01	91	CA
	Get T2 sensor value	1	03	20	01	00	01	DE	0A
	Get M2 measurement item value	1	03	40	01	00	01	C0	0A
	Get T1, T2 sensor values	1	03	20	00	00	02	CF	CB
	Get M1, M2 measurement item values	1	03	40	00	00	02	D1	CB
Channel value size 4 bytes	Get T1 sensor value	1	03	20	00	00	02	CF	CB
	Get M1 measurement item value	1	03	40	00	00	02	D1	CB
	Get T2 sensor value	1	03	20	02	00	02	6E	0B
	Get M2 measurement item value	1	03	40	02	00	02	70	0B
	Get T1, T2 sensor values	1	03	20	00	00	04	4F	C9
	Get M1, M2 measurement item values	1	03	40	00	00	04	51	C9

If you want to get the values of other channels or more channels and more measurements, look carefully at the differences in the table to find out the pattern (the main change is the register address and the number of registers) and recalculate the last 2 CRC bytes, can calculate the CRC online via this link, pay attention to the order of the high and low bytes of the CRC.

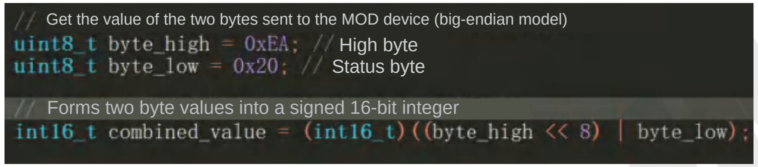
Register address	Description
0x2000	Sensor value base address
0x4000	Measured item value base address

MODBUS-RTU Instruction 0X03 Function Code Return Format										
	Descriptions		Code	Bytes	Register 1		Register 2		CRC	
	Byte position	1	2	3	4	5	6	7		
	Note	No.	Read holding register	Register byte	High byte	Low byte	High byte	Low byte	Low byte	High byte
Channel value size 2 bytes	T1 value returned	1	03	02	EA	20			F6	FC
	T1, T2 values returned	1	03	04	EA	20	0B	22	48	C8
Channel value size 4 bytes	T1 value returned	1	03	04	FF	F7	74	80	5D	75

The register value in the returned byte is the channel value.

When the size of the channel value is 2 bytes, the 2 bytes of the brown area register 1 are the value of channel T1, EA20, EA is the high byte and 20 is the low byte. Converted to signed decimal is: -5600, converted to displacement value need to divide by 10, get -560.0μm; light brown area register 2 of the 2 bytes for the value of the channel T2, 0B22, after the same conversion method, get 285.0μm.

The code reference is as follows:

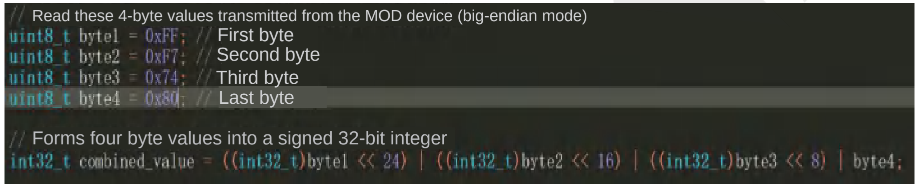


```
// Get the value of the two bytes sent to the MOD device (big-endian model)
uint8_t byte_high = 0xEA; // High byte
uint8_t byte_low = 0x20; // Status byte

// Forms two byte values into a signed 16-bit integer
int16_t combined_value = (int16_t)((byte_high << 8) | byte_low);
```

When the channel value size is 4 bytes, the 4 bytes of brown area registers 1 and 2 are the value of channel T1, FFFF77480, at from high to low. Conversion to signed decimal is: -560000, and conversion to a displacement value requires dividing by 1000 to get -560.0μm.

The code reference is as follows:



```
// Read these 4-byte values transmitted from the MOD device (big-endian mode)
uint8_t byte1 = 0xFF; // First byte
uint8_t byte2 = 0xF7; // Second byte
uint8_t byte3 = 0x74; // Third byte
uint8_t byte4 = 0x80; // Last byte

// Forms four byte values into a signed 32-bit integer
int32_t combined_value = ((int32_t)byte1 << 24) | ((int32_t)byte2 << 16) | ((int32_t)byte3 << 8) | byte4;
```

· RTUset state command——0×0B00

Descriptions		Code	register address		register value		CRC	
Byte position	1	2	3	4	5	6	7	8
Note	No.	Write single holding register	High byte	Low byte	High byte	Low byte	Low byte	High byte
Start measurement	1	06	0B	00	00	01	4A	2E
End measurement	1	06	0B	00	00	02	0A	2F
Reset measurement	1	06	0B	00	00	03	CB	EF

The Modbus slave returns the command, and the return command can be left unprocessed.

If the returned instruction is an error protocol packet, you need to locate the cause of the error based on the code in the error protocol packet.

To recalculate the CRC after changing any register address and register value, you can calculate the CRC online via this link, noting the order of the high and low bytes of the CRC.

Measurement Status Code Definition：

Register Value	Description
0x0001	Start measurement
0x0002	End measurement
0x0003	Reset measurement

· RTU Get Measurement Status Command——0×0B20

MODBUS-RTU Instruction 0X03 Function Code Request Format								
Descriptions		Code	register address		Number of registers		CRC	
Byte position	1	2	3	4	5	6	7	8
Note	No.	Read holding register	High byte	Low byte	High byte	Low byte	Low byte	High byte
Get measurement status	1	03	0B	20	00	01	87	E4

The measurement states are defined as follows：

Code	Meaning
00 00	Free state
00 01	Testing
00 02	Indicates completion of measurement
00 03	Measurement prohibition

In the above example, the returned register value is 0×0002, so it means that the current measurement is completed.

· RTU Get Measurement Result Command——0×0B40

MODBUS-RTU Instruction 0X03 Function Code Request Format								
Descriptions		Code	register address		Number of registers		CRC	
Byte position	1	2	3	4	5	6	7	8
Note	No.	Read holding register	High byte	Low byte	High byte	Low byte	Low byte	High byte
Get M1 results	1	03	0B	40	00	01	87	FA

Getting the results of the other measurements is simply a matter of changing the number of registers and register addresses, while recalculating the CRC.

MODBUS-RTU Instruction 0X03 Function Code Return Format							
Descriptions		Code	Byte	register value		CRC	
Byte position	1	2	3	4	5	6	7
Note	No.	Read holding register	Register byte	High byte	Low byte	Low byte	High byte
Measurement results return	1	03	02	00	02	39	85

In the above example, the returned register value is 0×0002, so it indicates the result of the 3rd tolerance area.

The result value is just a number, the exact meaning needs to be judged according to the tolerance interval set.

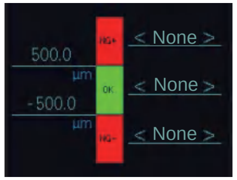


figure 40

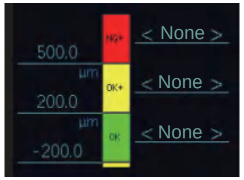


figure 41

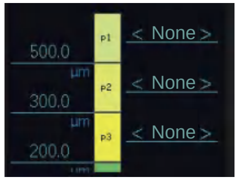


figure 42

Tolerance setting of Fig. 40, result 0002 means NG-

Tolerance setting of Fig. 40, result 0002 means OK.

Tolerance setting of Fig. 40, result 0002 means P3.

· RTU Zero command——0x0B60

MODBUS-RTU Instruction 0X03 Function Code Request Format Fixed 8 Bytes								
Descriptions		Code	register address		register value		CRC	
Byte position	1	2	3	4	5	6	7	8
Note	No.	Write single holding register	High byte	Low byte	High byte	Low byte	Low byte	High byte
M1 Zeroing	1	06	0B	60	00	01	4A	30
M1, M2 zeroing at the same time	1	06	0B	60	00	03	CB	F1
Simultaneous zeroing of 8 measurements	1	06	0B	60	00	FF	CB	B0

A register has 16 bits, each bit represents a measurement item, the corresponding bit writes 1 to indicate that the measurement item is zero-corrected, and the address from low to high corresponds to M1~M8 respectively.

Observe the change rule of register address and register value of M1 and M2 correction zero items, and you can perform correction zero operation for other measurement items.

The Modbus slave returns the command, and the return command can be left unprocessed.

If the returned instruction is an error protocol packet, you need to locate the cause of the error based on the code in the error protocol packet.

To recalculate the CRC after changing any register address and register value, you can calculate the CRC online via this link, paying attention to the order of the high and low bytes of the CRC. If the returned instruction is an error protocol packet, you need to locate the cause of the error according to the code in the error protocol packet.

· RTU Get Zero Status Command——0×0B60

MODBUS-RTU Instruction 0X03 Function Code Request Format								
Descriptions		Code	register address		Number of registers		CRC	
Byte position	1	2	3	4	5	6	7	8
Note	No.	Read holding register	High byte	Low byte	High byte	Low byte	Low byte	High byte
Get zero status	1	03	0B	60	00	01	C6	31

The gauge supports 8 measurement items, and 1 register can represent the calibration zero status of all measurement items, so the command to get the calibration zero status can be fixed as the example in the table above.

MODBUS-RTU Instruction 0X03 Function Code Return Format							
Descriptions		Code	Byte	register value		CRC	
Byte position	1	2	3	4	5	6	7
Note	No.	Read holding register	Register byte	High byte	Low byte	Low byte	High byte
Zero state return	1	03	02	00	01	79	84

The register value returned by this instruction is 0×0001, for the binary: 00000000 00000001 indicates that M1 is zero-calibrated; all other measurement items are not zero-calibrated.

A register has 16 bits, each representing a measurement item, the corresponding bit is 1 indicating that the measurement item has been zero-calibrated, and 0 indicating that it has not been zero-calibrated, and the address corresponds to M1~M8 from low to high, respectively.

· RTU Switching Programme Instructions——0×0B80

MODBUS-RTU Instruction 0X03 Function Code Request Format Fixed 8 Bytes								
Descriptions		Code	register address		register value		CRC	
Byte position	1	2	3	4	5	6	7	8
Note	No.	Write single holding register	High byte	Low byte	High byte	Low byte	Low byte	High byte
Switching programme	1	06	0B	80	00	01	4B	C6

The register value is set in the range of 1 to 10, representing the serial number of the programme.

This instruction returns an error protocol packet if there is no actual programme for the corresponding serial number. The register value is set in the range of 1 to 10, representing the serial number of the programme.

Read this register address to get the programme number of the current application, starting from 1.

To recalculate the CRC after changing any register address and register value, you can calculate the CRC online via this link, noting the order of the high and low bytes of the CRC.

· RTU Sensor Zeroing Command——0×0C00、0×0C40

MODBUS-RTU Instruction 0X03 Function Code Request Format Fixed 8 Bytes								
Descriptions		Code	register address		register value		CRC	
Byte position	1	2	3	4	5	6	7	8
Note	No.	Write single holding register	High byte	Low byte	High byte	Low byte	Low byte	High byte
T1 Zeroing	1	06	0C	00	00	01	4B	5A
T1 cancels zero setting	1	06	0C	40	00	01	4A	8E

A register has 16 bits, each bit represents a sensor, the corresponding bit writes 1 to indicate that the sensor is zeroed, the address from low to high corresponds to T1~T4 respectively.

The return instruction of this instruction can be left unprocessed.

If the returned instruction is an error protocol packet, you need to locate the cause of the error based on the code in the error protocol packet.

To recalculate the CRC after changing any register address and register value, you can calculate the CRC online via this link, noting the order of the high and low bytes of the CRC.

Register address	Descriptions
0x0C00	Zeroing address
0x0C40	Cancellation of zeroing address

· RTU Sensor Inversion Command——0×0C80

MODBUS-RTU Instruction 0X03 Function Code Request Format Fixed 8 Bytes

Descriptions		Code	register address		register value		CRC	
Byte position	1	2	3	4	5	6	7	8
Note	No.	Write single holding register	High byte	Low byte	High byte	Low byte	Low byte	High byte
T1 inverse	1	06	0C	80	00	01	4A	B2
T1 cancels the inverse	1	06	0C	80	00	00	8B	72

A register has 16 bits, each bit represents a sensor, the corresponding bit write 1 means to invert the sensor, the bit write 0 means to cancel the inverted state of the sensor, the address from low to high corresponds to T1~T4 respectively.

The return instruction of this instruction can be left unprocessed.

If the returned instruction is an error protocol packet, you need to locate the cause of the error based on the code in the error protocol packet.

To recalculate the CRC after changing any register address and register value, you can calculate the CRC online via this link, noting the order of the high and low bytes of the CRC.

· RTU Sensor Calibration Instructions——0×0C20

MODBUS-RTU Instruction 0X06 Function Code Request Format Fixed 8 Bytes

Descriptions		Code	register address		register value		CRC	
Byte position	1	2	3	4	5	6	7	8
Note	No.	Write single holding register	High byte	Low byte	High byte	Low byte	Low byte	High byte
T1 reads the value of standard 1	1	06	0C	20	00	01	4A	90
T2 reads the value of standard 2	1	06	0C	20	00	02	0A	91
T1 calibration	1	06	0C	20	00	03	CB	51
T1 reset calibration	1	06	0C	20	00	04	8A	93
T1 synchronisation	1	06	0C	20	00	05	4B	53

The return instruction of this instruction can be left unprocessed.

If the returned command is an error packet, it is necessary to locate the cause of the error based on the code in the error packet. Generally the calibration fails because the values of the large and small standards are the same, or the hardware circuit is faulty.

Calibrating sensors for other channels is simply a matter of changing the register address.

To recalculate the CRC after changing any register address and register value, you can calculate the CRC online via this link, noting the order of the high and low bytes of the CRC.

Sensor calibration code definition:

register value	Descriptions
0x0001	Read Standard 1 value
0x0002	Read Standard 2 value
0x0003	Calibration
0x0004	Reset
0x0005	Synchronisation to other sensors

· TCP Data Acquisition Command——0×2000、0×4000

MODBUS-TCP Instruction 0X03 Function Code Request Format Fixed 12 Bytes

	Descriptions	Transaction identifier	Protocol identifier	Length	Unit identifier	Code	Register address		Number of registers	
	Byte position	1、2	3、4	5、6	7	8	9	10	11	12
	Note	host-initiated slave-replicated	modbus fixed to 0	Next byte to the end	host-initiated slave-replicated	Read holding register	High byte	Low byte	High byte	Low byte
Channel value size 2 bytes	Get T1 value	0x9776	0x0000	0x0006	0x04	03	20	00	00	01
	Get M1 value	0x9776	0x0000	0x0006	0x04	03	40	00	00	01
	Get T2 value	0x9776	0x0000	0x0006	0x04	03	20	01	00	01
	Get M2 value	0x9776	0x0000	0x0006	0x04	03	40	01	00	01
	Get T1,T2 values	0x9776	0x0000	0x0006	0x04	03	20	00	00	02
	Get M1,M2 values	0x9776	0x0000	0x0006	0x04	03	40	00	00	02
Channel value size 4 bytes	Get T1 value	0x9776	0x0000	0x0006	0x04	03	20	00	00	02
	Get M1 value	0x9776	0x0000	0x0006	0x04	03	40	00	00	02
	Get T2 value	0x9776	0x0000	0x0006	0x04	03	20	02	00	02
	Get M2 value	0x9776	0x0000	0x0006	0x04	03	40	02	00	02
	Get T1,T2 values	0x9776	0x0000	0x0006	0x04	03	20	00	00	04
	Get M1,M2 values	0x9776	0x0000	0x0006	0x04	03	40	00	00	04

If you want to get the values of other channels or more channel values and more measurement items, look carefully at the differences in the table to find a pattern (the main changes are the register address and the number of registers).

In the table, the transaction identifier 0×9776 and the unit identifier 0×04 are only examples, and the actual application is based on the host's own MAP message.

Register address	Descriptions
0x2000	Sensor value base address
0x4000	Measured item value base address

MODBUS-TCP Instruction 0X03 Function Code Return Format

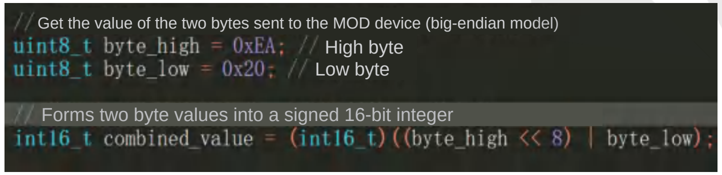
	Descriptions	Transaction identifier	Protocol identifier	Length	Unit identifier	Code	Register byte	Register 1		Register 2	
	Byte position	1、2	3、4	5、6	7	8	9	10	11	12	13
	Note	Copying data from the request frame	modbus fixed to 0	Next byte to the end	Copying data from the request frame	Read holding register	Register byte	High byte	Low byte		
Channel value size 2 bytes	Get T1 value	0x9776	0x0000	0x0005	0x04	03	02	EA	20		
	Get T1,t2 values	0x9776	0x0000	0x0007	0x04	03	04	EA	20	0B	22
Channel value size 4 bytes	Get T1 value	0x9776	0x0000	0x0007	0x04	03	04	FF	F7	74	80

Note how the lengths 0×0005 and 0×0007 are obtained in the table, from the unit identifier byte all the way down to the number of the last 1 byte.

The register value in the returned byte is the channel value.

When the size of the channel value is 2 bytes, the 2 bytes of the brown area register 1 are the value of channel T1, EA20, EA is the high byte and 20 is the low byte. Converted to signed decimal is: -5600, converted to displacement value need to divide by 10, get -560.0μm; light brown area register 2 of the 2 bytes for the value of the channel T2, 0B22, after the same conversion method, get 285.0μm.

The calculation code is referenced below:

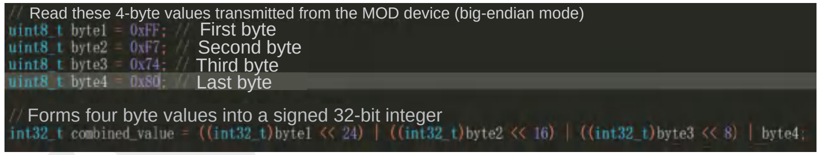


```
// Get the value of the two bytes sent to the MOD device (big-endian model)
uint8_t byte_high = 0xEA; // High byte
uint8_t byte_low = 0x20; // Low byte

// Forms two byte values into a signed 16-bit integer
int16_t combined_value = (int16_t)((byte_high << 8) | byte_low);
```

When the channel value size is 4 bytes, the 4 bytes of brownfield registers 1 and 2 are the value of channel T1, FFFF77480, addressed from high to low. Conversion to signed decimal is: -560000, and conversion to a displacement value requires dividing by 1000 to get -560.0μm.

The calculation code is referenced below:



```
// Read these 4-byte values transmitted from the MOD device (big-endian mode)
uint8_t byte1 = 0xFF; // First byte
uint8_t byte2 = 0xF7; // Second byte
uint8_t byte3 = 0x74; // Third byte
uint8_t byte4 = 0x80; // Last byte

// Forms four byte values into a signed 32-bit integer
int32_t combined_value = ((int32_t)byte1 << 24) | ((int32_t)byte2 << 16) | ((int32_t)byte3 << 8) | byte4;
```

· TCP Set Measurement Status Command——0×0B00

MODBUS-TCP Instruction 0X06 Function Code Request Format Fixed 12 Bytes

Descriptions	Transaction identifier	Protocol identifier	Length	Unit identifier	Code	Register address		Register value	
Byte position	1、2	3、4	5、6	7	8	9	10	11	12
Note	host-initiated slave-replicated	modbus fixed to 0	Next byte to the end	host-initiated slave-replicated	Write single holding register	High byte	Low byte	High byte	Low byte
Start measurement	0x9776	0x0000	0x0006	0x04	06	0B	00	00	01
End measurement	0x9776	0x0000	0x0006	0x04	06	0B	00	00	02
Reset measurement	0x9776	0x0000	0x0006	0x04	06	0B	00	00	03

The Modbus slave returns the command, and the return command can be left unprocessed.

If the returned instruction is an error protocol packet, you need to locate the cause of the error based on the code in the error protocol packet.

Measurement Status Code Definition:

Register value	Descriptions
0x0001	Start measurement
0x0002	End measurement
0x0003	Reset measurement

· TCP Get Measurement Status Command——0×0B20

MODBUS-TCP Instruction 0X03 Function Code Request Format										
Descriptions	Transaction identifier	Protocol identifier	Length	Unit identifier	Code	Register address		Number of registers		
Byte position	1、2	3、4	5、6	7	8	9	10	11	12
Note	host-initiated slave-replicated	modbus fixed to 0	Next byte to the end	host-initiated slave-replicated	Read holding register	High byte	Low byte	High byte	Low byte
Get measurement status	0x9776	0x0000	0x0006	0x04	03	0B	20	00	01

MODBUS-TCP Instruction 0X03 Function Code Return Format								
Descriptions	Transaction identifier	Protocol identifier	Length	Unit identifier	Code	Register byte	Register value	
Byte position	1、2	3、4	5、6	7	8	9	10	11
Note	Copying data from the request frame	modbus fixed to 0	Next byte to the end	Copying data from the request frame	Read holding register	Register byte	High byte	Low byte
Return measurement status	0x9776	0x0000	0x0005	0x04	03	02	00	02

Note how the length 0×0005 is obtained in the table, from the unit identifier byte all the way down to the number of the last 1 byte.

The measurement states are defined as follows:

Code	Descriptions
00 00	Idle state
00 01	Testing
00 02	Testing complete
00 03	Measurement disabled

In the above example, the returned register value is 0×0002, so it means that the current measurement is completed.

· TCP Get Measurement Result Command——0×0B40

MODBUS-TCP Instruction 0X03 Function Code Request Format										
Descriptions	Transaction identifier	Protocol identifier	Length	Unit identifier	Code	Register address		Number of registers	
Byte position	1、2	3、4	5、6	7	8	9	10	11	12
Note	host-initiated slave-replicated	modbus fixed to 0	Next byte to the end	host-initiated slave-replicated	Read holding register	High byte	Low byte	High byte	Low byte
Get M1,M2 values	0x9776	0x0000	0x0006	0x04	03	0B	40	00	02

Getting the results of other measurements is simply a matter of changing the number of registers and register addresses.

MODBUS-TCP Instruction 0X03 Function Code Return Format											
Descriptions	Transaction identifier	Protocol identifier	Length	Unit identifier	Code	Register byte	Register 1		Register 2	
Byte position	1、2	3、4	5、6	7	8	9	10	11	12	13
Note	Copying data from the request frame	modbus fixed to 0	Next byte to the end	Copying data from the request frame	Read holding register	Register byte	High byte	Low byte	High byte	Low byte
M1,M2 results returned	0x9776	0x0000	0x0007	0x04	03	04	00	02	00	02

Note how the length 0x0007 is obtained in the table, from the unit identifier byte all the way down to the number of the last 1 byte.

In the above example, the returned register value is 0x0002, so it indicates the result of the 3rd tolerance region.

The result value is just a number, the exact meaning needs to be judged according to the tolerance interval set.

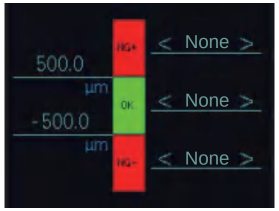


figure 43

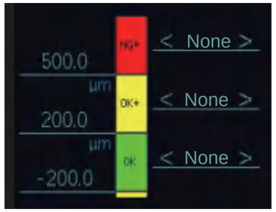


figure 44

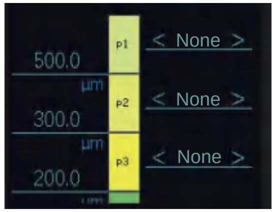


figure 45

Tolerance setting of Fig. 43, result 0002 means NG-

Tolerance setting of Fig. 44, result 0002 means OK.

Tolerance setting of Fig. 45, result 0002 means P3.

· TCP Zero command——0x0B60

MODBUS-TCP Instruction 0X06 Function Code Request Format Fixed 12 Bytes

Descriptions	Transaction identifier	Protocol identifier	Length	Unit identifier	Code	Register address		Register value	
Byte position	1、2	3、4	5、6	7	8	9	10	11	12
Note	host-initiated slave-replicated	modbus fixed to 0	Next byte to the end	host-initiated slave-replicated	Write single holding register	High byte	Low byte	High byte	Low byte
M1 zeroing	0x9776	0x0000	0x0006	0x04	06	0B	60	00	01
M1 M2 simultaneous zeroing	0x9776	0x0000	0x0006	0x04	06	0B	60	00	03
Simultaneous zeroing of 8 measurement	0x9776	0x0000	0x0006	0x04	06	0B	60	00	FF

A register has 16 bits, each bit represents a measurement item, the corresponding bit writes 1 to indicate that the measurement item is zero-corrected, and the address from low to high corresponds to M1~M8 respectively.

Observe the change rule of register address and register value of M1 and M2 correction zero items, and you can perform correction zero operation on other measurement items.

The Modbus slave returns the command, and the return command can be left unprocessed.

If the returned instruction is an error protocol packet, you need to locate the cause of the error based on the code in the error protocol packet.

· TCP Get Zero Status Command——0x0B60

MODBUS-TCP Instruction 0X03 Function Code Request Format

Descriptions	Transaction identifier	Protocol identifier	Length	Unit identifier	Code	Register address		Number of registers	
Byte position	1、2	3、4	5、6	7	8	9	10	11	12
Note	host-initiated slave-replicated	modbus fixed to 0	Next byte to the end	host-initiated slave-replicated	Read holding register	High byte	Low byte	High byte	Low byte
Get Zero Status	0x9776	0x0000	0x0006	0x04	03	0B	60	00	01

The gauge supports 8 measurement items, and 1 register can represent the calibration zero status of all measurement items, so the command to get the calibration zero status can be fixed as the example in the table above.

MODBUS-TCP Instruction 0X03 Function Code Return Format

Descriptions	Transaction identifier	Protocol identifier	Length	Unit identifier	Code	Register type	Register value	
Byte position	1、2	3、4	5、6	7	8	9	10	11
Note	Copying data from the request frame	modbus fixed to 0	Next byte to the end	Copying data from the request frame	Read holding register	Register type	High byte	Low byte
Get Zero Status	0x9776	0x0000	0x0005	0x04	03	02	00	01

Note how the length 0x0005 is obtained in the table, from the unit identifier byte all the way down to the number of the last 1 byte.

This instruction returns a register value of 0x0001, for which the binary is: 00000000 00000001 indicating that M1 is zero-calibrated; all other measurement items are not zero-calibrated. A register has 16 bits, each bit represents one measurement item.

A register has 16 bits, each representing a measurement item, the corresponding bit is 1 indicating that the measurement item has been zero-calibrated, and 0 indicating that it has not been zero-calibrated, and the address corresponds to M1~M8 from low to high, respectively.

· TCP Switching programme instructions——0x0B80

MODBUS-TCP Instruction 0X06 Function Code Request Format Fixed 12 Bytes

Descriptions	Transaction identifier	Protocol identifier	Length	Unit identifier	Code	Register address		Register value	
Byte position	1、2	3、4	5、6	7	8	9	10	11	12
Note	host-initiated slave-replicated	modbus fixed to 0	Next byte to the end	host-initiated slave-replicated	Write single holding register	High byte	Low byte	High byte	Low byte
Switching programme	0x9776	0x0000	0x0006	0x04	06	0B	80	00	01

The register value is set in the range of 1 to 10, representing the serial number of the programme.

This instruction returns an error protocol packet if there is no actual programme for the corresponding serial number.

Read this register address to get the programme number of the current application, starting from 1.

· TCP Sensor Zero Command——0x0C00、0x0C40

MODBUS-TCP Instruction 0X06 Function Code Request Format Fixed 12 Bytes									
Descriptions	Transaction identifier	Protocol identifier	Length	Unit identifier	Code	Register address		Register value	
Byte position	1、2	3、4	5、6	7	8	9	10	11	12
Note	host-initiated slave-replicated	modbus fixed to 0	Next byte to the end	host-initiated slave-replicated	Write single holding register	High byte	Low byte	High byte	Low byte
T1 zeroing	0x9776	0x0000	0x0006	0x04	06	0C	00	00	01
T1, T2 zeroed simultaneously	0x9776	0x0000	0x0006	0x04	06	0C	00	00	03
Simultaneous zeroing of 4 sensors	0x9776	0x0000	0x0006	0x04	06	0C	00	00	0F
T1 cancels zeroing	0x9776	0x0000	0x0006	0x04	06	0C	40	00	01

Note how the length 0x0006 is obtained in the table, from the unit identifier byte all the way down to the number of the last 1 byte.

A register has 16 bits, each bit represents a sensor, the corresponding bit writes 1 to indicate that the sensor is zeroed, the address from low to high corresponds to T1~T4 respectively.

The return instruction of this instruction can be left unprocessed.

If the returned instruction is an error protocol packet, you need to locate the cause of the error based on the code in the error protocol packet.

Register address	Descriptions
0x0C00	Constant-zero address
0x0C40	Cancel constant-zero address

· TCP Sensor Inversion Command——0x0C80

MODBUS-TCP Instruction 0X06 Function Code Request Format Fixed 12 Bytes									
Descriptions	Transaction identifier	Protocol identifier	Length	Unit identifier	Code	Register address		Register value	
Byte position	1、2	3、4	5、6	7	8	9	10	11	12
Note	host-initiated slave-replicated	modbus fixed to 0	Next byte to the end	host-initiated slave-replicated	Write single holding register	High byte	Low byte	High byte	Low byte
T1 inverse	0x9776	0x0000	0x0006	0x04	06	0C	80	00	01
T1, T2 simultaneous inversion	0x9776	0x0000	0x0006	0x04	06	0C	80	00	03
Simultaneous inversion of 4 sensors	0x9776	0x0000	0x0006	0x04	06	0C	80	00	0F
Cancel all channel inversions	0x9776	0x0000	0x0006	0x04	06	0C	80	00	00

Note how the length 0x0006 is obtained in the table, from the unit identifier byte all the way down to the number of the last 1 byte.

A register has 16 bits, each bit represents a sensor, the corresponding bit write 1 means to invert the sensor, the bit write 0 means to cancel the inverted state of the sensor, the address from low to high corresponds to T1~T4 respectively.

The return instruction of this instruction can be left unprocessed.

If the returned instruction is an error protocol packet, you need to locate the cause of the error based on the code in the error protocol packet.

· TCP Sensor Calibration Instructions——0x0C20

Descriptions	Transaction identifier	Protocol identifier	Length	Unit identifier	Code	Register address		Number of register	Byte	Register 1		Register 2	
Byte position	1、2	3、4	5、6	7	8	9	10	11、12	13	14	15	16	17
Note	host-initiated slave-replicated	modbus fixed to 0	Next byte to the end	host-initiated slave-replicated	Write multiple holding registers	High byte	Low byte		Register type	High byte	Low byte	High byte	Low byte
T1&T2 read standard 1 value simultaneously	0x9776	0x0000	0x000B	0x04	10	0C	20	0x0002	04	00	01	00	01
T1&T2 read standard 2 value simultaneously	0x9776	0x0000	0x000B	0x04	10	0C	20	0x0002	04	00	02	00	02
T1&T2 simultaneous calibration	0x9776	0x0000	0x000B	0x04	10	0C	20	0x0002	04	00	03	00	03
T1&T2 simultaneous reset calibration	0x9776	0x0000	0x000B	0x04	10	0C	20	0x0002	04	00	04	00	04

Note how the length 0x000B is obtained in the table, from the unit identifier byte all the way down to the number of the last 1 byte.

The return instruction of this instruction can be left unprocessed.

If the returned command is an error packet, you need to locate the cause of the error based on the code in the error packet. Usually the calibration fails because the value of the large and small standards are the same, or the hardware is faulty.

Sensor calibration code definition:

Register value	Descriptions
0x0001	Read standard 1 value
0x0002	Read standard 2 value
0x0003	Calibration
0x0004	Reset
0x0005	Synchronisation to other sensors

3.2 CharLink protocol communication

CharLink is a simple character protocol for outputting channel values or measured values directly as text. The format is as follows:

Multiple measured value：

CH1	TAB	CH2	TAB	CH3	TAB	CH4	CR LF

Each measurement is separated from the others by the TAB key, and the terminator ends with a carriage return line feed.

Individual measured value：

CH1	CR LF

Individual measurements are followed by a terminator.

Note: The unit of the measured value is the same as the unit set by the gauge.

When this communication protocol is used, when the current mode is real-time measurement, a [Send] button will appear on the main screen, and by clicking this button, the current measured value will be sent to other devices (PC or PLC) actively. If the current mode is Trigger Measurement, after the measurement is completed, the measured value will be sent to other devices (PC or PLC) actively. If the sending is successful, there will be a success message on the screen. When there is failure or timeout, please check the communication connection cable and the receiving device to see if there is any upper case OK character returned.

3.2 Json protocol communication

The Json format and field meanings are listed here.

json_data_value(Basic information on workpiece) description

Property Name	Parameter type	Mandatory	Descriptions
EquipmentID	int	Y	Device number
Programme	string	Y	Detection programme
ProductCode	string	Y	Product code (scanning gun scanning code)
Data	List<json_data_value_data>	Y	List of detected values

json_data_value_data(Basic information on workpiece) description

Property Name	Parameter type	Mandatory	Descriptions
Part Result	string	Y	Part/workpiece results: 0 pass, 1 fail
Part Data	List<json_data_value_data>	Y	List of detected values

json_data_value_data_partdata(Basic information on workpiece) description

json_data_value_data(Basic information on workpiece) description

Property Name	Parameter type	Mandatory	Descriptions
DataTime	string	Y	Measurement time, accurate to the second
MeasureTypeID	string	Y	No need for a while
ColumnName	string	Y	Measurement item name/ID/identifier
ColumnDescribe	string	Y	Measurement item description/name
ColumnData	string	Y	Measured value
ColumnResult	string	Y	Measurement results: 0 pass, 1 fail
UpLimitValue	string	N	Upper tolerance
LwLimitValue	string	N	Lower tolerance
NominalValue	string	N	Nominal value
MeasUdm	string	N	Unit

jsonData example:

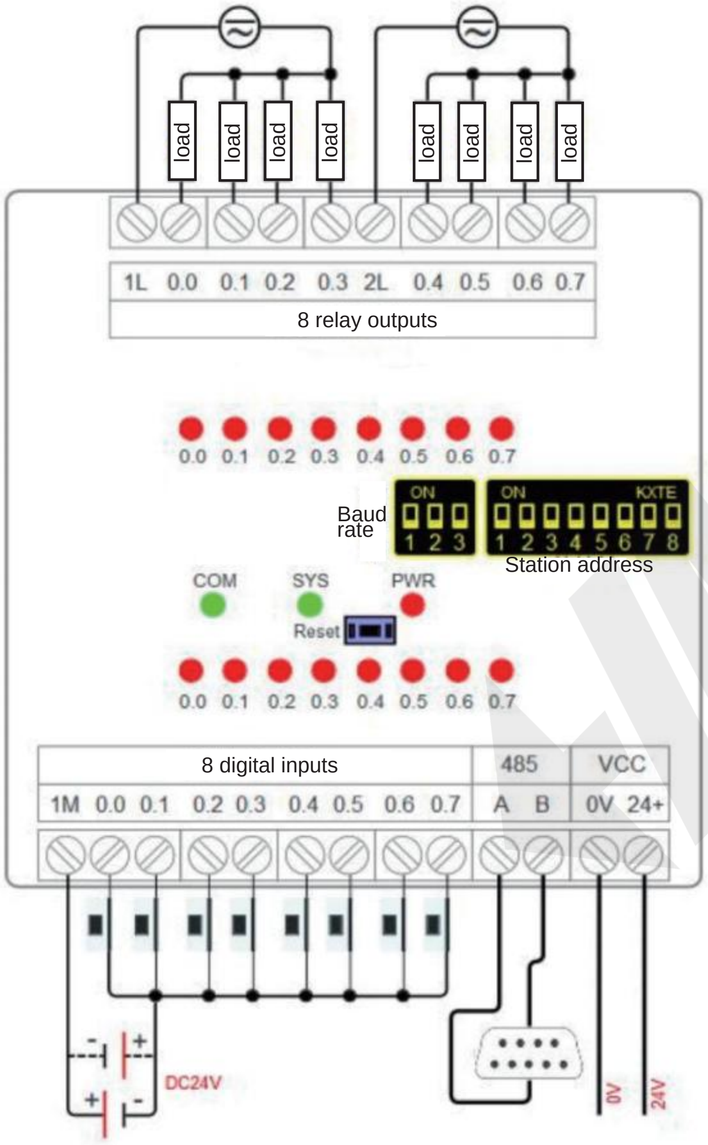
```
{
    "Code":"0",
    "Value":{
        "Programme":123,
        "EquipmentID":11,
         "ProductCode":"123456789",
        "Data":[
            {
                "PartResult":"0",
                "PartData":[
                    {
                        "DataTime":"2020-11-11 11:11:11",
                        "MeasureTypeID":"0",
                        "ColumnName":"M1",
                        "ColumnDescribe":"maximum values",
                        "ColumnData":"1.22",
                        "ColumnResult":"0"
                    },
                    {
                        "DataTime":"2020-11-11 11:11:11",
                        "MeasureTypeID":"0",
                        "ColumnName":"M2",
                        "ColumnDescribe":"Minimum Value ",
                        "ColumnData":"-1.22",
                        "ColumnResult":"0"
                    }
                ]
            },
            {
                "PartResult":"0",
                "PartData":[
                    {
                        "DataTime":"2020-11-11 11:11:11",
                        "MeasureTypeID":"0",
                        "ColumnName":"M1",
                        "ColumnDescribe":"maximum values",
                        "ColumnData":"1.22",
                        "ColumnResult":"0"
                    },
                    {
                        "DataTime":"2020-11-11 11:11:11",
                        "MeasureTypeID":"0",
                        "ColumnName":"M2",
                        "ColumnDescribe":"Minimum Value ",
                        "ColumnData":"-1.22",
                        "ColumnResult":"0"
                    }
                ]
            }
        ]
    }
}
```

When the receiver receives the data, just return {″code″: 0}

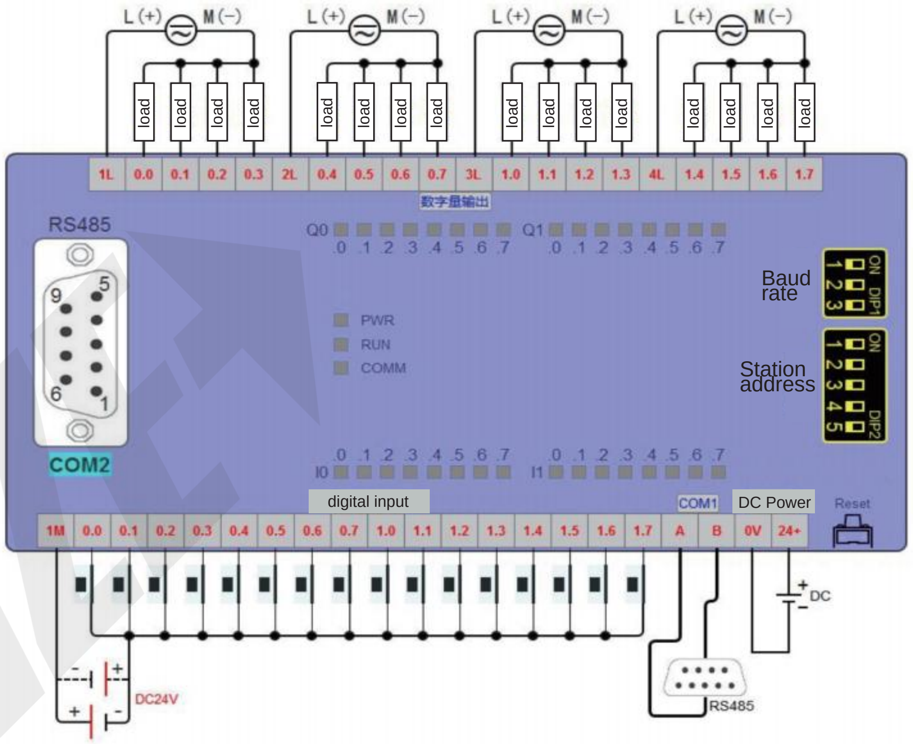
4.IO

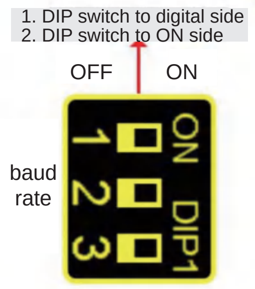
V4.0 version or higher, IO units with standard Modbus support are required. The gauge communicates with the IO unit via modbus. We have two types of units available, 8 in 8 out and 16 in 16 out. Customers can order according to their needs.

8-in 8-out pinout：

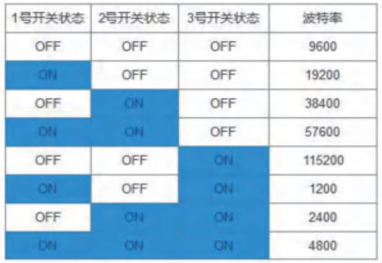


16-in 16-out pinout：





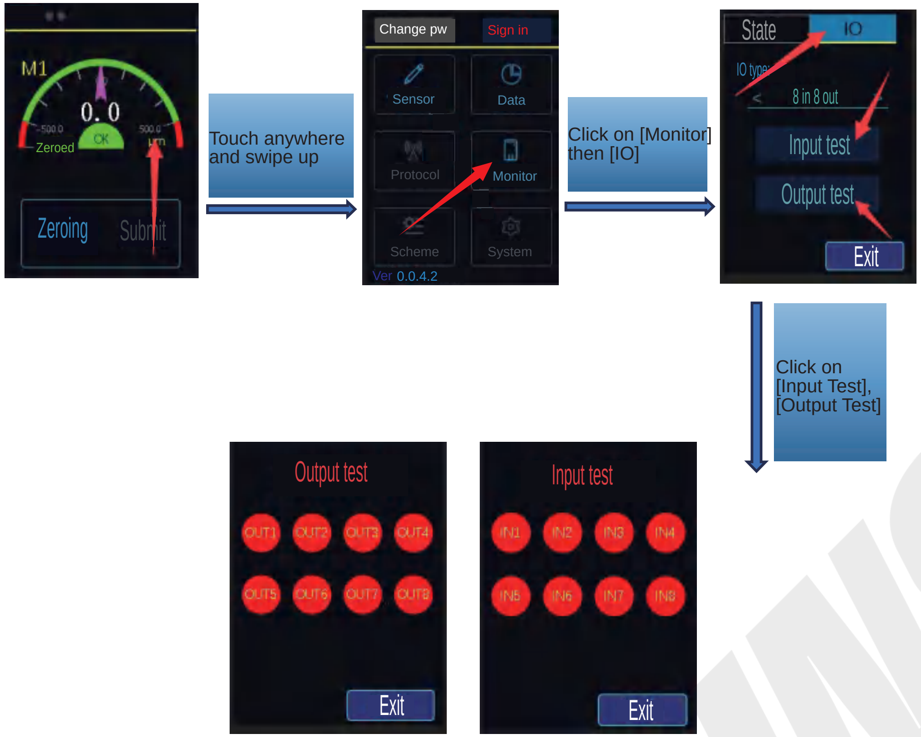
Baud rate dialling code corresponds to serial port baud rate correspondence table



1号开关状态	2号开关状态	3号开关状态	波特率
OFF	OFF	OFF	9600
ON	OFF	OFF	19200
OFF	ON	OFF	38400
ON	ON	OFF	57600
OFF	OFF	ON	115200
ON	OFF	ON	1200
OFF	ON	ON	2400
ON	ON	ON	4800

When shipped from the factory, the baud rate has been set, fixed at 115200, can not be changed; station address is also kept at 1, can not be changed, otherwise the communication will be unsuccessful.

After the IO unit is connected, you can enter the IO test interface for testing, follow the steps below to enter:


Touch anywhere and swipe up

Click on [Monitor] then [IO]

Click on [Input Test], [Output Test]

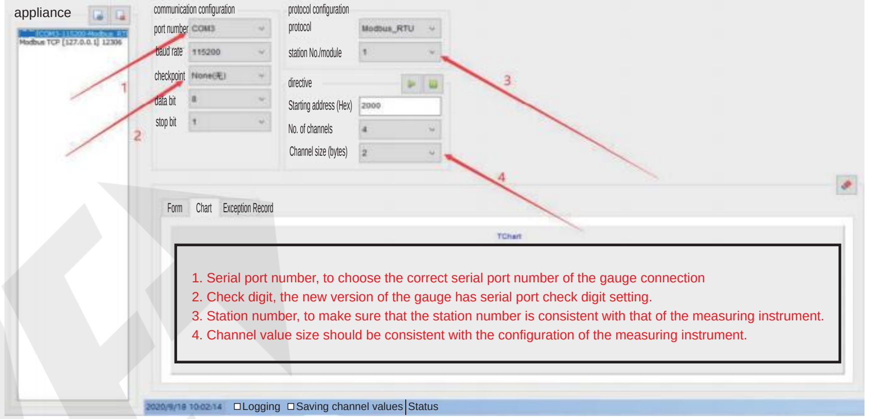
5.COMMUNICATIONS TESTING

Gauges can communicate with PLCs, computers and other devices. Due to the different communication needs of users, only computers are used for communication as an example.

- Serial communications

Firstly, suppose the gauge is configured as follows:
Port: RS232
Baud rate: 115200
Channel value type: 2
Protocols: modbus-rtu slave
Station number: 1

Then open the desktop software, TestForMeterControl.exe. Configure the software configuration items as described above.


communication configuration
port number COM3
baud rate 115200
checkpoint None(无)
data bit 8
stop bit 1
protocol configuration
protocol Modbus_RTU
station No./module 1
Starting address (Hex) 2000
No. of channels 4
Channel size (bytes) 2

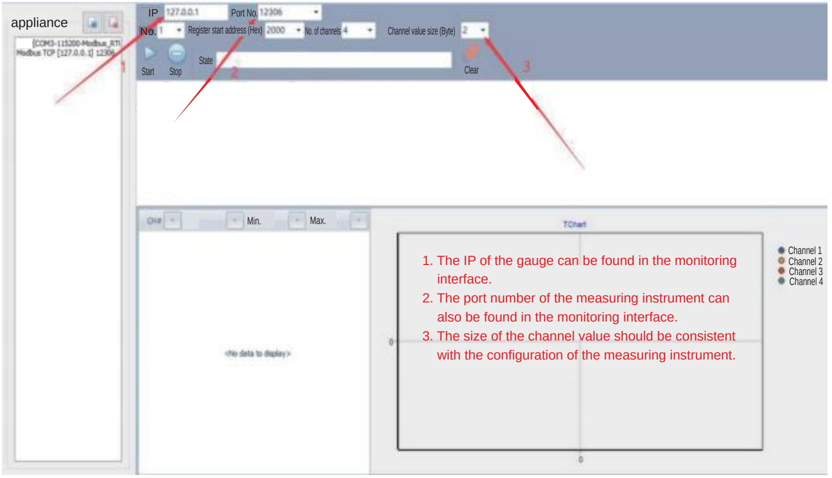
1. Serial port number, to choose the correct serial port number of the gauge connection
2. Check digit, the new version of the gauge has serial port check digit setting.
3. Station number, to make sure that the station number is consistent with that of the measuring instrument.
4. Channel value size should be consistent with the configuration of the measuring instrument.

2020/9/18 10:02:14 □Logging □Saving channel values Status

Once configured, click on the green start button in the figure to see the data for all channels on the gauge in the channel value list.

- network port communication

When communicating through the network port, the measuring instrument is a TCP server, so the client (PC or PLC) has to be configured as a TCP client, and the measuring instrument can be configured as fixed IP or DHCP to get IP. The gauge can be configured as fixed IP and DHCP to get IP. You can check the current IP and port number in [Monitor] interface, while the client can connect to the gauge by inputting the IP and port number of the gauge.


IP 127.0.0.1
Port No. 12306
No. 1
Register start address (Hex) 2000
No. of channels 4
Channel value size (Byte) 2

1. The IP of the gauge can be found in the monitoring interface.
2. The port number of the measuring instrument can also be found in the monitoring interface.
3. The size of the channel value should be consistent with the configuration of the measuring instrument.

Note that the device IP in the figure above is the IP address of the gauge, all items need to be entered after clicking [Enter] software will be correctly modified. After the setting is completed, click Start to connect the gauge.

6.COMMUNICATION FAILURE DESCRIPTION

In the communication of the measuring instrument, it may be encountered that the communication is not working properly. The problems and solutions found are listed below for users' reference.

· network port communication

When the modbus-rtu protocol is selected, you need to specify the station number for the gauge, which is usually 1. When the communication encounters problems, first check whether the station number of the gauge is the same as the requested station number of the master device.

Example: If the command station number requested by the master is 1 and the station number of the gauge is set to 0, communication will fail.

· The issue of the number of channels

When the master requests data from the gauge, the master needs to provide the number of channels for which data is requested. When the number of requested registers is greater than the maximum register address corresponding to the channel, a communication error occurs, and there is usually a register address error indication on the gauge. When 2 is selected for the channel value type of the gauge, the number of registers is the number of channels; when 4 is selected for the channel value type, 2 registers represent 1 channel.

Example: The number of gauge channels is set to 8.

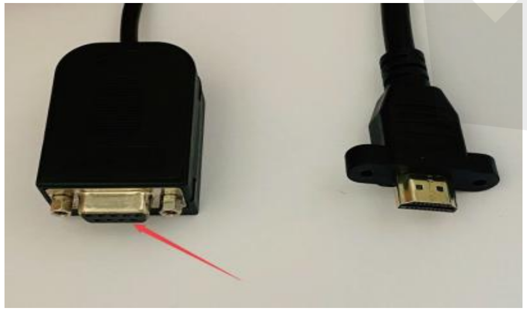
When the channel value type is 2, when the master requests all channel data, the number of registers should be set to 8. If it is greater than 8, the communication will fail.

When the channel value type is 4 and the number of registers is set to 16, communication will fail if it is greater than 16.

· Hardware connection problems

The most common communication failure is a problem with the connecting cable.

1.When using RS232 communication, you need to confirm whether the serial cable that comes with the gauge is a bare cable or a 9-pin serial port holder, as shown in the figure:



Shown in the figure for the 9-pin serial port holder, this interface is generally configured with a two-pronged male serial port cable, which can be directly connected to the device's serial port holder can be.

2.The use of RS485 communication, 485 communication interface will generally lead to the factory 4 wires, respectively, A, B, GND, earth. Normal conditions only need to access A, B can be, if the communication is not normal, the earth into the equipment on the ground can be.

7.DOCUMENT MODIFICATION RECORD

Version number	Revision content
V3.0	1. Add programme selection instructions 2. Modify the command triggering flow instruction diagram
V4.0	Overhaul the document and add detailed explanation of modbus commands.