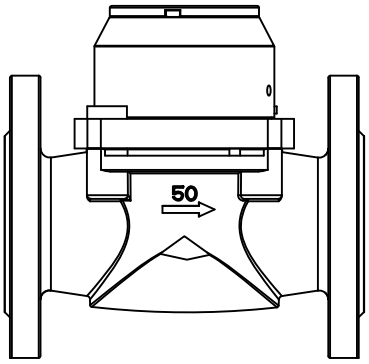
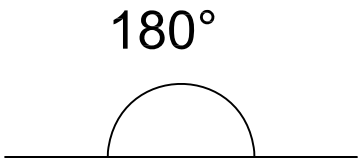
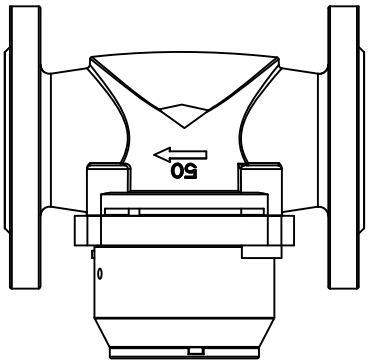
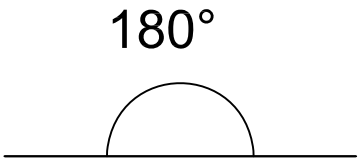
Flange End Smart Water Meter Installation Requirement





Horizontal Position ✓





Inverted Position ✗

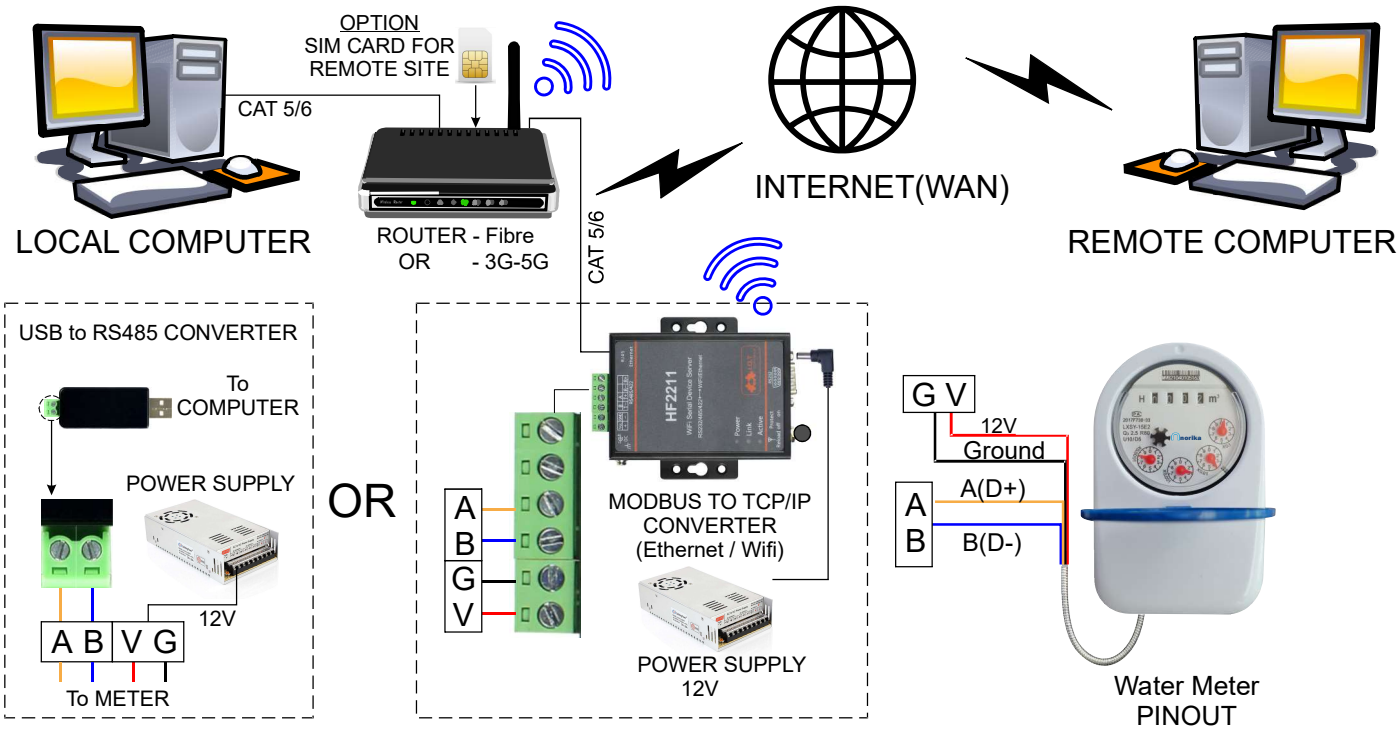
Installation Requirements

1.The meter should be installed in horizontal position with the direction of the flow as indicated by the arrow cast in the meter body with the register face upwards.

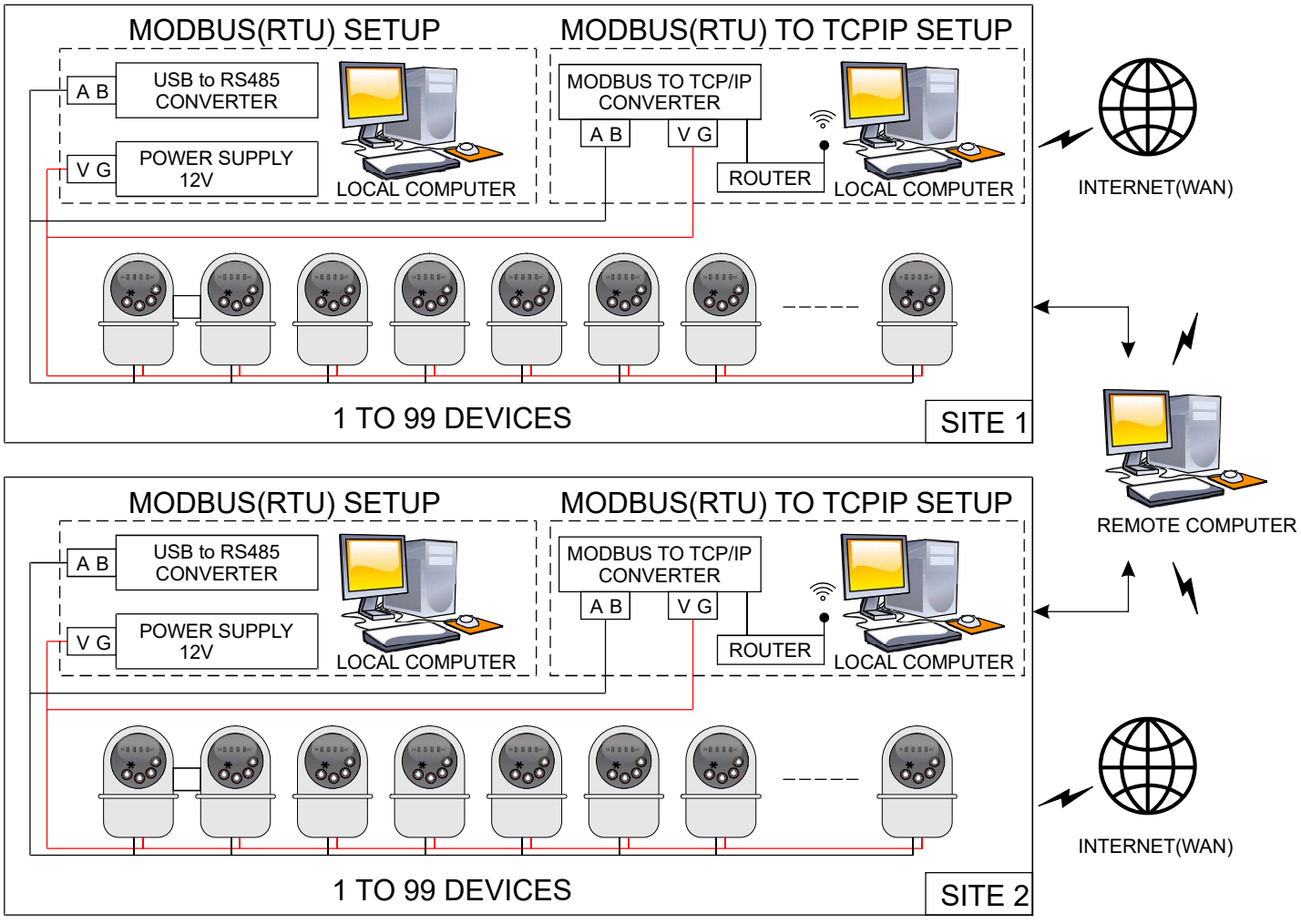
2.Pipeline must be flushed before installation.

3.The meter should be constantly full of water during operation.

CONNECTION PINOUT - WATER METER



CONNECTION NETWORK - TOPOLOGY



Water METER MODBUS-RTU Communication Protocol

MODBUS RTU(Remote Terminal Unit) communication protocol is used in this water meter. Modbus protocol defines the verification code and data sequence in detail, which are necessary for specific data exchange. Modbus protocol uses master slave connection (half duplex) on one communication line, which means that signals are transmitted in two opposite directions on a single communication line. First, the signal of the main computer is addressed to a unique terminal device (slave), and then the response signal from the terminal device is transmitted to the host in the opposite direction.

MODBUS protocol only allows communication between the host (PC, PLC, etc.) and terminal equipment, but does not allow data exchange between independent terminal devices, so that each terminal device will not occupy the communication line during their initialization, but only respond to the query signal arriving at the local computer.

MODBUS - RTU format has no start and end characters, but it needs to add a waiting time, and the waiting time is not less than 3.5 characters send time.

1) Transmission mode: the information transmission is asynchronous and takes byte as unit. The communication information transmitted between the host and slave is in 10 bit word format, including 1start bit, 8 data bits (the least significant bit is sent first), no parity bit and 1 stop bit. Fig.1

Fig.1

Serial Port	
Baud Rate	: 9600
Data bits	: 8
Parity	: None
Stop bits	: 1

2) Data frame format

Address Code	Function Code	Data Area	CRC Check Code
1 byte	1 byte	n byte	2 byte

Address code: at the beginning of the frame, the address code is composed of a byte (8-bit binary code). The decimal system is 0-255. Only 1-247 is used in the water meter, and other addresses are reserved. These bits indicate the address of the user specified terminal device, which will receive data from the connected host. The address of each terminal device must be unique. Only the addressed terminal will respond to the query containing the address. When the terminal sends back a response, the slave address data in the response tells the host which terminal is communicating with it.

Function code: function code tells the addressed terminal what function to perform. The following table lists the function codes used in this series of instruments, as well as their significance and functions.

Function Code		Description
1	Read Coil	Read Meter Valve Open/Close
3	Read Register	Read Meter Consumption
5	Write Coil	Execute to Open / Close Valve

Data area: the data area contains the data required by the terminal to perform a specific function or the data collected by the terminal in response to a query. The contents of these data may be values, reference addresses or set values. For example, the function code tells the terminal to read a register, and the data area needs to indicate which register to start from and how many data to read. The embedded address and data vary according to the type and different contents between the slave computers.

CRC check code: the error check (CRC) field takes two bytes and contains a 16 bit binary value. The CRC value is calculated by the transmission device, and then added to the data frame. The receiving device recalculates the CRC value when receiving data, and then compares it with the value in the received CRC domain. Ifthe two values are not equal, an error occurs.

The process of generating a CRC is as follows: ‘

1. Preset a 16 bit register as OFFFFH (all 1), which is called CRC register.
2. XOR operation is performed between the 8 bits of the first byte in the data frame and the low byte in the CRC register, and the result is saved back to the CRC register.
3. Move CRC register to the right, fill in 0 at the highest position, and move out the lowest position and detect.
4. If the lowest position is 0, repeat step 3 (next shift); if the lowest position is 1, the CRC register is exclusive or calculated with a preset fixed value (OAOUIH).
5. Repeat steps 3 and 4 until 8 shifts. This completes a complete eight bit.
6. Repeat steps 2 through 5 to process the next octet until all byte processing ends.
7. The value of the final CRC register is the value of the CRC.
   In addition, there is a method to calculate CRC by using preset tables. Its main feature is that the calculation speed is fast, but the table needs a large storage space. This method is not covered here, please refer to the relevant information.

4. If the lowest position is 0, repeat step 3 (next shift); if the lowest position is 1, the CRC register is exclusive or calculated with a preset fixed value (0A001H).
5. Repeat steps 3 and 4 until 8 shifts. This completes a complete eight bit.
6. Repeat steps 2 through 5 to process the next octet until all byte processing ends.
7. The value of the final CRC register is the value of the CRC. . .

In addition, there is a method to calculate CRC by using preset tables. Its main feature is that the calculation speed is fast, but the table needs a large storage space. This method is not covered here, please refer to the relevant information.

Detailed explanation of communication application format

(1) Function code 03H: read register

This function allows users to obtain the data and system parameters collected and recorded by the device. There is no limit on the number of data requested by the host at a time, but it cannot exceed the defined address range.
The following example is the basic data collected from No.01 machine reading (2 bytes for each address in the data frame). The collected data is the total water consumption (occupying 2 bytes}, and its address is 00H.

Host send		Message
Address code		01H
Function Code		03H
Starting address	High byte	00H
	Low byte	00H
No. of registers	High byte	00H
	Low byte	02H
CRC Check code	High byte	C4H
	Low byte	0BH

Slave Return		Message
Address code		01H
Function Code		03H
No. of bytes		04H
Register data	High byte	00H
	Low byte	12H
Register data	High byte	D6H
	Low byte	87H
CRC Check code	High byte	44H
	Low byte	34H

(2) Function code 01H: read coil

This function allows users to obtain the data and system parameters collected and recorded by the device. There is no limit on the number of data requested by the host at a time, but it cannot exceed the defined address range.
The following example is the basic data collected from No.63 machine reading (2 bytes for each address in the data frame). The collected data is the valve open/close status (occupying 1 byte)

Host send		Message
Address code		63H
Function Code		01H
Starting address	High byte	00H
	Low byte	01H
No. of coil	High byte	00H
	Low byte	01H
CRC Check code	High byte	A4H
	Low byte	48H

Slave Return		Message	Message
Address code		63H	63H
Function Code		01H	01H
Starting address	High byte	00H	00H
	Low byte	01H	01H
No. of coil	High byte	00H	00H
	Low byte	FFH	00H
CRC Check code	High byte	25H	65H
	Low byte	C8H	88H
FF = Valve open / 00 = Valve close			

(3) Function code 05H: write coil

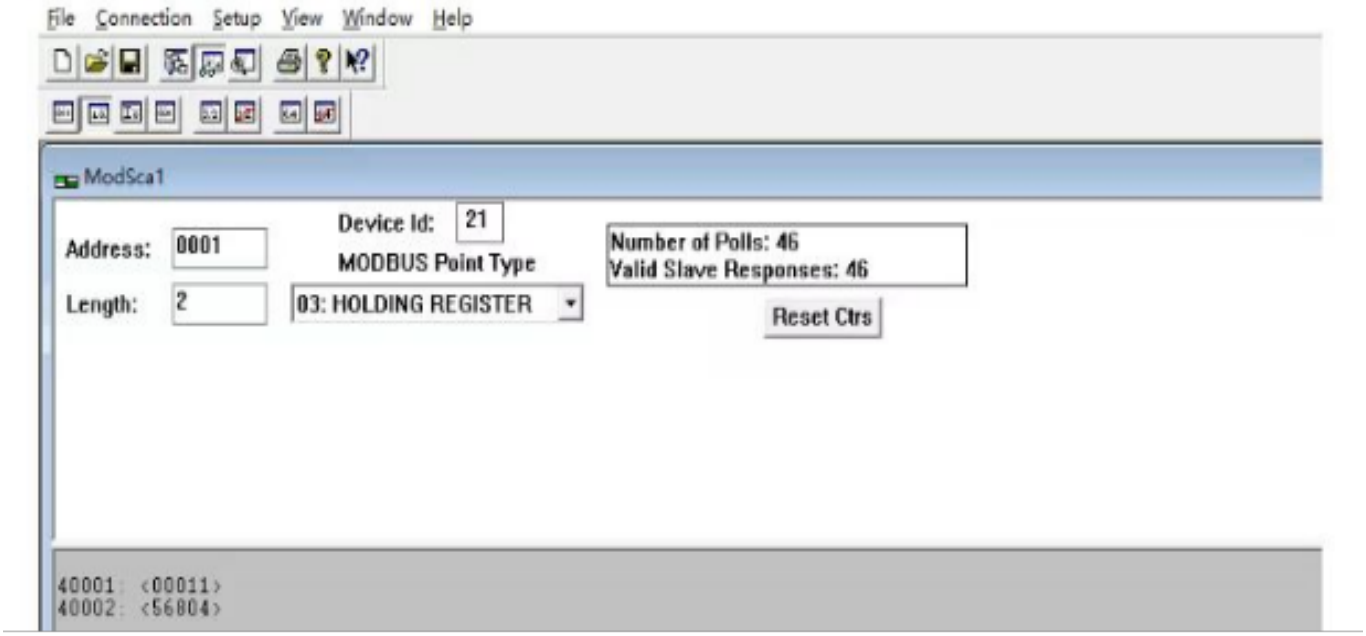
This function allows users to write the data and system parameters in the device. There is no limit on the number of data written by the host at a time, but it cannot exceed the defined address range.
The following example is the basic data wrote to No.63 machine writing coil. The written data is the valve open/close command.

Host send		Message	Message
Address code		63H	63H
Function Code		05H	05H
Starting address	High byte	00H	00H
	Low byte	01H	01H
No. of coil	High byte	00H	00H
	Low byte	FFH	00H
CRC Check code	High byte	D4H	94H
	Low byte	08H	48H
FF = Valve open / 00 = Valve close			

Slave Return		Message	Message
Address code		63H	63H
Function Code		05H	05H
Starting address	High byte	00H	00H
	Low byte	01H	01H
No. of coil	High byte	00H	00H
	Low byte	FFH	00H
CRC Chec code	High byte	D4H	94H
	Low byte	08H	48H

Infrared Meter Reading

Users can be equipped with handheld devices to achieve handheld meter reading, which is used to record accumulated flow rate, operating time, and other information of
the water meter. The meter reading method can refer to the operating instructions of the handheld device



As shown in the above figure:
The value stored in register 40001 is: 00011
The value stored in register 40002 is 56804
Due to the maximum value that these two registers can store being 65536, when
sending this reading, the highest bit is sent first, and then the lowest bit is sent. From this, it can be concluded that the reading of the table is 00011X65536+56804=777700, which has a multiplier of 100 times inside. Dividing the reading by 100 will give the reading of the table as 7777.

*Note:

The software can only access the main register data. Subdial readings are not captured and must be manually checked on the physical water meter.

QUICK REFERENCE GUIDE - WATER METER

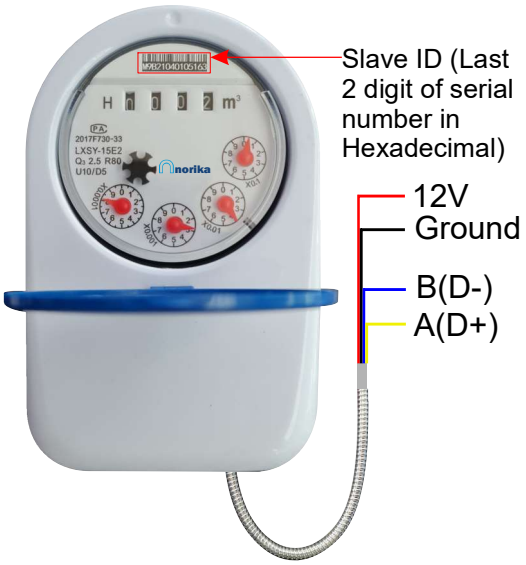
Connections and Pinout

Serial Com Port Settings.

Baud Rate : 9600
Data bits : 8
Parity : None
Stop bits : 1

Connection PinOut

Red : DC 12V
Black : Ground
Blue : Data - (B)
Yellow : Data + (A)



Modbus RTU Communication Protocol

Data Frame Format			
Address Code	Function Code	Data Area	CRC Check Code
1 byte	1 byte	n byte	2 byte

Function Code		
Function Code		Description
1	Read Coil	Read Meter Valve Open/Close
3	Read Register	Read Meter Consumption
5	Write Coil	Execute to Open / Close Valve

Function code 03: Read Water Meter

Host Send		Message		Slave Return		Message
Address code		XXH		Address code		XXH
Function Code		03H		Function Code		03H
				No. of bytes		04H
Starting Address	High byte	00H		Register data	High byte	00H
	Low byte	00H			Low byte	12H
No. of registers	High byte	00H		Register	High byte	D6H
	Low byte	02H		data	Low byte	87H
CRC Check code	High byte	XXH		CRC Check code	High byte	XXH
	Low byte	XXH			Low byte	XXH

Convert
Hex to decimal X 0.01= __m^3

Function code 01: Read Valve Status

Host send		Message		Slave Return		Message	Message
Address code		63H		Address code		63H	63H
Function Code		01H		Function Code		01H	01H
Starting address	High byte	00H		Starting address	High byte	00H	00H
	Low byte	01H			Low byte	01H	01H
No. of coil	High byte	00H		No. of coil	High byte	00H	00H
	Low byte	01H			Low byte	FFH	00H
CRC Check code	High byte	A4H		CRC Check code	High byte	25H	65H
	Low byte	48H			Low byte	C8H	88H
				FF = Valve open / 00 = Valve close			

Function code 05: Write Valve Open/Close

Host send		Message	Message	Slave Return		Message	Message
Address code		63H	63H	Address code		63H	63H
Function Code		05H	05H	Function Code		05H	05H
Starting address	High byte	00H	00H	Starting address	High byte	00H	00H
	Low byte	01H	01H		Low byte	01H	01H
No. of coil	High byte	00H	00H	No. of coil	High byte	00H	00H
	Low byte	FFH	00H		Low byte	FFH	00H
CRC Check code	High byte	D4H	94H	CRC Check code	High byte	D4H	94H
	Low byte	08H	48H		Low byte	08H	48H
FF = Valve open / 00 = Valve close							