9428-B series

MULTICHANNEL INDUCTANCE PROBE DISPLAY OPERATION MANUAL

www.insize.com

https://m.insize.com/9428-B.html

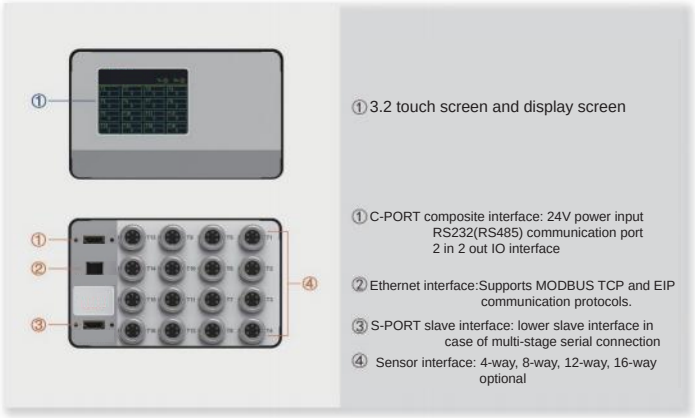
EN -- Please scan the QR code or visit the website for operation manual.
IT --- Scansiona il codice QR oppure visita il sito web per il manuale d'uso.
CZ -- Pro návod prosím naskenujte QR kód nebo navštivte webovou stránku.
ES -- Por favor, escanee el código QR o visite la página web para ver el manual 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-9428-B-E
V0

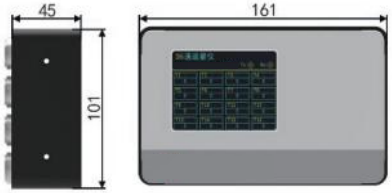
1.Product Introduction

9428-B series multichannel inductance probe displays，Can access 4-16 channels of LVDT sensors, or multi-stage series connection to access 256 channels of LVDT signals.Data can be output to PC and PLC via RS232, RS485 or USB for measurement of complex workpieces.

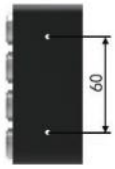
1.1 Clarification



· Installation And Dimensions(mm)



mounting hole：



M3 threaded hole

SPECIFICATION

Channel number *	1~16 (selectable)
Display mode	banner display
Unit	μm, mm
Range (mm)	±0.5, ±1, ±2, ±5, ±10 (adjustable)
Resolution	0.1μm
Sampling time	2ms
Operation environment	0~60°C, 20~85% (non-condensing)
Storage environment	-20~70°C, 20~85% (non-condensing)
Power supply	DC 24V
Dimension (L×W×H)	161×101×45mm
Weight	502g

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

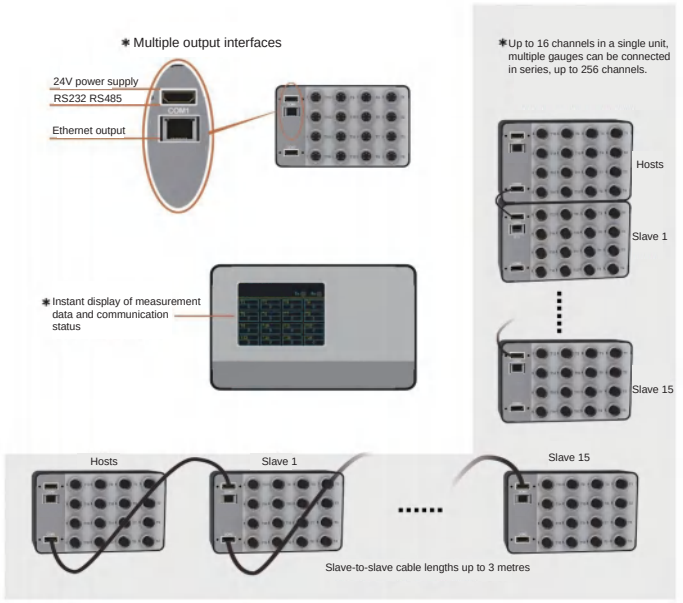
SENSOR TYPE

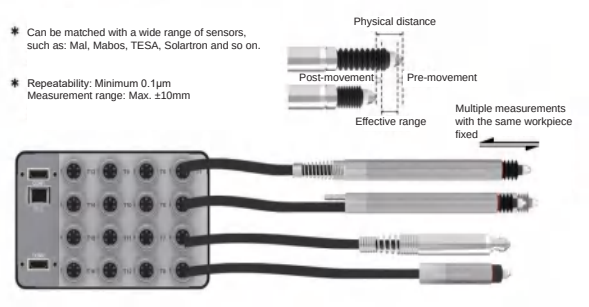
T1	full bridge
T2	half bridge

COMMUNICATION PORT

P1	network port (Modbus TCP and EtherNet/IP)
P2	RS232 (Modbus RTU)
P3	RS485 (Modbus RTU)

1.2 Product Features





2.Operation Guide

2.1 Main Monitoring Interface

After powering up the gauge, it will automatically enter the monitoring interface, presenting the data of all channels, as well as the communication status. As shown in the figure below:

Figure 1

· Mark 1: Cache light green indicates that caching is currently in progress, red indicates that the cache is full. The cache is a memory area where the gauge stores high-speed sampled values, which can be read when a large number of consecutive sampling points are required.

· Mark 2: Tx and Rx lamps indicate the transmission and reception status of communication with external devices respectively; grey indicates no communication, green indicates successful communication and red indicates communication failure.

· Mark 3: Channel value display area, including: length value, change bar, unit. Units are displayed when the total number of channels is less than or equal to 12.

· Mark 4: Currently configured communication parameters, including: interface type, baud rate, networking status, etc.

· Mark 5: When a small blue light appears in the display area of each channel, it means that the channel has been zeroed; when a small green light appears, it means that the channel has been inverted.

2.2 Fixed Zero and Inverse

The gauge provides the ability to zero and invert individual channels, firstly by clicking on any area of the main screen in Figure 1, which will bring up the screen in Figure 2:

Figure 2

Click on Zero Setting or Inverse in Figure 2 and enter the password:0000 to enter the interface of Zero Setting and Inverse. As shown in Figure 3:

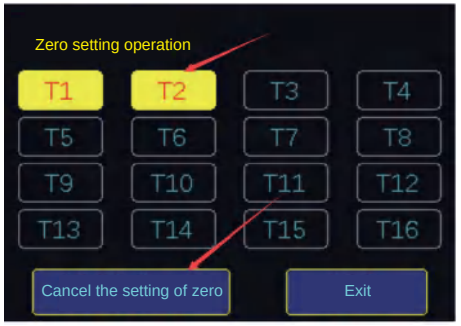


Figure 3

Click any button of T1~T16 in Fig. 3 to zero or cancel the corresponding channel. When the bottom colour of the button changes to yellow and the font changes to red, it means the current channel has been zeroed, and you can cancel the zeroing by clicking it again.

The operation of taking the inverse is the same as that described above.

2.3 Communication Interface Configuration

There are 3 types of communication interfaces, RS232, RS485, and wired network port.

Serial port configuration:

Among them, RS232 and RS485 belong to the serial interface, the configuration items are basically the same, as shown in the following figure.

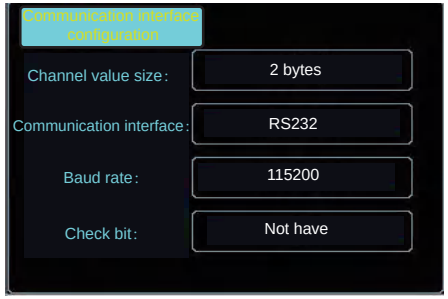


Figure 4

Serial Port Format Default Configuration:

baud rate	115200
data bit	8-bit
stop bit	1-bit
check digit	not have

Modbus protocols (rtu and tcp) need to acknowledge the size of the channel value, i.e. the channel value needs to be represented in several registers. Usually 1 register of modbus is 16 bits, 2 bytes, the maximum number that can be expressed is ±2^15, i.e. -32767~32767. after changing to the representation format of 0.1μm accuracy, the maximum displacement value that can be expressed is -3276.7μm and 3276.7μm. so if the total range of the sensor is larger than this limit, we should use 2 registers (4 bytes) to to represent 1 channel value. Users need to choose the size here according to the range of their sensor.

Network Port Configuration:

The gauge is used as a TCP server in network communication, and the device is required to connect to the gauge as a TCP client.

The configuration items are shown below:

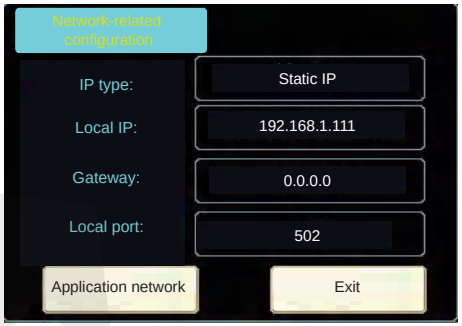


Figure 5

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

measuring instrument IP	192.168.1.111
gateways	0.0.0.0
Measuring instrument port	502

If the gauge is connected to the terminal after passing through the router device, you can choose the dynamic IP, so that the gauge will get the corresponding IP automatically, and the user only needs to set the port number, and generally keep 502 unchanged.

After configuring the network items, you have to click the [Apply Network] button.

2.4 Communication Status View

Gauges generally need to communicate with external devices, including PLCs and host computers. When communicating, you can check whether the current communication byte stream is correct in the status interface of the gauge, which is convenient for debugging. Firstly, click the status button in the main interface of Figure 2 to enter the interface shown in Figure 6 below:

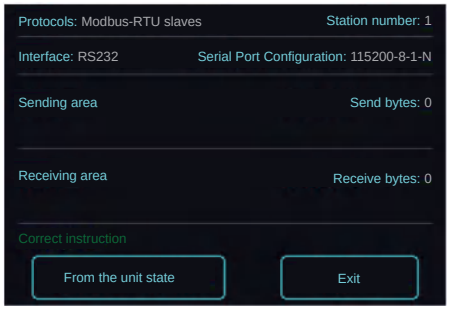


Figure 6

In the Status Monitor screen, you can view the current protocol type, interface type, and interface configuration information. When the communication starts, you will see the correct byte stream in the send and receive area, and when the communication is unsuccessful, here you can also check whether the command corresponding to the byte stream is correct.

2.5 Sensor Calibration

Gauges are generally calibrated at the factory for the optional transducer, and if the transducer is replaced the gauge needs to be recalibrated for magnification.

Click on the Sensors screen and click on the Linear Calibration button, then select a channel, such as T1, which will appear as shown in Figure 7 below:

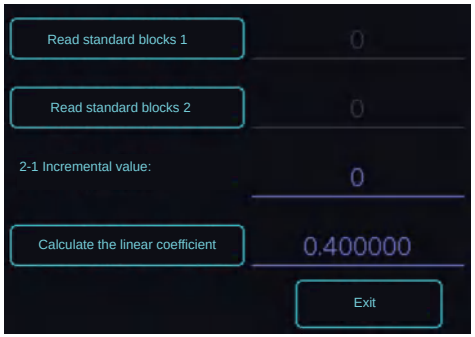


Figure 7

The calibration procedure consists of 4 steps:

1. Put standard block 1 into place and press [Read standard block 1 value], its length value will appear.
2、Remove the standard block 1, put the standard block 2 into place, and then press [Read the value of standard block 2], its length value will appear.
3, and then manually enter the difference between the two standard blocks.
4, finally, press[calculate the linear coefficient], you can calibrate. After the calibration is completed, you can test it, if it is not accurate, you can repeat steps 1-4, calibrate again, you must follow the above steps.

Note: Sensor calibration can also be performed by directly modifying the value of the linearity factor.

3.Description of Communications

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 transmitted on a TCP/IP network, with one more message header (MBAP), less CRC checksum, and the use of port 502 of TCP; RTU has more device address and CRC checksum.

The gauge supports 3 function codes: 0x03, 0x06 and 0x10.

0x03	0x06	0x10
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 Register Address

Register Address	Element	limits authority	Function code
0x2000	Real-time values of the current sensor, channel values from 0x2000 and so on backwards	read-only	0x03
0x0C0A	Takes the inverse address. Currently supports zeroing 64 channels simultaneously. Write command: 1 for zero fixing, 0 for no operation. Read command: 1 means zero setting, 0 means no zero setting.	Reading and writing	0x03、 0x06、0x10
0x0C20	Fixed-zero address. Currently supports zeroing 64 channels simultaneously. Write command: 1 for zeroing, 0 for no operation. Read command: 1 means zeroed, 0 means not zeroed.	Reading and writing	0x06、0x10
0x0C30	Cancels the fixed-zero address. Currently, 64 channels are supported to be zeroed at the same time. Write command: 1 for cancellation, 0 for no operation.	Write	0x06、0x10

3.1.2 MODBUS Error Protocol Package

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

MODBUS-TCP					
Transaction Identification	Protocol Identifier	lengths	unit identifier	Exception Function Code	Error Code
2 bytes	2 bytes	2 bytes	1 bytes	function code+0x80	0x01~0x08

Error Code Definition:

Error Code	Hidden Meaning
0x01	Illegal function codes, i.e. the system does not support the function code
0x02	Illegal data addresses For example, if 4 sensors are currently supported, the legal address is 0x0C20-0x0C24, beyond the change address, that is, the data address error is returned.
0x03	illicit data value
0x08	CRC calibration error

3.1.3 MODBUS Directives

· RTU Data Acquisition Command——0x2000

The function code request format of the MODBUS-RTU instruction 0x03 is fixed at 8 bytes									
	Descriptions		Function code	Register Addresses		Number of registers		CRC	
	Byte Position	1	2	3	4	5	6	7	8
	Remark	station number	Read Holding Register	High Byte	Low Byte	High Byte	Low Byte	Low Byte	High Byte
The size of the channel value is 2 bytes	Get T1 sensor value	1	03	20	00	00	01	8F	CA
	Get T2 sensor value	1	03	20	01	00	01	DE	0A
	Get T1, T2 sensor values	1	03	20	00	00	02	CF	CB
The size of the channel value is 4 bytes	Get T1 sensor value	1	03	20	00	00	02	CF	CB
	Get T2 sensor value	1	03	20	02	00	02	6E	0B
	Get T1, T2 sensor values	1	03	20	00	00	04	4F	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 changes are the register address and the number of registers) and just recalculate the last 2 CRC bytes, you can calculate the CRC online via this link, pay attention to the order of the high and low bytes of the CRC.

MODBUS-RTU Command 0x03 Function Code Return Format										
	Descriptions		Function code	Byte Count	processor register 1		processor register 2		CRC	
	Byte Position	1	2	3	4	5	6	7		
	Remark	station number	Read Holding Register	Register Bytes	High Byte	Low Byte	High Byte	Low Byte	Low Byte	High Byte
The size of the channel value is 2 bytes	T1 value returned	1	03	02	EA	20			F6	FC
	T1,T2 value returned	1	03	04	EA	20	0B	22	48	C8
The size of the channel value is 4 bytes	T1 value returned	1	03	04	FF	F7	74	80	5D	75

Among the returned bytes, the register value 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:

```
// Obtain the value of two bytes sent to the Modbus device (big-endian mode)
uint8_t byte_high = 0xEA; // High-order byte
uint8_t byte_low = 0x20; // Low byte

// Combine the values of two bytes into a signed 16-bit integer
int16_t combined_value = (int16_t) ((byte_high << 8))
```

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:

```
// Combine the values of two bytes into a signed 16-bit integer
uint8_t byle1  = 0xEF; // The first byte
uint8_t byte2 = 0xF7; // The second byte
uint8_t byle3  = 0x74; // The third byte
uint8_t byte4 = 0x80; // The second byte
// Combine four byte values into a signed 32-bit integer value
int32_t combined_value  = ((int32_t) bytel << 24) | ((int32_t)byte2 << 16) | ((int32_t) << 8)| byte4
```

· RTU sensor zero command——0x0C20、0x0C30

MODBUS-RTU command 0x06 function code request format fixed 8 bytes								
Descriptions		Function code	Register Addresses		Register Value		CRC	
Byte Position	1	2	3	4	5	6	7	8
Remark	station number	Write Single Holding Register	High Byte	Low Byte	High Byte	Low Byte	Low Byte	High Byte
T1 Zero setting	1	06	0C	20	00	01	4A	90
T1,T2 Zero setting	1	06	0C	20	00	03	CB	51
16-channel zeroing	1	06	0C	20	FF	FF	8A	E0

When two measuring instruments are combined to set the zero, the 0x10 function code is required.

MODBUS-RTU command 0x10 function code request format												
Descriptions		Function code	Register Addresses		Number of registers	byte count	registers 1		registers 2		CRC	
Byte Position	1	2	3	4	5、6	7	8	9	10	11	12	13
Remark	station number	Write Holding Register	High Byte	Low Byte		Register Bytes	High Byte	Low Byte	High Byte	Low Byte	Low Byte	High Byte
32-channel zeroing	1	10	0C	20	0x0002	04	FF	FF	FF	FF	A5	23

The value of register 1 is the zeroing operation for the first measuring instrument, and the value of register 2 is the zeroing operation for the second one.

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	Explanation
0x0C20	Writing 1 to a certain value in a register indicates a fixed zero, while writing 0 is not processed
0x0C30	Writing 1 to a certain value in a register indicates the cancellation of zero setting, while writing 0 is not processed

· RTU get fixed-zero state command——0x0C20

MODBUS-RTU Command 0x03 function code request format fixed 8 bytes								
Descriptions		Function code	Register Addresses		Number of registers		CRC	
Byte Position	1	2	3	4	5	6	7	8
Remark	station number	Read hold register	High Byte	Low Byte	High Byte	Low Byte	Low Byte	High Byte
16-channel fixed zero state	1	03	0C	20	00	01	86	90

The return format of the function code 0x03 of the MODBUS-RTU instruction							
Descriptions		Function code	Register byte	Register value		CRC	
Byte Position	1	2	3	4	5	7	8
Remark	station number	Read hold register		High Byte	Low Byte	Low Byte	High Byte
16-channel zeroing state return	1	03	02	33	3F	EC	A4

The register value of the return value, 0x333F, represents the fixed-zero state of 16 channels, each bit represents the state of 1 channel, 0x333F is converted to binary : 0011001100111111111, from left to right, the address is from high to low, which represents from 16 channels to 1 channel, in which 16, 15, 12, 11, 8, 7 channels are not fixed zero, and the other channels have been fixed zero.

· The RTU sensor takes the reverse instruction——0x0C0A

The return format of the function code 0x03 of the MODBUS-RTU instruction								
Descriptions		Function code	Register Addresses		Register value		CRC	
Byte Position	1	2	3	4	5	6	7	8
Remark	station number	Write a single hold register	High Byte	Low Byte	High Byte	Low Byte	Low Byte	High Byte
T1 take the opposite value	1	06	0C	0A	00	01	6B	58
T1,T2 take the opposite value	1	06	0C	0A	00	03	EA	99
16 channels take the opposite value	1	06	0C	0A	FF	FF	AB	28

Inverting the 2 combinations together requires the 0x10 function code.

MODBUS-RTU instruction 0x10 function code request format												
Descriptions		Function code	Register Addresses		Number of registers	byte count	registers 1		registers 2		CRC	
Byte Position	1	2	3	4	5、6	7	8	9	10	11	12	13
Remark	station number	Write hold register	High Byte	Low Byte		Register byte	High Byte	Low Byte	High Byte	Low Byte	Low Byte	High Byte
16 channels take the opposite value	1	10	0C	0A	0x0002	04	FF	FF	FF	FF	27	44

Register 1 is the inverse operation for the first MU and register 2 is the inverse operation for the second gauge.

· RTU Get Reverse Status Command——0x0C0A

MODBUS-RTU instruction 0x10 function code request format								
Descriptions		Function code	Register Addresses		Number of registers		CRC	
Byte Position	1	2	3	4	5	6	7	8
Remark	station number	Read hold register	High Byte	Low Byte	High Byte	Low Byte	Low Byte	High Byte
16-channel inversion state	1	03	0C	0A	00	01	A7	58

Descriptions		Function code	Register byte	Register value		CRC	
Byte Position	1	2	3	4	5	7	8
Remark	station number	Read hold register		High Byte	Low Byte	Low Byte	High Byte
16-channel inverse status return	1	03	02	FF	FF	B9	F4

The register value of the return value, 0xFFFF, represents the inversion status of 16 channels, with each bit representing the status of 1 channel. 0xFFFF is converted to binary: 1111111111111111, from left to right, the address from high to low, indicating that from 16 channels to 1 channel, all channels have been inverted.

· TCP data fetch command——0x2000

The function code request format of the MODBUS-TCP instruction 0x03 is fixed at 12 bytes										
	Descriptions	Transaction processing identifier	Protocol identification	lengths	Unit identifier	Function code	Register address		The number of registers	
	Byte Position	1、2	3、4	5、6	7	8	9	10	11	12
	Remark	The host initiates and the slave replicates	modbus is fixed to 0	Next byte to the end	The host initiates and the slave replicates	Read hold register	High Byte	Low Byte	High Byte	Low Byte
The size of the channel value is 2 bytes	Obtain the T1 value	0x9776	0x0000	0x0006	0x04	03	20	00	00	01
	Obtain the T2 value	0x9776	0x0000	0x0006	0x04	03	20	01	00	01
	Obtain the values of T1 and T2	0x9776	0x0000	0x0006	0x04	03	20	00	00	02
The size of the channel value is 4 bytes	Obtain the T1 value	0x9776	0x0000	0x0006	0x04	03	20	00	00	02
	Obtain the T2 value	0x9776	0x0000	0x0006	0x04	03	20	02	00	02
	Obtain the values of T1 and T2	0x9776	0x0000	0x0006	0x04	03	20	00	00	04

If you want to get the values of other channels or more channels, 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 0x9776 and the unit identifier 0x04 are just examples, and the actual application is based on the host's own MAP message.

MODBUS-RTU instruction 0x10 function code request format												
	Descriptions	Transaction processing identifier	Protocol identification	lengths	Unit identifier	Function code	Register byte	Registers 1		Registers 2	
	Byte Position	1、2	3、4	5、6	7	8	9	10	11	12	13
	Remark	Copy the data of the request frame	modbus is fixed to 0	Next byte to the end	Copy the data of the request frame	Read hold register	Register byte	High Byte	Low Byte		
The size of the channel value is 2 bytes	Obtain the T1 value	0x9776	0x0000	0x0005	0x04	03	02	EA	20		
	Obtain the values of T1 and T2	0x9776	0x0000	0x0007	0x04	03	04	EA	20	0B	22
The size of the channel value is 4 bytes	Obtain the T1 value	0x9776	0x0000	0x0007	0x04	03	04	FF	F7	74	80

Note how the lengths 0x0005 and 0x0007 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:

```
// Obtain the value of two bytes sent to the Modbus device (big-endian mode)
uint8_t byte_high = 0xEA; // High-order byte
uint8_t byte_low = 0x20; // Low byte

// Combine the values of two bytes into a signed 16-bit integer
int16_t combined_value = (int16_t) ((byte_high << 8))
```

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, conversion to displacement value needs to be divided by 1000 to get -560.0μm.

The calculation code is referenced below:

```
// Combine the values of two bytes into a signed 16-bit integer
uint8_t byle1  = 0xEF; // The first byte
uint8_t byte2 = 0xF7; // The second byte
uint8_t byle3  = 0x74; // The third byte
uint8_t byte4 = 0x80; // The second byte
// Combine four byte values into a signed 32-bit integer value
int32_t combined_value  = ((int32_t) bytel << 24) | ((int32_t)byte2 << 16) | ((int32_t) << 8)| byte4
```

· TCP Sensor Zero Command——0x0C20、0x0C30

The function code request format of the MODBUS-TCP instruction 0x06 is fixed at 12 bytes									
Descriptions	Transaction processing identifier	Protocol identification	lengths	Unit identifier	Function code	Register address		Register value	
Byte Position	1、2	3、4	5、6	7	8	9	10	11	12
Remark	The host initiates and the slave replicates	modbus is fixed to 0	Next byte to the end	The host initiates and the slave replicates	Write a single hold register	High Byte	Low Byte	High Byte	Low Byte
T1 Zero setting	0x9776	0x0000	0x0006	0x04	06	0C	20	00	01
T1,T2 Zero setting	0x9776	0x0000	0x0006	0x04	06	0C	20	00	03
16-channel zeroing	0x9776	0x0000	0x0006	0x04	06	0C	20	FF	FF

When two measuring instruments are combined to set the zero, the 0x10 function code is required.

MODBUS-TCP instruction 0x10 function code request format													
Descriptions	Transaction processing identifier	Protocol identification	lengths	Unit identifier	Function code	Register address		The number of registers	Number of bytes	Register 1		Register 2	
Byte Position	1、2	3、4	5、6	7	8	9	10	11、12	13	14	15	16	17
Remark	The host initiates and the slave replicates	modbus is fixed to 0	Next byte to the end	The host initiates and the slave replicates	Write hold register	High Byte	Low Byte	High Byte	Register byte	High Byte	Low Byte	High Byte	Low Byte
32-channel zeroing	0x9776	0x0000	0x0006	0x04	10	0C	20	0x0002	04	FF	FF	FF	FF

The value of register 1 is the zero setting operation for the first measuring instrument, and the value of register 2 is the zero setting operation for the second measuring instrument.

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	Explanation
0x0C20	Writing 1 to a certain value in a register indicates a fixed zero Writing to 0 is not processed
0x0C30	Writing a 1 to a bit of the register value cancels the zero setting Writing 0 is not processed.

· TCP Sensor Get Fixed-Zero State Command——0x0C20

The function code request format of the MODBUS-TCP instruction 0x03 is fixed at 12 bytes									
Descriptions	Transaction processing identifier	Protocol identification	lengths	Unit identifier	Function code	Register address		The number of registers	
Byte Position	1、2	3、4	5、6	7	8	9	10	11	12
Remark	The host initiates and the slave replicates	modbus is fixed to 0	Next byte to the end	The host initiates and the slave replicates	Read hold register	High Byte	Low Byte	High Byte	Low Byte
16-channel fixed zero state	0x9776	0x0000	0x0006	0x04	03	0C	20	00	01

MODBUS-TCP instruction 0x03 function code return format								
Descriptions	Transaction processing identifier	Protocol identification	lengths	Unit identifier	Function code	Register byte	Register value	
Byte Position	1、2	3、4	5、6	7	8	9	10	11
Remark	Copy the data of the request frame	modbus is fixed to 0	Next byte to the end	Copy the data of the request frame	Read hold register	Register byte	High Byte	Low Byte
Return in a fixed zero state	0x9776	0x0000	0x0005	0x04	03	02	33	3F

The register value of the return value, 0x333F, represents the fixed-zero state of 16 channels, each bit represents the state of 1 channel, 0x333F is converted to binary : 0011001100111111111, from left to right, the address is from high to low, which represents from 16 channels to 1 channel, in which 16, 15, 12, 11, 8, 7 channels are not fixed zero, and the other channels have been fixed zero.

· TCP Sensor Inverse Command——0x0C0A

The function code request format of the MODBUS-TCP instruction 0x06 is fixed at 12 bytes									
Descriptions	Transaction processing identifier	Protocol identification	lengths	Unit identifier	Function code	Register address		Register value	
Byte Position	1、2	3、4	5、6	7	8	9	10	11	12
Remark	The host initiates and the slave replicates	modbus is fixed to 0	Next byte to the end	The host initiates and the slave replicates	Write a single hold register	High Byte	Low Byte	High Byte	Low Byte
T1 take the opposite value	0x9776	0x0000	0x0006	0x04	06	0C	0A	00	01
T1,T2 take the opposite value	0x9776	0x0000	0x0006	0x04	06	0C	0A	00	03
16-channel inversion	0x9776	0x0000	0x0006	0x04	06	0C	0A	FF	FF

When two measuring instruments are combined to reverse, the 0x10 function code is required.

MODBUS-TCP instruction 0x10 function code request format												
Descriptions	Transaction processing identifier	Protocol identification	lengths	Unit identifier	Function code	Register address		The number of registers	Number of bytes	Register 1		Register 2	
Byte Position	1、2	3、4	5、6	7	8	9	10	11、12	13	14	15	16	17
Remark	The host initiates and the slave replicates	modbus is fixed to 0	Next byte to the end	The host initiates and the slave replicates	Write hold register	High Byte	Low Byte		Register byte	High Byte	Low Byte	High Byte	Low Byte
32-channel inversion	0x9776	0x0000	0x0006	0x04	10	0C	0A	0x0002	04	FF	FF	FF	FF

Register 1 performs the reverse operation on the first measuring instrument, and register 2 performs the reverse operation on the second measuring instrument.

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 Get Reverse Status Command——0x0C0A

The function code request format of the MODBUS-TCP instruction 0x03 is fixed at 12 bytes									
Descriptions	Transaction processing identifier	Protocol identification	lengths	Unit identifier	Function code	Register address		The number of registers	
Byte Position	1、2	3、4	5、6	7	8	9	10	11	12
Remark	The host initiates and the slave replicates	modbus is fixed to 0	Next byte to the end	The host initiates and the slave replicates	Read hold register	High Byte	Low Byte	High Byte	Low Byte
16-channel inversion state	0x9776	0x0000	0x0006	0x04	03	0C	0A	00	01

MODBUS-TCP instruction 0x03 function code return format								
Descriptions	Transaction processing identifier	Protocol identification	lengths	Unit identifier	Function code	Register byte	Register value	
Byte Position	1、2	3、4	5、6	7	8	9	10	11
Remark	Copy the data of the request frame	modbus is fixed to 0	Next byte to the end	Copy the data of the request frame	Read hold register	Register byte	High Byte	Low Byte
Inverse status return	0x9776	0x0000	0x0005	0x04	03	02	FF	FF

The register value of the return value, 0xFFFF, represents the inversion status of 16 channels, with each bit representing the status of 1 channel. 0xFFFF is converted to binary: 1111111111111111, from left to right, the address from high to low, indicating that from 16 channels to 1 channel, all channels have been inverted.

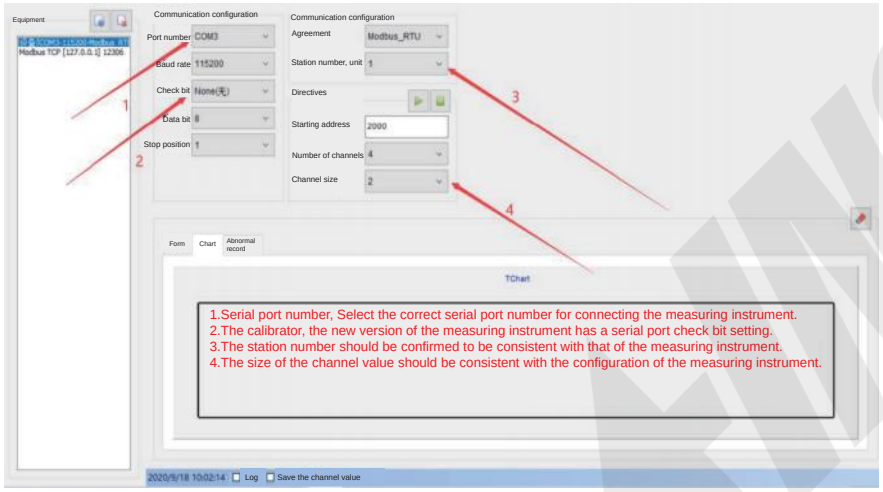
4.Communications Test

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 port communication

First, suppose the measuring instrument is configured as follows:
Interface: RS232
Baud rate: 115200
Channel value type: 2
Agreement: modbus-rtu slave
Station number: 1

Then open the desktop software, TestForMeterControl.exe。Align the software configuration items with the above configuration.

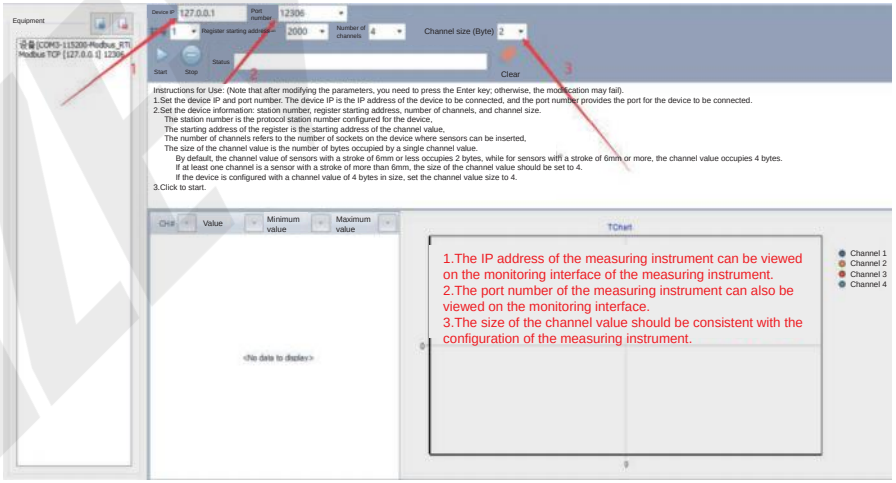


After configuration, click the green start button in the picture, and you can see the data of all channels on the measuring instrument in the channel value list.

· Network port communication

When communicating through the network port, the measuring instrument acts as a TCP server, so the client (computer or PLC) needs to be configured as a TCP client.

The measuring instrument can be configured to obtain an IP address either through a fixed IP address or DHCP. In the "Monitoring" interface, you can view the current IP and port number. The client only needs to enter the IP and port number of the measuring instrument to connect.



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.

5.Communication Failure Description

In the communication of measuring instruments, there may be cases where the communication is not working properly. The problems and solutions found are listed below for users' reference.

· Station numbering issues

When choosing the modbus-rtu protocol, it is necessary to specify the station number for the measuring instrument, which is generally 1. When communication encounters problems, first check whether the station number of the measuring instrument is consistent with the requested station number of the main 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 being 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:

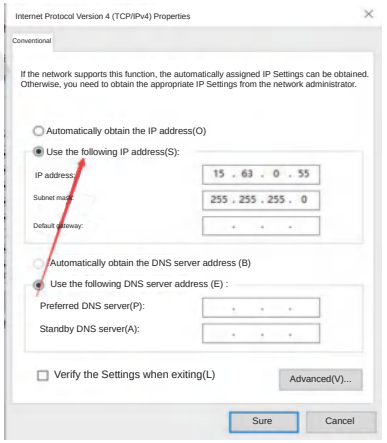
The 9-pin serial port is shown in Figure 1. For this type of interface, we usually configure a two-end male serial port cable at the factory, which can be directly connected to the serial port of the device.

Bare wires are shown in Figure 2, and generally 4 wires are drawn from the factory: TXD, RXD, GND, and shielded wires. The TXD of the gauge is connected to the RXD of the user's equipment, the RXD pin of the gauge is connected to the TXD of the equipment (i.e., cross-connected), the GND is connected to the GND of the equipment, and the shielding wire is connected to the shielding layer of the equipment or to the earth. If the shield is not connected, it may be interfered with, causing unstable communication.

2. The use of RS485 communication, 485 communication interface generally factory will lead to four 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.

· TCP communication issues

Usually the most common problem with TCP communication is the IP setting problem. When choosing a fixed IP to connect to the computer through the switch, the IP address in the wired network settings on the computer must choose a fixed IP, as shown in the figure below:



The IP address in the diagram can be entered at will and the subnet mask is fixed at 255.255.255.0

In addition, when PLC communicates with the gauge, it is mainly a Modbus-tcp format problem, please read the chapter on data parsing carefully as well as analyse where the byte stream in the [Monitor] interface of the gauge has gone wrong.

Siemens PLC comes with Modbus module, read the gauge data, the address written must be written as a decimal 48193, of which 40000 or more is Siemens modbus normal read, 8192 is a decimal 0x2000, and eventually add 1. Equivalent to the input of 48193, modbus byte stream is the correct 0x2000 this address code.