Modbus / Industrial Ethernet / MQTT Gateway
GT200-MQ-IE

User Manual

V 3.2

Rev B

SST Automation

Email: support@sstautomation.com

www.SSTAutomation.com

Important Information

Warning

The data and examples in this manual cannot be copied without authorization. SST Automation reserves the right to upgrade the product without notifying users.

The product has many applications. The users must make sure that all operations and results are in accordance with the safety of relevant fields, and the safety includes laws, rules, codes and standards.

Copyright

Trademark

SSTCOMM is the registered trade mark of SST Automation.

Technical Support Contact Information

www.sstautomation.com

www.sstcomm.com

E-mail: support@sstautomation.com

Catalog

1 Product Overview

1.1 Product Function

The GT200-MQ-IE is a RS485 and 10/100M-RJ45 based wireless gateway with GPS function. The data transmission is based on 4G mobile network. The gateway can connect Modbus RTU/ASCII and Modbus TCP devices, or EtherNet/IP PLC to the IoT Cloud by MQTT. It supports the MQTT connection to various IoT cloud platforms, such as Microsoft Azure, Amazon AWS IoT, and customized MQTT Servers.

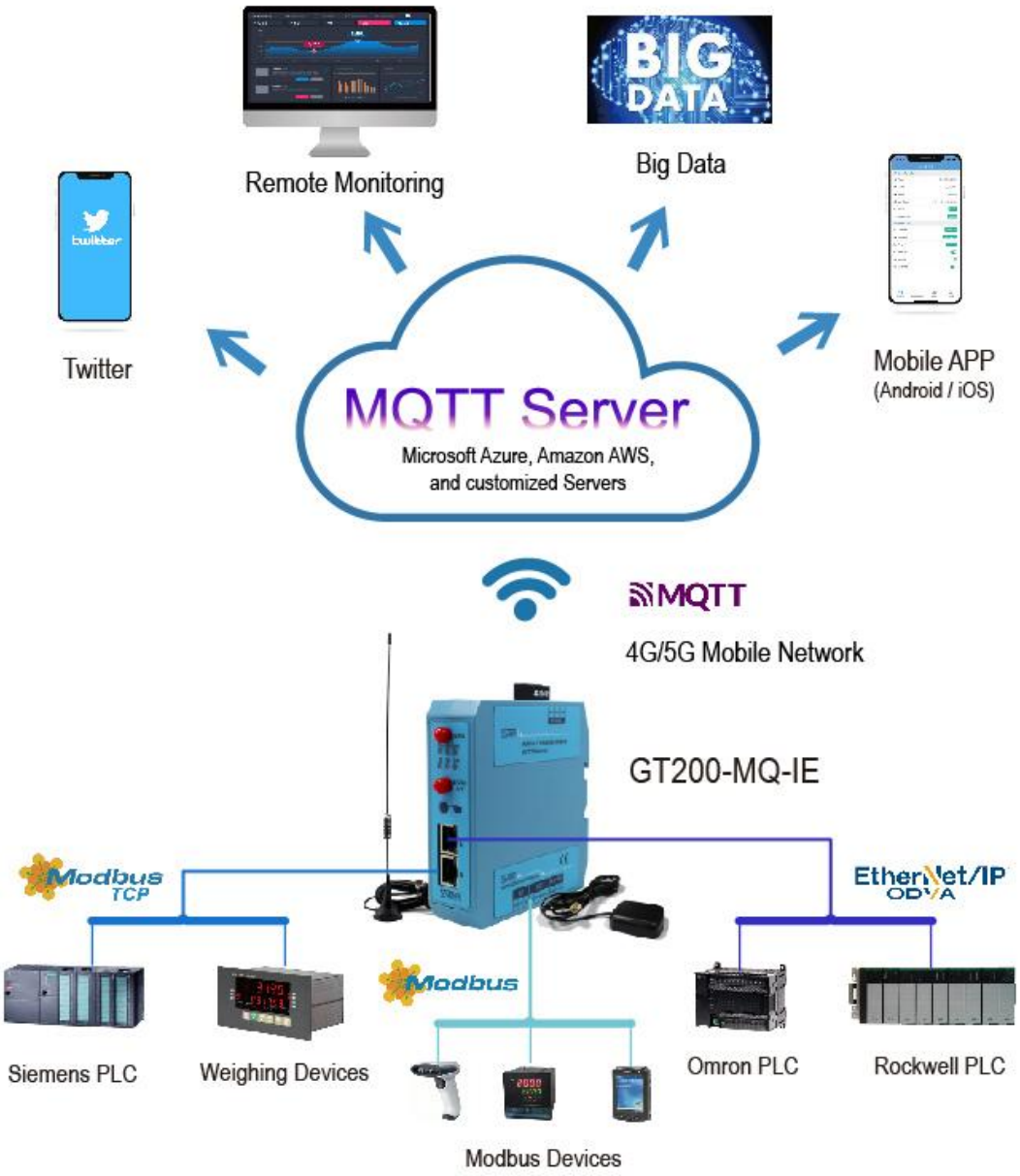
The GT200-MQ-IE integrates three running modes, switched by downloading the BIN files on Appendix B: Change Product Firmware:

Mode	Description
Modbus TCP S / Modbus M/S	Modbus TCP Server, Modbus RTU/ASCII Master/Slave/Transparent Transmission, 2 digital inputs and 2 digital outputs
Modbus TCP C / Modbus M	Modbus TCP Client, Modbus RTU/ASCII Master, 2 digital inputs and 2 digital outputs
EtherNet/IP Adapter	EtherNet/IP Adapter, 2 digital inputs and 2 digital outputs

The GT200-MQ-IE supports different countries/regions mobile network, please specify the SKU when purchasing:

SKU	Supported Country/Region
GT200-MQ-IE-AP	EMEA and SEA
GT200-MQ-IE-NA	USA and Canada
GT200-MQ-IE-MX	Mexico

The GT200-MQ-IE is wildly used in unmanned industrial spots such as water quality monitor stations, environmental monitor stations, intelligent building systems, intelligent agriculture systems, and energy and electricity systems that require remote monitoring.



1.2 Product Feature

- One RS485 Interface - Modbus Master/Slave, Transparent transmission
  - As Modbus Master interface, connects field devices to the IoT Cloud Platform, implementing data acquisition and device control via 4G mobile network.

- ✧ As Modbus Slave interface, integrates PLC, DCS and other Modbus master station with the IoT Cloud Platform via 4G mobile network.
- ✧ The RS485 interface also supports transparent data transmission between filed devices.

➢ Dual Ethernet Interfaces - Modbus TCP Client/Server, EtherNet/IP Adapter

- ✧ As Modbus TCP Client interface, connects Modbus TCP Server devices, such as intelligent instruments.
- ✧ As Modbus TCP Server interface, connects Modbus TCP Client, such as SCADA systems.
- ✧ As EtherNet/IP Adapter interface, connects EtherNet/IP Scanner(Master), such as AB PLC, OMRON PLC, etc.

➢ 2 digital input and 2 digital output I/O interfaces.

➢ Supports GPS.

➢ 4G mobile network is always online. Supports connection detect and auto reconnect functions.

➢ Supports MQTT broker URL connection.

➢ SSL V3.0 and TLS V1.0/V1.1/V1.2. Supports custom Client ID.

➢ QoS 0 and 1 supported.

➢ Supports two publish mode: publish when data changes or publish periodically at custom cycle time, which can reduce the data traffic.

➢ Supports two Message Types: register message or property message.

➢ Real-time device status monitoring on the IoT Cloud Platform.

➢ 4G network quality detection and display.

➢ Applies to industrial field with high reliability. Supports real-time monitoring and automatic reset.

1.3 Technical Specifications

[1] 4G/3G/2G mobile network

- ◆ Supported standard:
  - ● GT200-MQ-IE-AP: FDD LTE: B1/B3/B7/B8/B20/B28A
    TDD LTE: B38/B40/B41
    WCDMA: B1/B8

GSM/EDGE: B3/B8

- GT200-MQ-IE-NA&GT200-MQ-IE-MX:

  FDD LTE: B2/B4/B5/B12/B13/B14/B66/B71

  WCDMA: B2/B4/B5

Note: Some wireless network carriers require exact APN settings, which is mandatory when configuring the GT200-MQ-IE. Please refer to the carriers' support information. For example, the SIM card of AT&T will work with the following settings:

APN: NXTGENPHONE

APN Username & Password: Not set (leave as default)

◆ Transmission speed:

- GT200-MQ-IE-AP: FDD LTE (Mbps): 150(DL)/50(UL)

  FDD LTE(Mbps): 150(DL)/50(UL)

  TDD LTE(Mbps): 130(DL)/30(UL)

  WCDMA(Kbps): 384(DL)/384(UL)

  EDGE(Kbps): 296(DL)/236.8(UL)

  GPRS(Kbps): 107(DL)/86.5(UL)

- GT200-MQ-IE-NA&GT200-MQ-IE-MX:

  FDD LTE(Mbps): 150(DL)/50(UL)

  WCDMA(Kbps): 384(DL)/384(UL)

[2] One serial interface

◆ Physical standard: RS485, half-duplex.

◆ Parity: Odd, Even, None, Mark, Space.

◆ Stop bits: 1 or 2.

◆ Baud rate: 600~115200 bps.

◆ Protocol: Modbus RTU/ASCII Master/Slave, Transparent transmission.

◆ Modbus command: 01, 02, 03, 04, 05, 06, 15, 16.

◆ Interface type: 3.81 mm 3-pin port.

◆ Galvanic isolation: 1kV

[3] 2 Ethernet interfaces available simultaneously

- Physical standard: RJ-45, full-duplex and half-duplex.
- Speed: 10/100 Mbps, auto-detected.
- Protocol: Modbus TCP Client/Server (supports command: 01, 02, 03, 04, 05, 06, 15, 16)

  EtherNet/IP Adapter (Slave)
- Supports DHCP.

[4] Maximum data and connections:

- Modbus RTU/ASCII:

  Input bytes ≤ 512 bytes

  Output bytes ≤ 512 bytes

  Maximum connections: 31 serial devices

  Maximum commands: 100 commands (as Modbus master)
- Modbus TCP:

  Input bytes ≤ 512 bytes

  Output bytes ≤ 512 bytes

  Maximum connections: 4 Clients or 4 Servers

  Maximum commands: 100 commands (as Modbus TCP Client)
- EtherNet/IP:

  Input bytes ≤ 492 bytes

  Output bytes ≤ 492 bytes
- Maximum properties:

  Mode Modbus TCP S / Modbus M/S and Mode Modbus TCP C / Modbus M: 1000 properties.

  Mode EtherNet/IP Adapter: No limits.

[5] 2 Digital inputs

- Port: 3.81mm 3-pin port.
- Supported DI contact: Dry contact.

[6] 2 Digital outputs:

- Port: 3.81mm 4-pin port.

- Maximum output: 30 VDC / 220 VAC, 0.9A

Note: The digital outputs are available only with at least 24V DC power on.

[7] Power supply: 24 VDC (11~30 VDC)

[8] Operating temperature: -20°C~60°C (-4°F~140°F). Humidity: 5%~95% (non-condensing).

[9] External dimensions (W*H*D): 1.33 in*4.56 in*4.21 in (34mm*116mm*107mm).

[10] Installation: 1.38 in (35 mm) DIN RAIL.

[11] Protection level: IP20.

1.4 Related Products

The related products include: IOT600-TWX-TS, GT200-HT-MT, GT200-HT-RS and GT200-HT-EI etc.

To get more information about related products, please visit SSTCOMM website: www.sstautomation.com.

1.5 Revision History

Revision	Date	Chapter	Description
V3.2	3/2/2021	ALL	New release
V3.2, Rev A	4/1/2021	Chapter 1, 5	Revision
V3.2, Rev B	5/21/2021	Chapter 1	Picture updated and revision

2 Hardware Descriptions

2.1 Product Appearance

Note: This picture is for reference only. The product appearance is subject to the actual product.

2.2 LED Indicators

LED	State	State description
STA	OFF	4G module is not started.
	Green, slow blinking (200 ms ON / 1800 ms OFF)	The gateway is searching available network.
	Green, slow blinking (1800 ms ON / 200 ms OFF)	The 4G module is in standby.
	Green, quick blinking	Data transmission via 4G mobile network.
SG	OFF	The SIM card is invalid.
	Green, ON	The connection is being established or could not be established.
	Green, slow blinking (1000 ms ON / 1000 ms OFF)	Poor signal
	Green, blinking (500 ms ON / 500 ms OFF)	Weak signal
	Green, quick blinking (200 ms ON / 200 ms OFF)	Strong signal
TX/RX	Green, blinking	No data transmission on RS485 interface.
	Green	Data transmission on RS485 interface.
DI(DO)	OFF	The digital inputs (outputs) are not connected.
	Red	The DI1 (DO1) is connected.
	Green	The DI2 (DO2) is connected.
	Orange	The DI1 (DO1) and DI2 (DO2) are connected.
ENS	Green, slow blinking	No Modbus TCP or EtherNet/IP connection.
	Green, quick blinking	Fixed IP Address: 192.168.0.10
	Red, blinking	Locating the gateway.
	Green	Normal connections.

2.3 Configuration Button

After powering on the GT200-MQ-IE, long press the button for 5 seconds. The IP address will then be set to 192.168.0.10 and the ENS indicator will quickly blink green. This function can fix the IP address when the

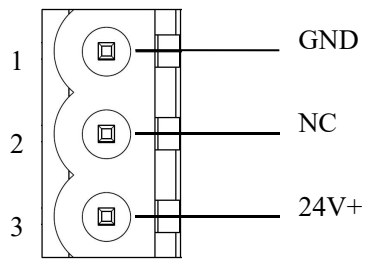
GT200-MQ-IE can't be searched in normal status. Press the button again, the GT200-MQ-IE will be restarted and begin operating normally.

After powering on the GT200-MQ-IE, double press the button to disable uploading/downloading configuration. The GT200-MQ-IE can be searched but can't be configured. This function can prevent the wrong operations. Press the button again to set back to normal status.

Button action	Description
Long press for 5 seconds	The IP address is set to 192.168.0.10 and the ENS is blinking green quickly. Press the button again to set back to normal status.
Double press	Uploading/downloading configuration is disabled. The GT200-MQ-IE can be searched but can't be configured. Press the button again to set back to normal status.

2.4 Interface

2.4.1 Power Interface



Pin	Function
1	Power ground
2	Not connected
3	24V+, DC

2.4.2 RS-485 Serial Interface

The GT200-MQ-IE has a RS-485 serial interface (Under mode "EtherNet/IP Adapter...", this interface is disabled).

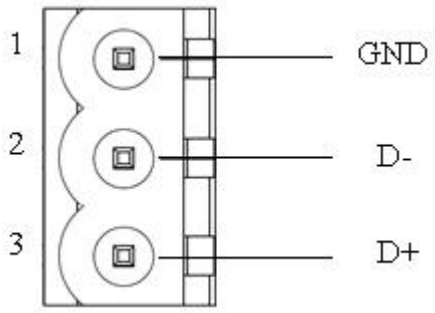
RS-485 Specification:

1. Network topology: Linear bus with active terminal resistors at both sides.
2. Baud rate: 600 bps ~115.2K bps.

3. Physical Media: Shielded twisted-pair cable that also can cancel the shielding, depending on environmental conditions (EMC).
4. Station number: 31 stations per section (without repeater), and up to 127 stations in total (with repeaters).
5. Plug connection: 3-pin port.

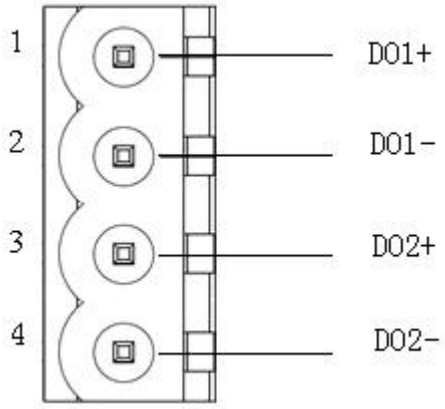
RS-485 Installation:

1. All stations are connected with RS-485 bus.
2. Up to 31 stations can be connected per section.
3. Connect a terminal resistor (120Ω, 1/2W) in parallel at both ends of the communication lines to ensure stable communication.
4. The interface is a 3-pin port and the wiring is as follows:



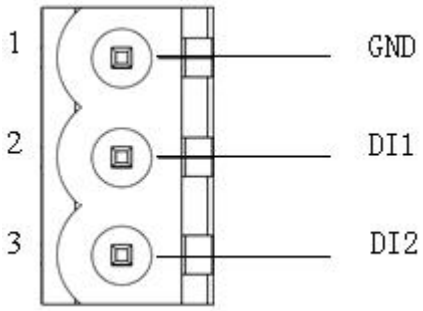
Pin	Function
1	GND
2	D-
3	D+

2.4.3 Digital Output



Pin	Function
1	DO1+
2	DO1-
3	DO2+
4	DO2-

2.4.4 Digital Input



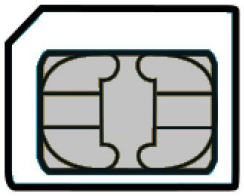
Pin	Function
1	GND
2	DI1
3	DI2

2.4.5 Ethernet Interface

The Ethernet interface is a RJ-45 socket.

Pin	Signal description
S1	TXD+
S2	TXD-
S3	RXD+
S4	-
S5	-
S6	RXD-
S7	-
S8	-

2.4.6 Micro SIM Card Slot



Format: Micro-SIM (3FF)

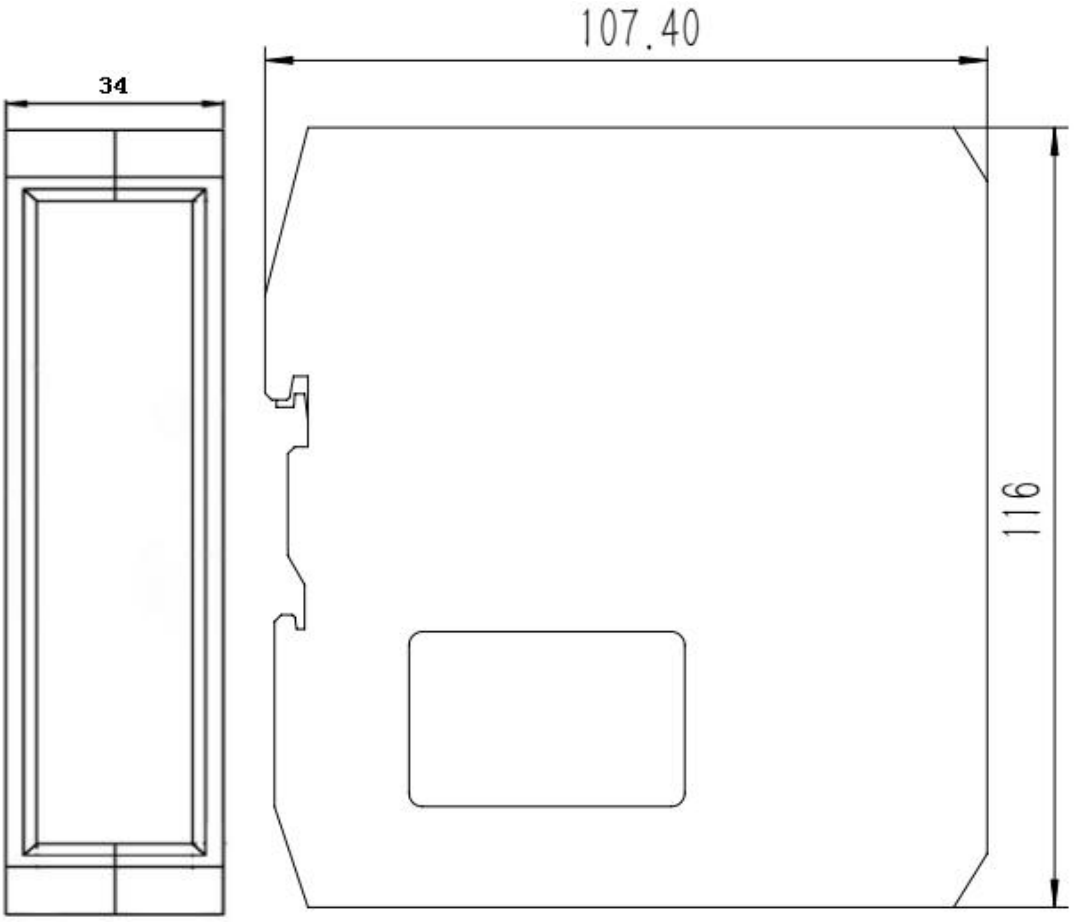
Length * Width: 15 mm * 12 mm (0.59 in * 0.47 in)

Thickness: 0.76 mm (0.030 in)

3 Hardware Installation

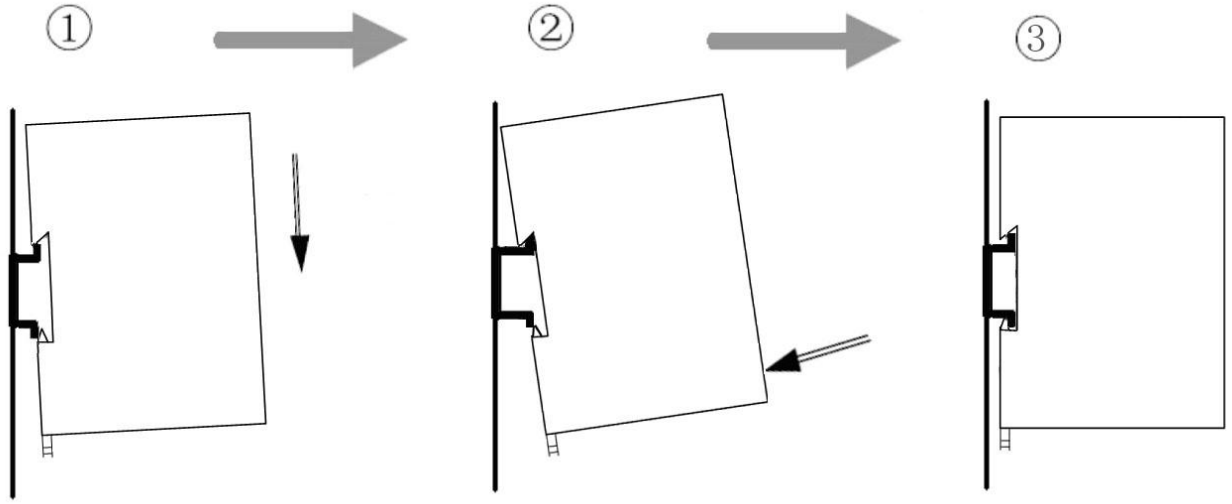
3.1 Mechanical Dimensions

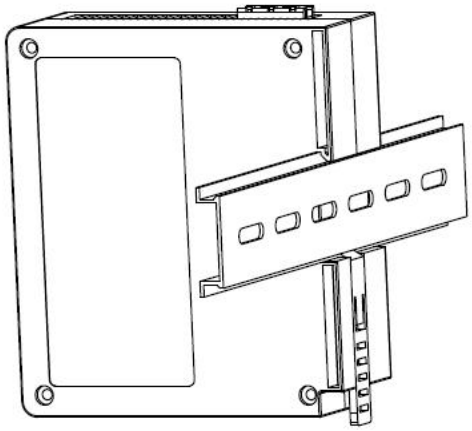
Size: 34mm * 116mm * 107.4mm / 1.34 in * 4.57 in * 4.22 in (width * height * depth)



3.2 Installation Method

Use 13.8 in (35 mm) DIN Rail.


①
②
③

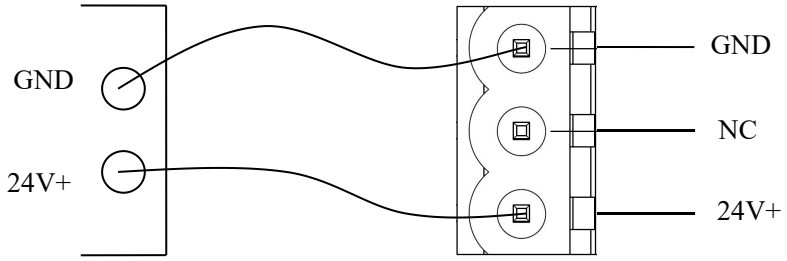


4 Quick Start Guide

4.1 Connection

1. Insert the Micro SIM card into the slot behind the GT200-MQ-IE correctly.
2. Connect the power supply as follows.

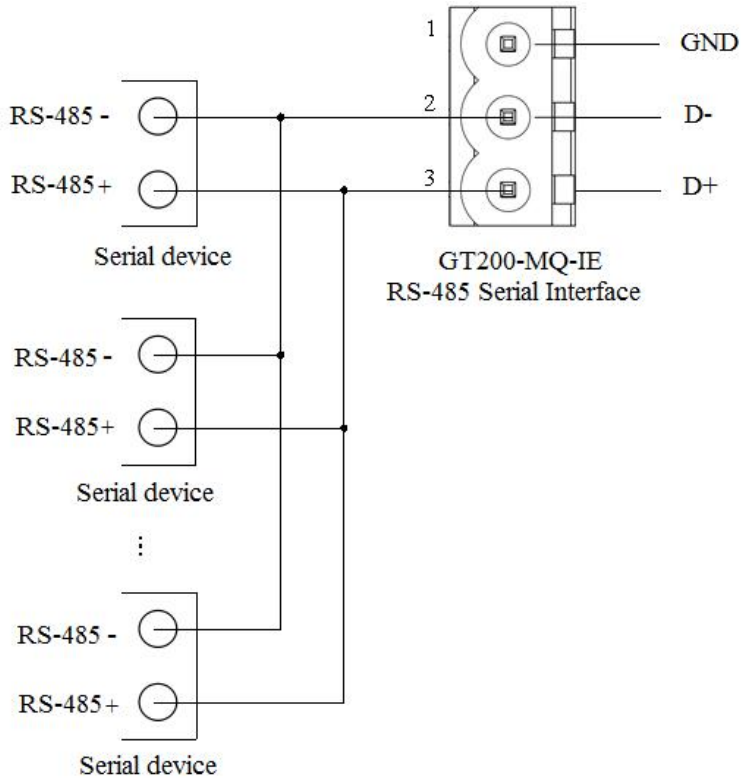
   Note: Please do not power on the devices before finishing all the connections.



24V DC Power GT200-MQ-IE Power Interface

3. Connect the serial devices. (Under EtherNet/IP Adapter mode, the serial interface is disabled.)

   Note: The GT200-MQ-IE doesn't have the terminal resistor in the RS-485 serial interface. When connecting multiple RS-485 serial devices, it's recommended to connect a terminal resistor (120Ω, 1/2W) in parallel at both ends of the communication lines to ensure stable communication.



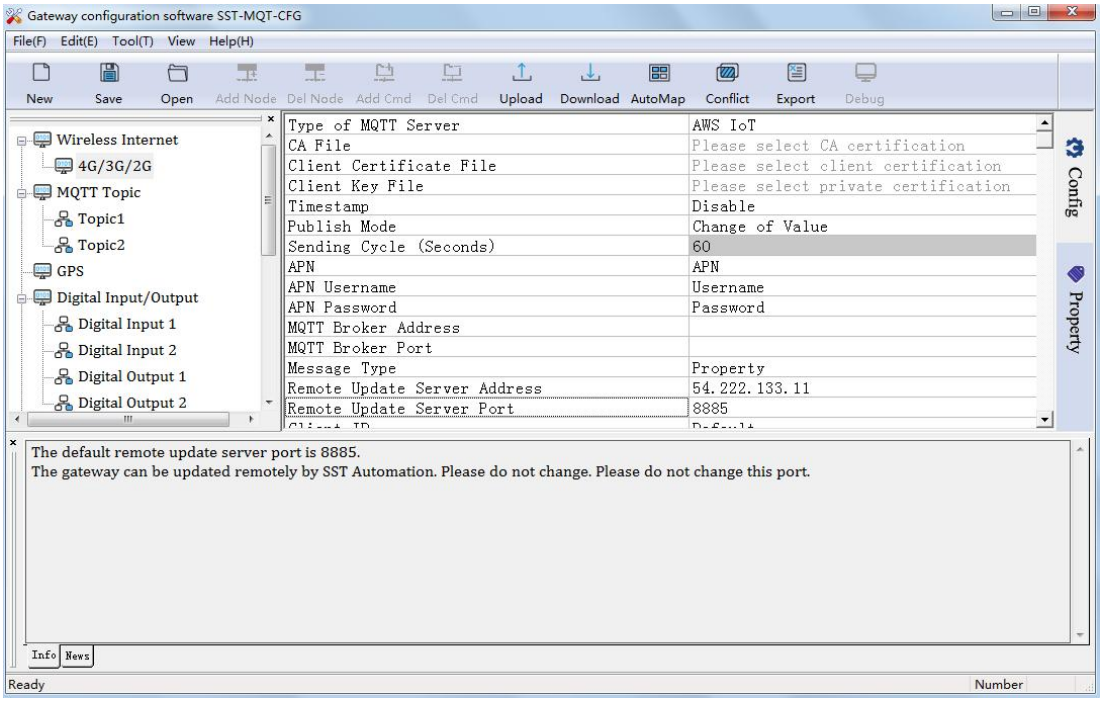
4. Connect the Ethernet/IP scanner (master) or Modbus TCP clients/servers via Ethernet cable.
5. Connect the GT200-MQ-IE with the PC via Ethernet cable.
6. Power on the GT200-MQ-IE.

4.2 Configuration

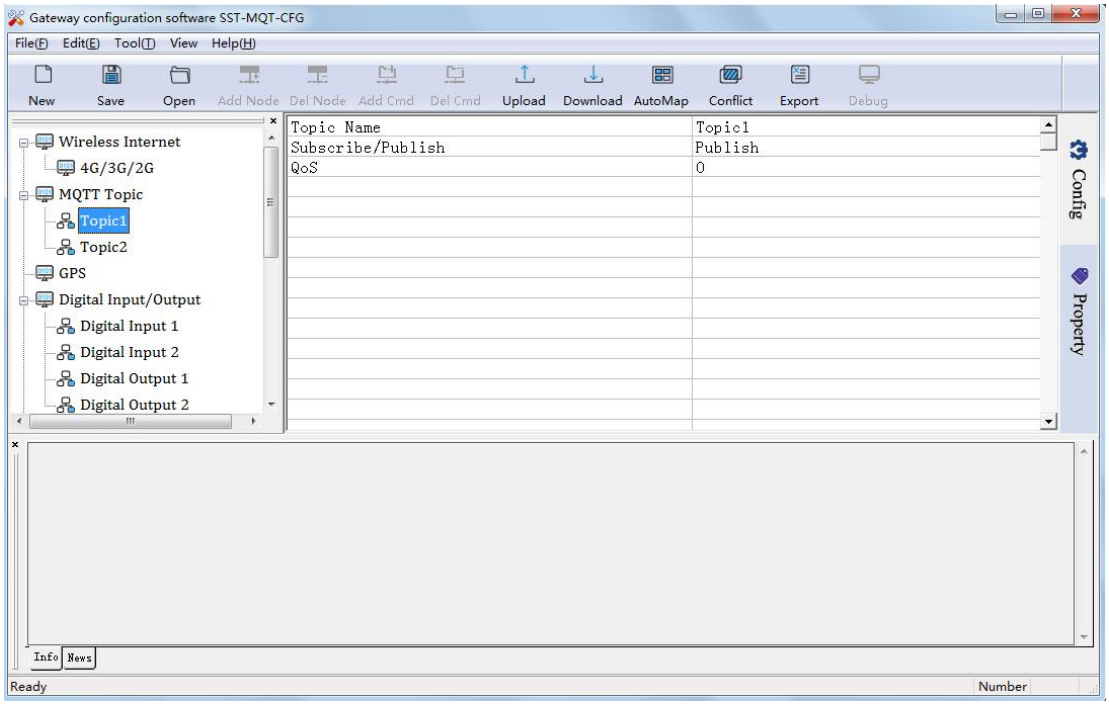
Please see "SST-MQT-CFG Software Content" for details.

1. Download the configuration software SST-MQT-CFG from www.sstautomation.com and install it. Open the software and select a device.

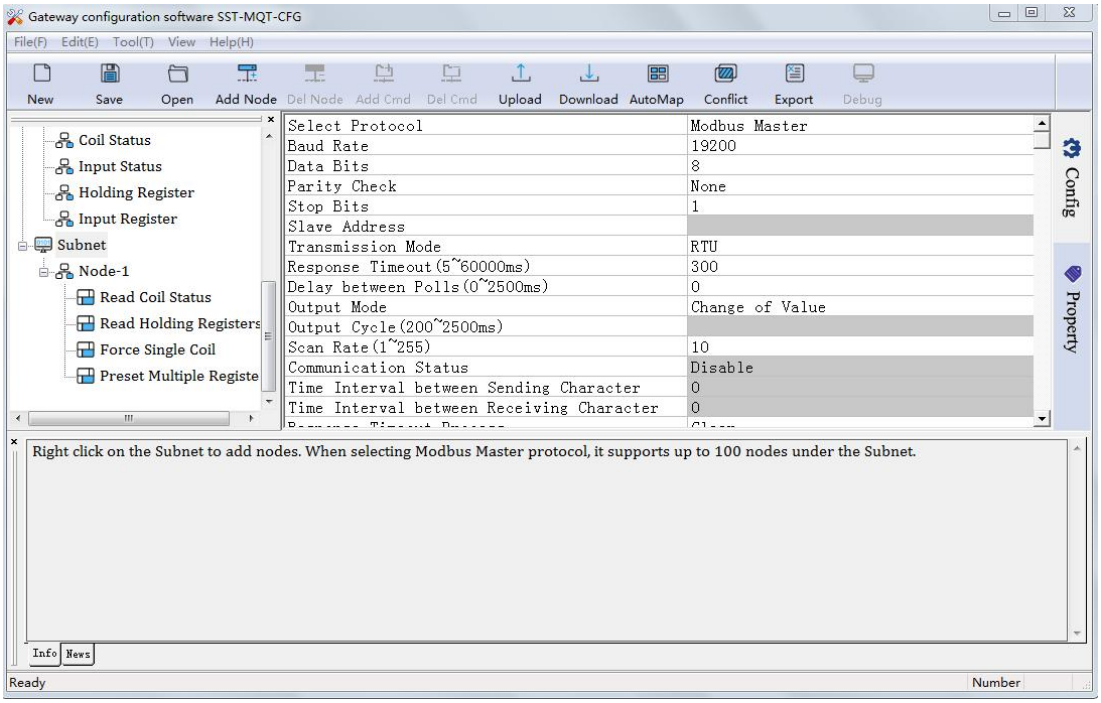
2. Configure the Wireless Internet (MQTT Server configuration).



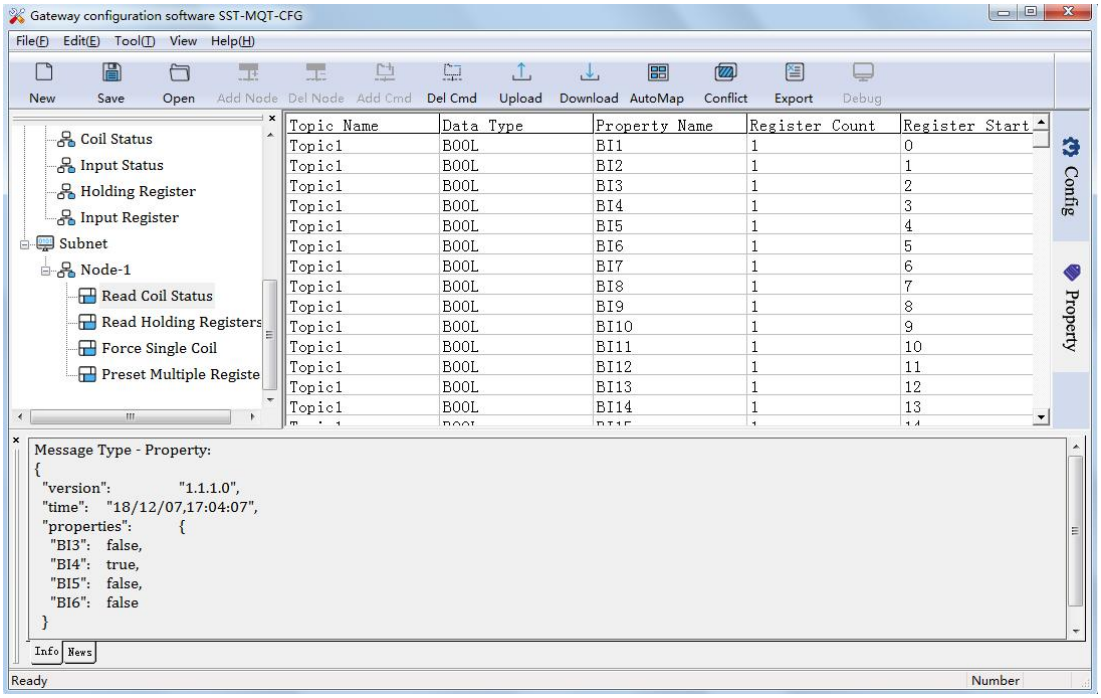
3. Add or edit MQTT Topic.



4. Configure the Digital Input/Output, Modbus TCP, Subnet and EtherNet/IP Interface parameters.



5. Configure the properties.



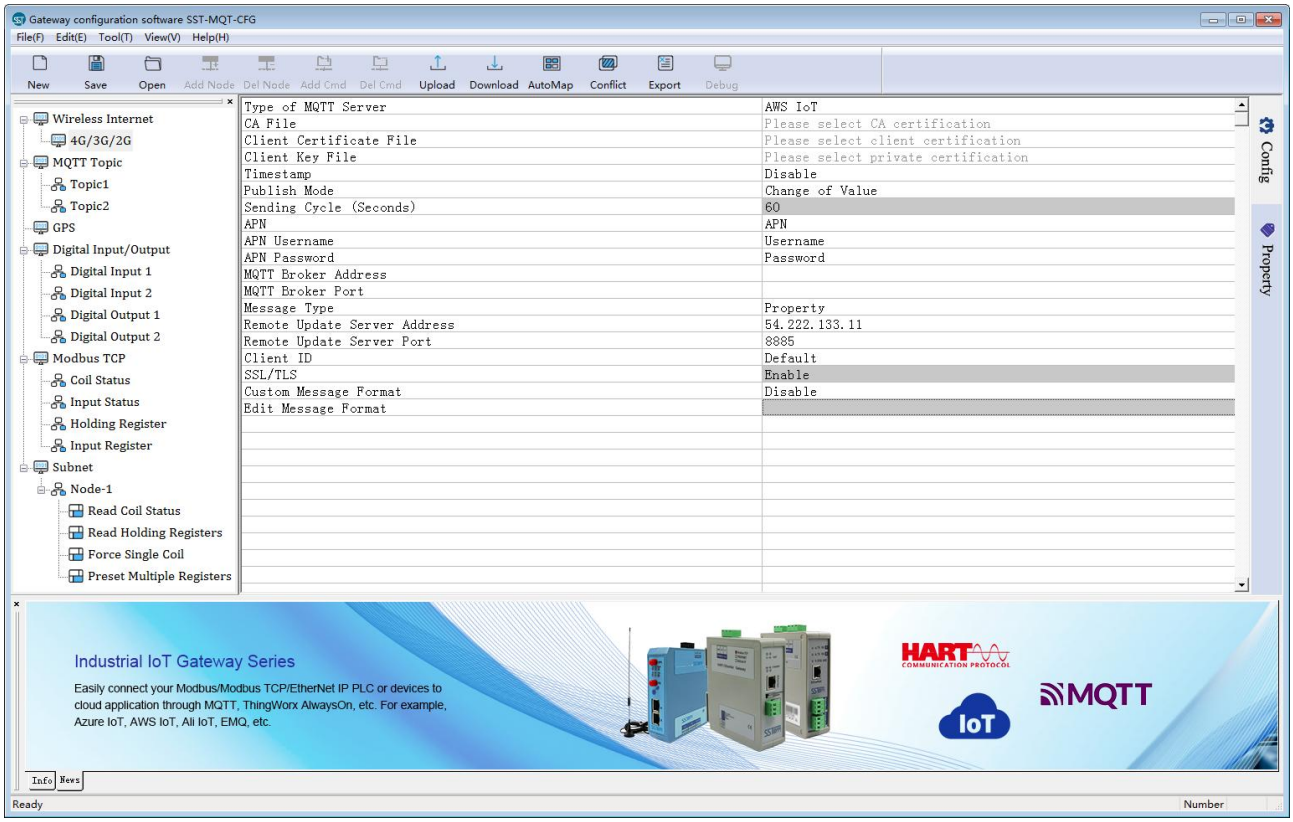
6. Check the mapping buffer and the property names. Use the “Auto Mapping” and “Name Properties in Order” functions.

7. Save the configuration project and download it to the GT200-MQ-IE.

5 Configuration Software SST-MQT-CFG

SST-MQT-CFG is the configuration software which can be used to configure GT200-MQ-IE. It is based on Windows OS. Download the software on www.sstautomation.com and run the setup program to begin the installation. Please follow the prompts to install the software.

For detailed software content, please see "SST-MQT-CFG Software Content". Or open the SST-MQT-CFG, select "Help" >> "Content" on the menu bar.

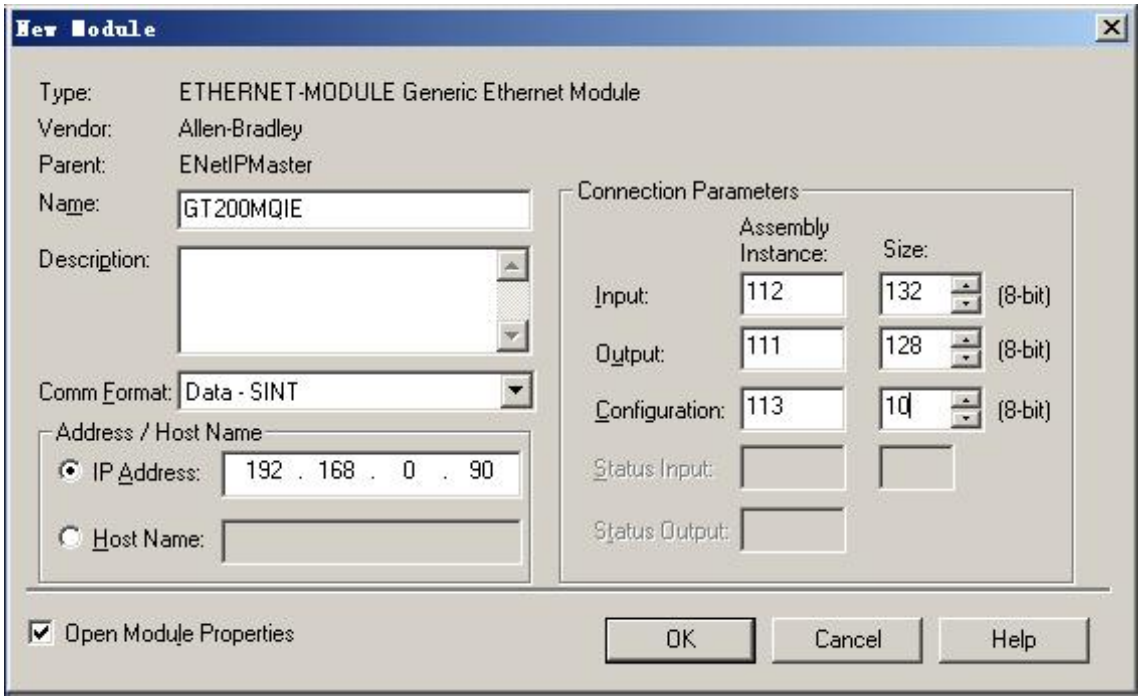


6 EtherNet/IP Connection Parameters

The GT200-MQ-IE supports the following connection parameters:

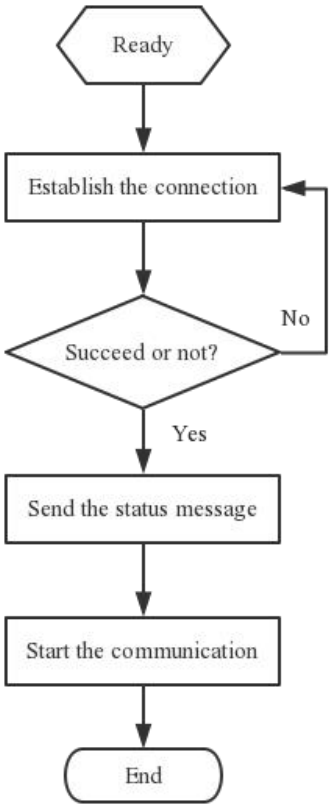
Parameters \ Data Size	128 Bytes	256 Bytes	492 Bytes
Input Instance	102	112	122
Output Instance	101	111	121
Configuration Instance	103	113	123

To configure the Connection Parameters of an EtherNet/IP Module, please see 8.1 I/O Configuration.



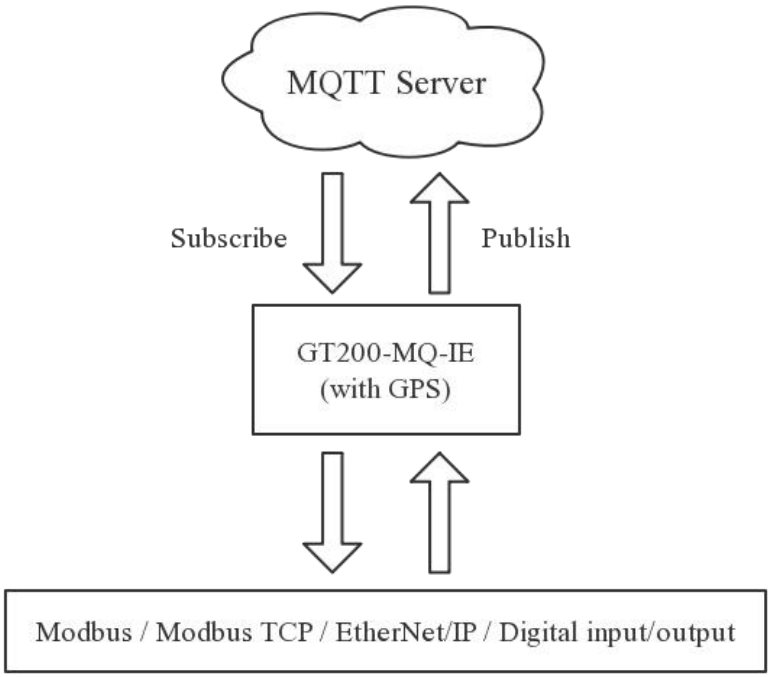
7 Working Principle

7.1 Connection Process



7.2 Data Exchange

The GT200-MQ-IE is able to connect Modbus RTU/ASCII, Modbus TCP, Digital Input/output and EtherNet/IP devices to MQTT Servers, such as Microsoft Azure, Amazon AWS IoT, and customized MQTT Servers.

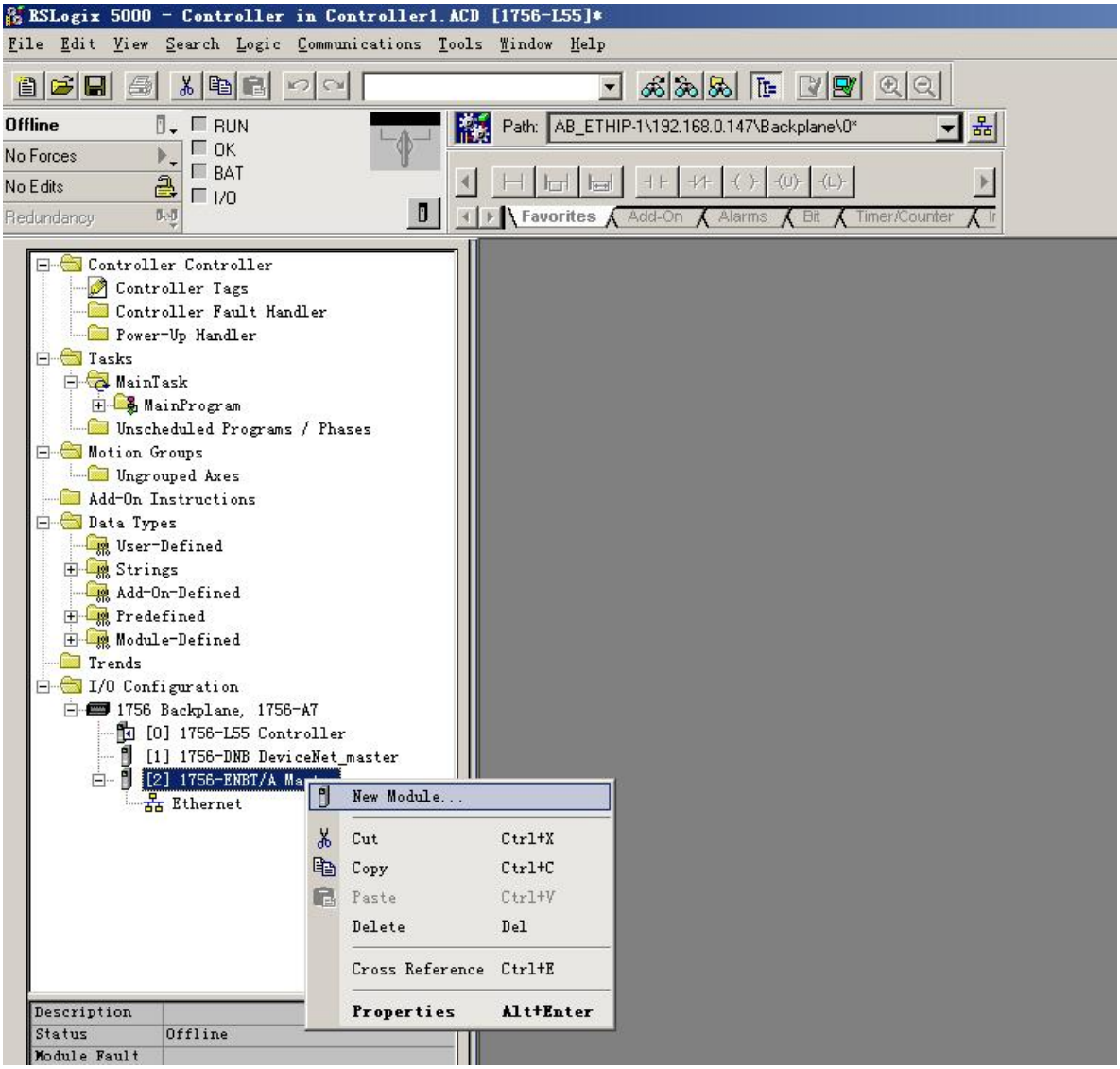

MQTT Server
Subscribe
Publish
GT200-MQ-IE
(with GPS)
Modbus / Modbus TCP / EtherNet/IP / Digital input/output

8 Read/Write Data in EtherNet/IP Network

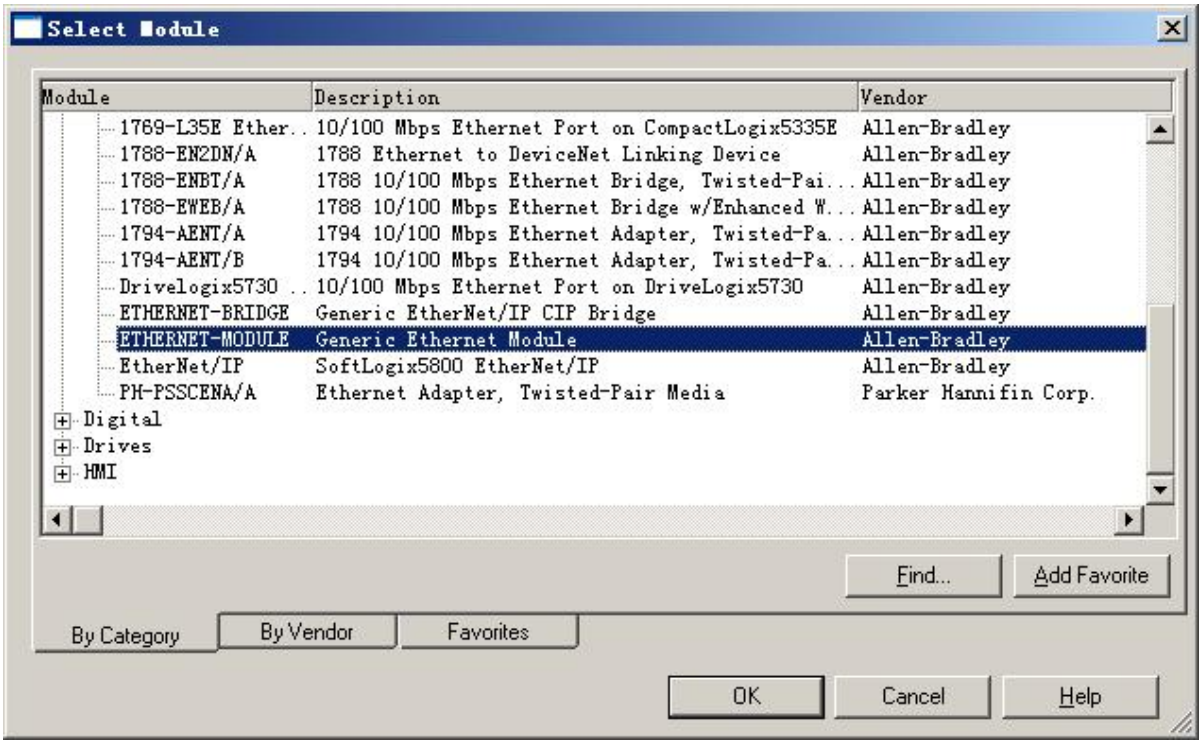
Take RSLogix 5000 for example.

8.1 I/O Configuration

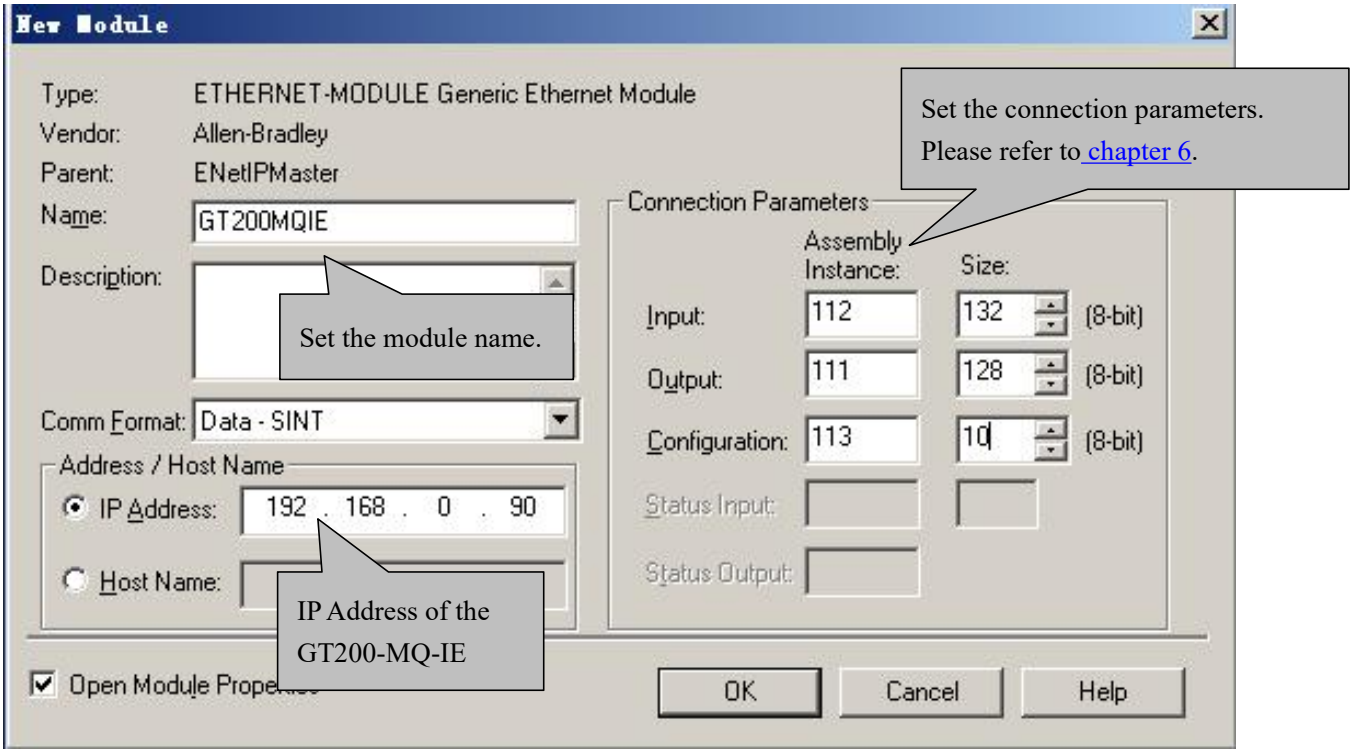
Right click on the EtherNet/IP master module and select “New Module”:



In the pop-up interface, click “+” to unfold the “Communications” and select “ETHERNET-MODULE”. Click “OK”.



Configure the module parameters:



The above parameters need to be configured are:

Name - The module name of the EtherNet/IP slave module.

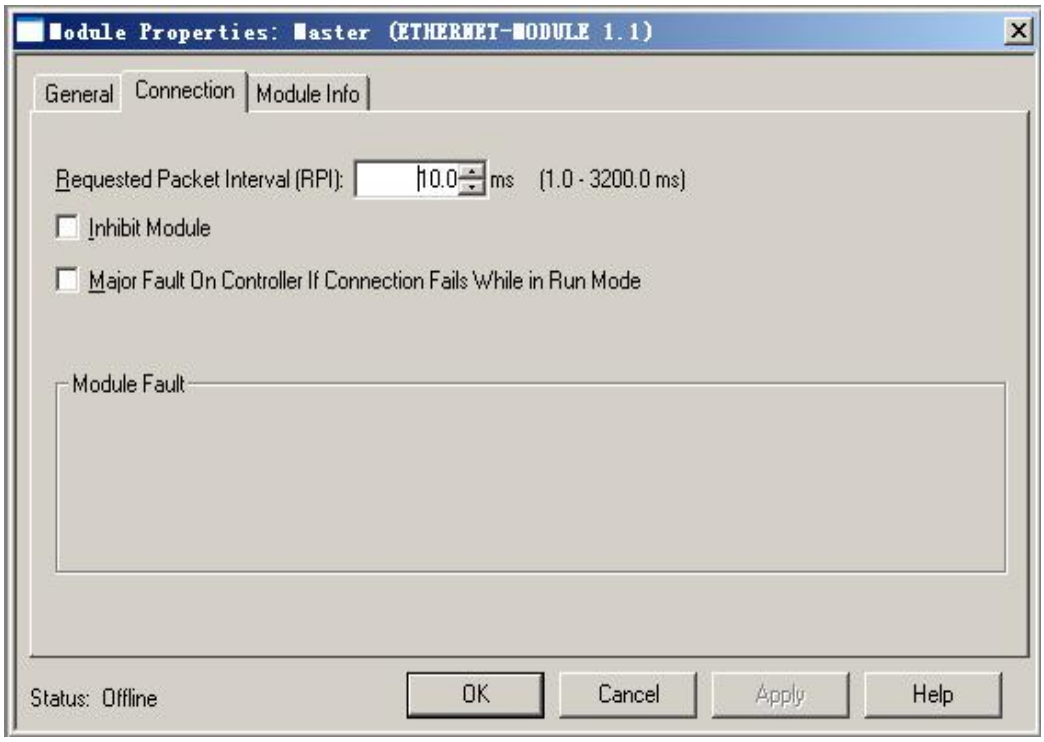
Comm Format - The data type. Users can choose DINT, INT, SINT and REAL, etc. The comm format can't be

changed after creating the module. If you want to change the data type, please delete the module and add a new one.

IP Address -The IP address of the EtherNet/IP slave module.

Connection Parameters - The connection parameters are used for I/O communication. The Size should be consistant with the corresponding input/output/configuration bytes. Please refer to chapter 6.

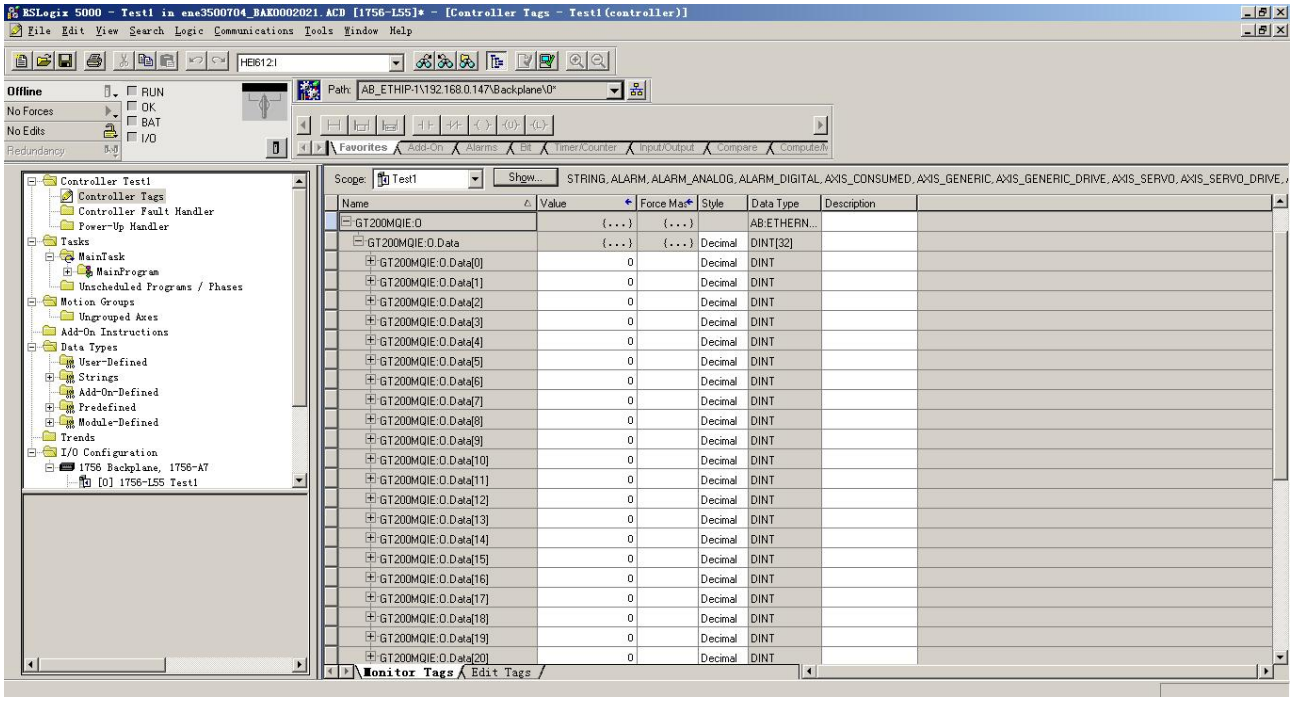
Click “OK”. Set the Module Properties:



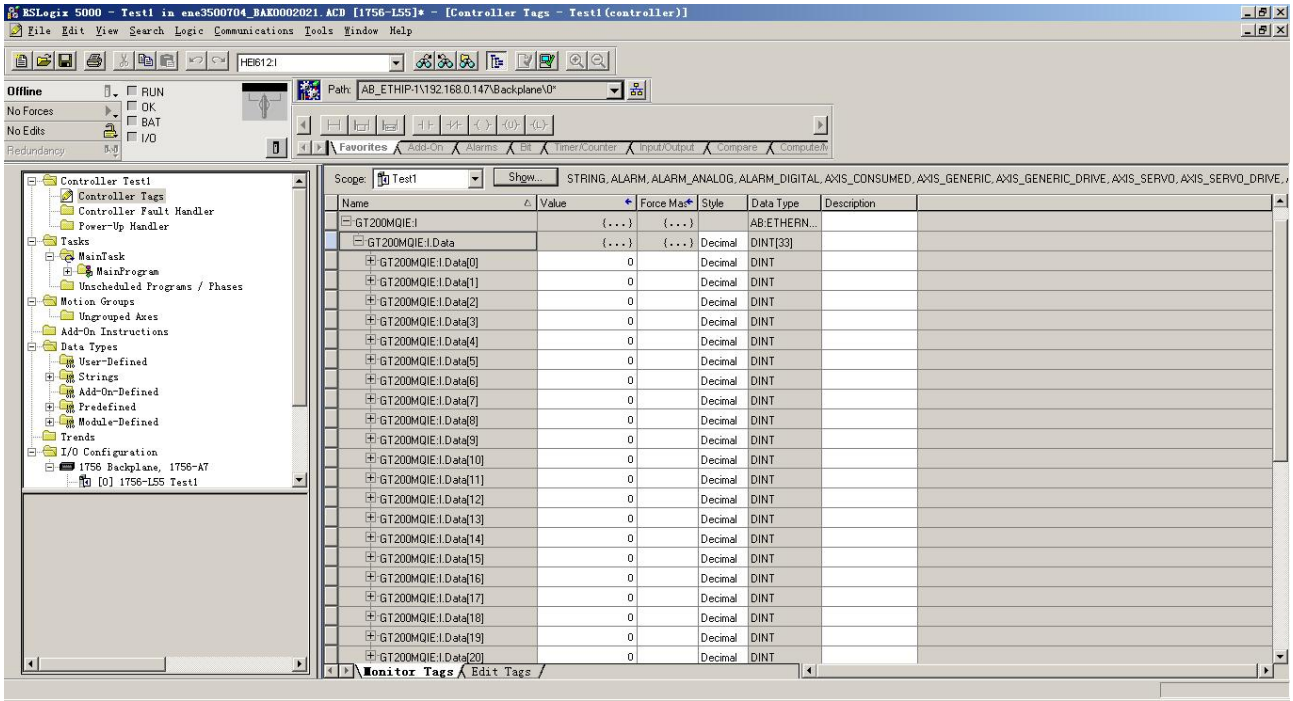
Click “OK” to save the configuration.

Double click on the “Controller Tags” to open the “Monitor Tags”, as below.

The GT200MQIE:O.Data[0]~[127] are the output data of the EtherNet/IP master.



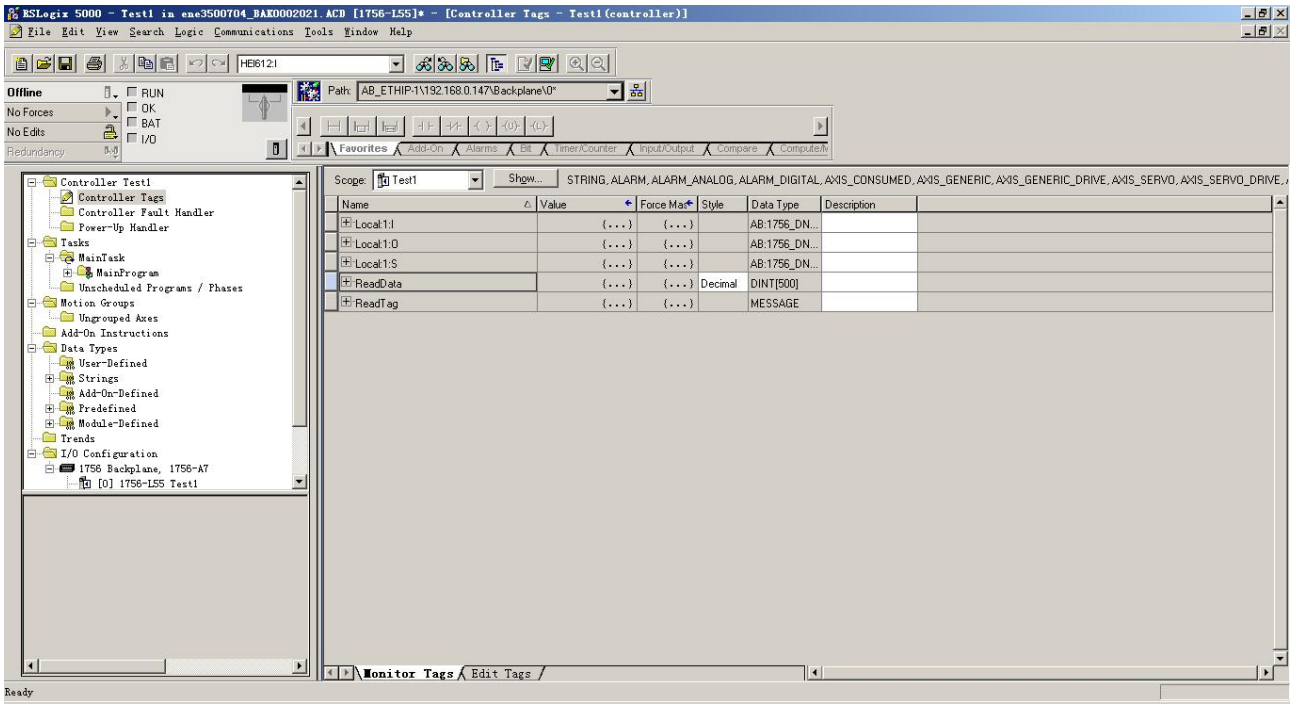
The first 4 bytes are the real-time frame head of the EtherNet/IP slave. The other are the input data.



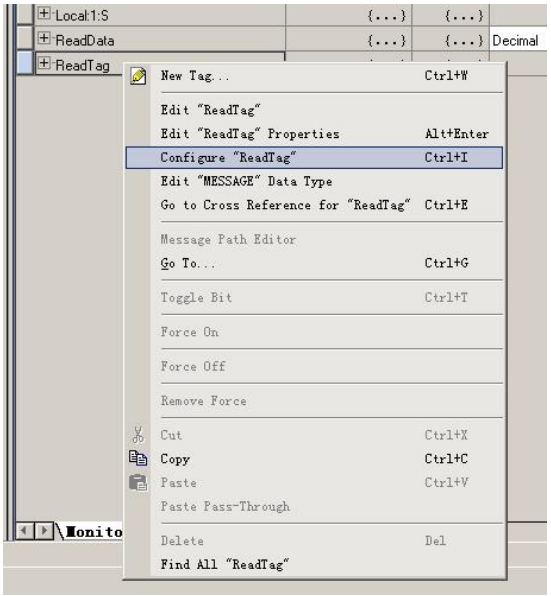
8.2 MSG Configuration

8.2.1 Read Data

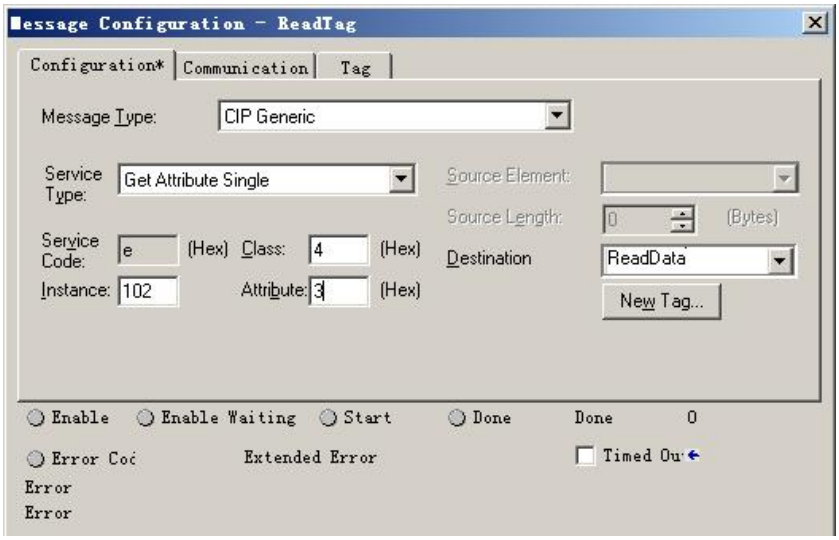
Add two new tags in the “Controller Tags”. In this example, name the tags as “ReadTag” and “ReadData”. Set the ReadTag type as “MESSAGE” and the ReadData type as “DINT[500]”.



Right click on the ReadTag and select Configure “ReadTag”.



Configure the parameters in the pop-up window:



In “Configuration*” page:

Message Type: Select “CIP Generic”.

Service Type: Select “Get Attribute Single” and the Service Code will be set to “e (HEX)”.

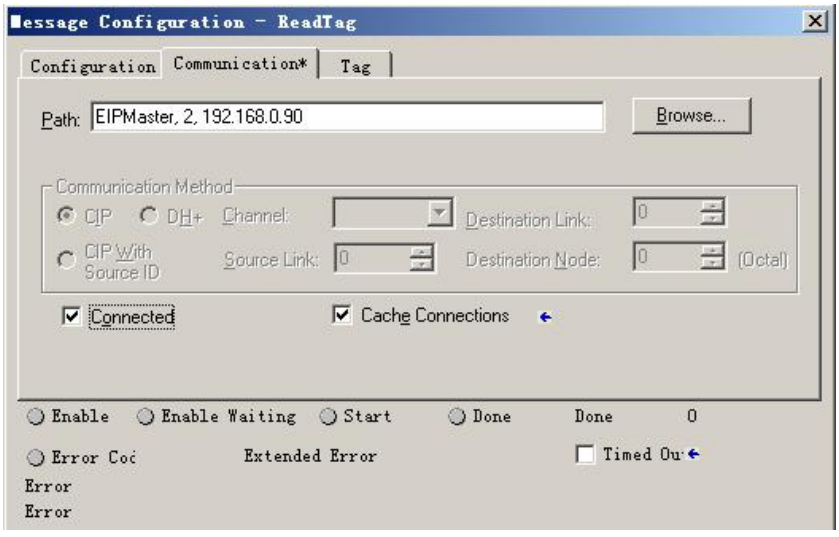
Class: 4

Instance: 102 (128 bytes), 112 (256 bytes) and 122 (492 bytes) are optional.

Attribute: 3 (Hex)

Destination: Select “ReadData” and all the read(input) data will be stored in the ReadData tag.

Jump to the “Communication” page.



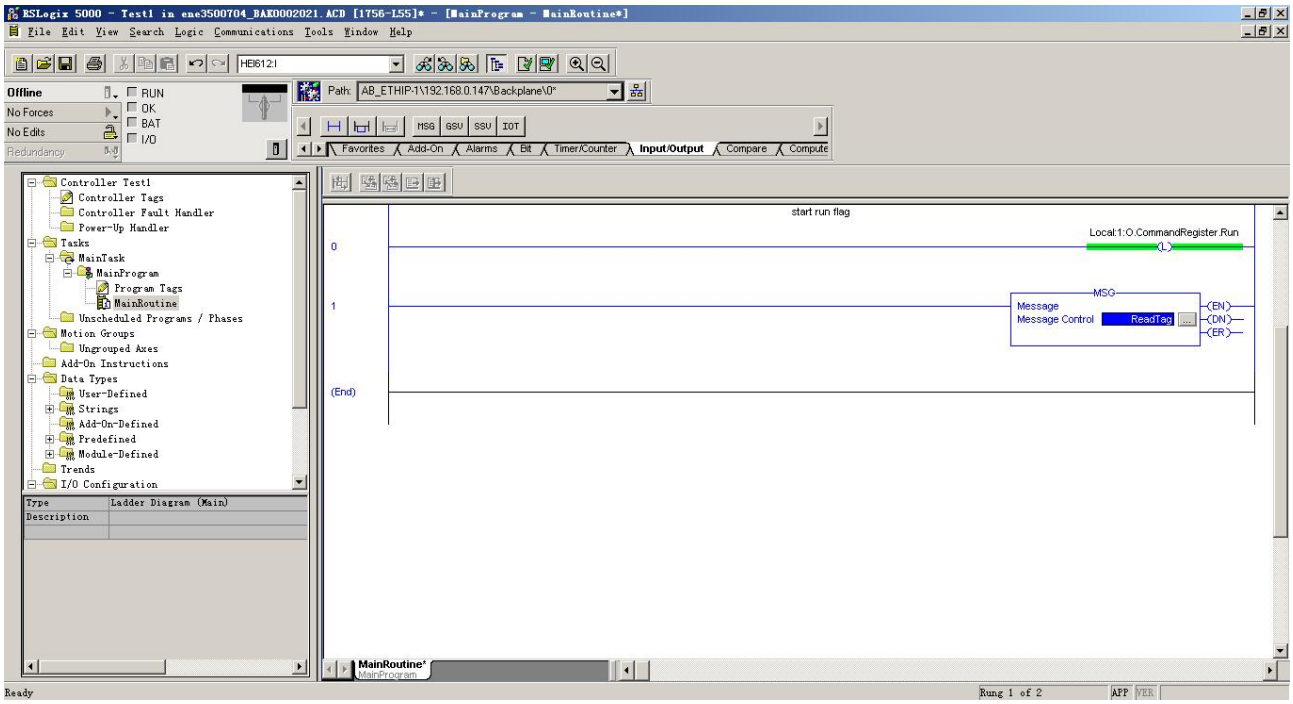
Input the EtherNet/IP slave path. The path format is “EtherNet/IP master name”, “slot number”, “EtherNet/IP

slave IP address". In this example, the EtherNet/IP master name is "EIPMaster", the slot number is "2" and the EtherNet/IP slave (GT200-MQ-IE) IP address is 192.168.0.90.

Click "OK" or "Apply" to save and apply the configuration.

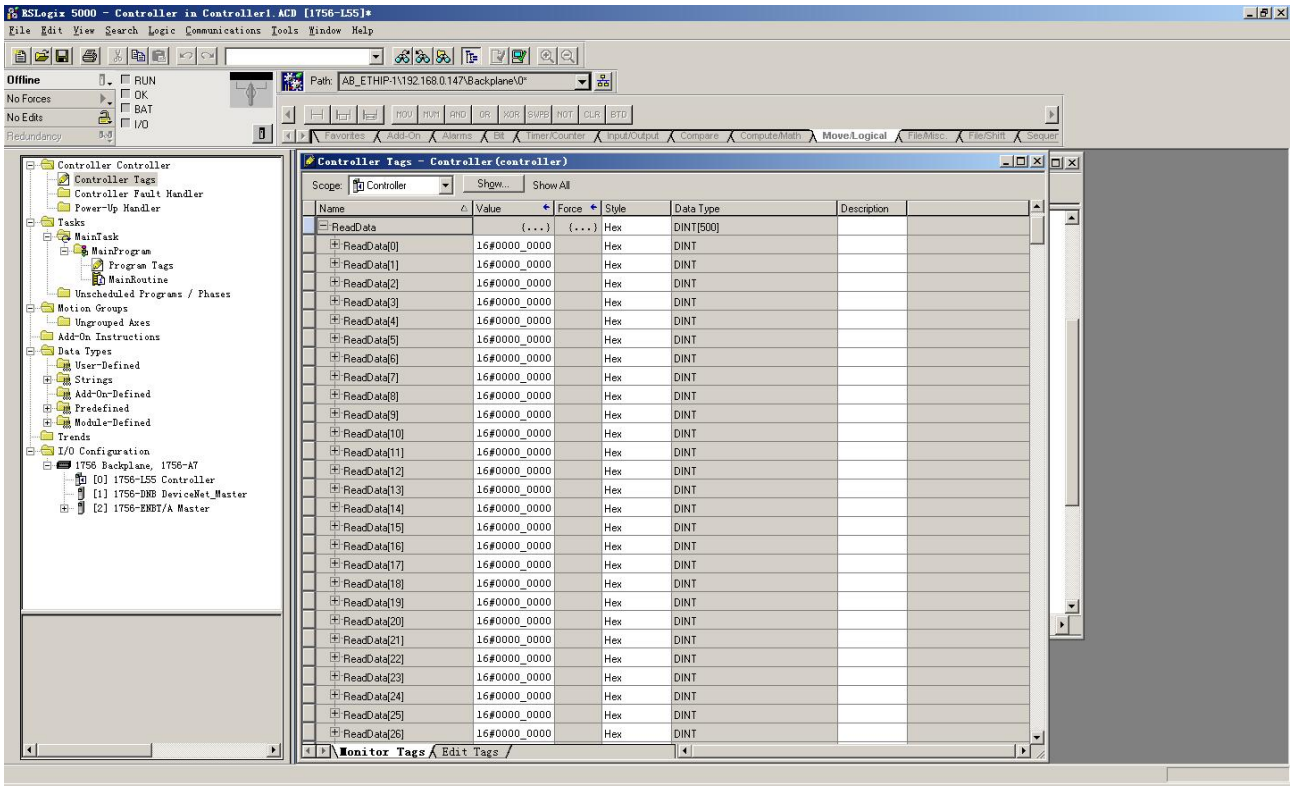
Open the "MainProgram" >> "ManRoutine" in the left tree view and add a "MSG" command. Select the "ReadTag" as the Message Control.

This command can send one simple read query, users need to configure more logic blocks to trigger the command.



Download the configuration to the PLC and set to online.

Double click on the "Controller Tags" to open the "Monitor Tags", as below:

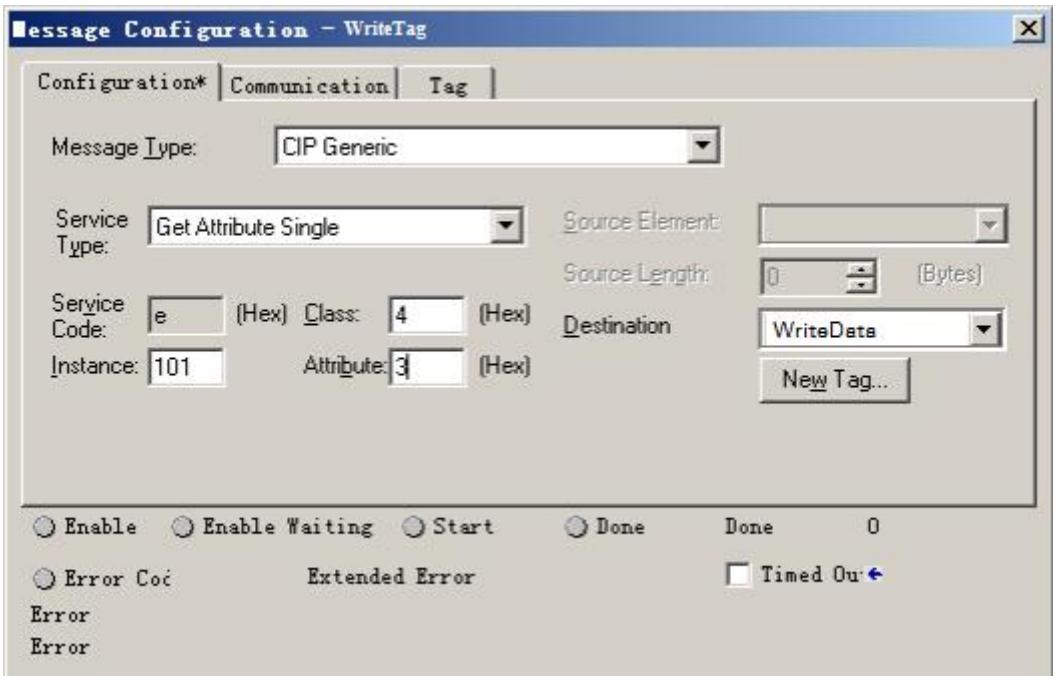


8.2.2 Write Data

Add two new tags in the “Controller Tags”. In this example, name the tags as “WriteTag” and “WriteData”. Set the WriteTag type as “MESSAGE” and the WriteData type as “DINT[500]”.

Right click on the WriteTag and select Configure “WriteTag”.

Configure the parameters in the pop-up window:



In “Configuration*” page:

Message Type: Select “CIP Generic”.

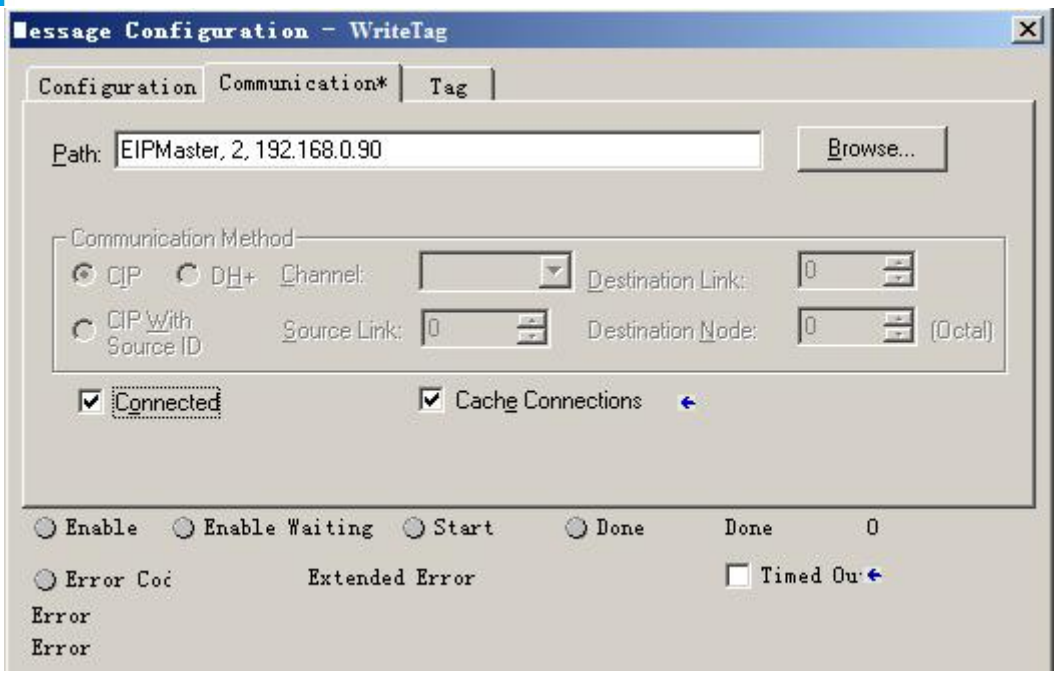
Service Type: Select “Get Attribute Single” and the Service Code will be set to “e (HEX)”.

Class: 4

Instance: 101 (128 bytes), 111(256 bytes) and 121 (492 bytes) are optional.

Attribute: 3 (Hex)

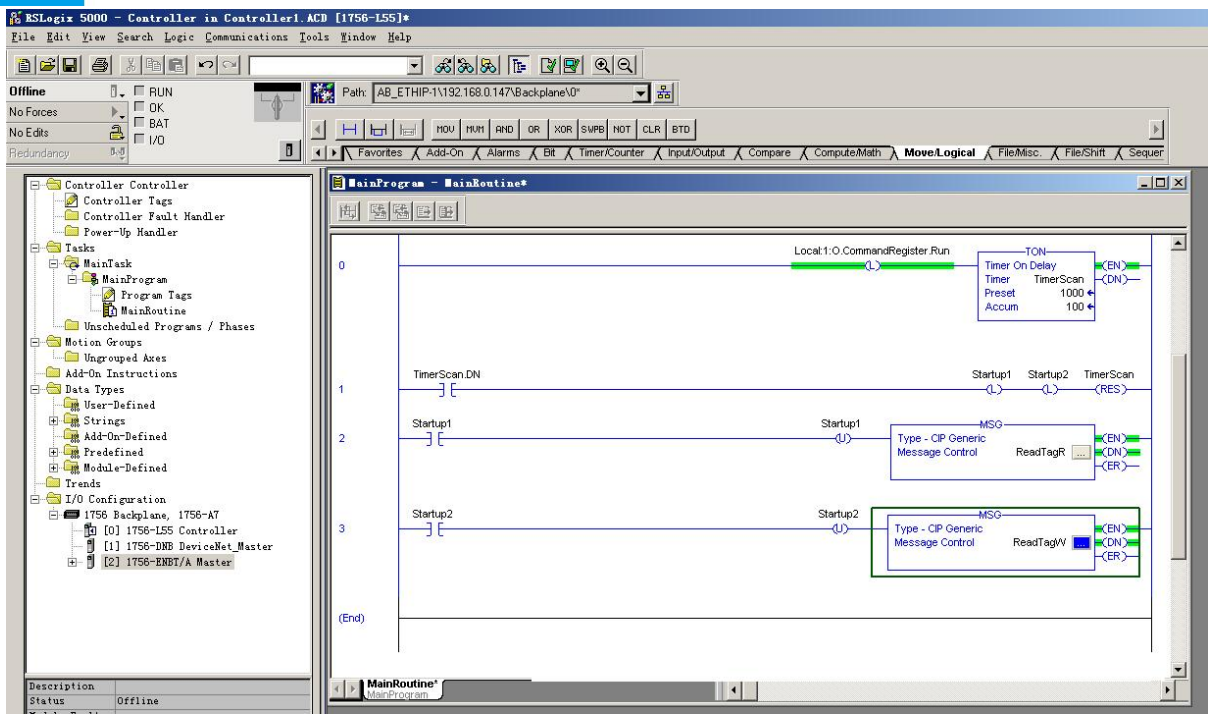
Destination: Select “WriteData” and all the write(output) data will be stored in the WriteData tag.



Jump to the “Communication” page. Input the EtherNet/IP slave path. The path format is “EtherNet/IP master name”, “slot number”, “EtherNet/IP slave IP address”. In this example, the EtherNet/IP master name is “EIPMaster”, the slot number is “2” and the EtherNet/IP slave (GT200-MQ-IE) IP address is 192.168.0.90.

Click “OK” or “Apply” to save the apply configuration.

Open the “MainProgram” >> “ManRoutine” in the left tree view and add a “MSG” command. Select the “WriteTag” as the Message Control.



Download the configuration to the PLC and set to online.

Double click on the “Controller Tags” to open the “Monitor Tags” to monitor data.

Appendix A: MQTT Message Format

A.1 Default Format

A.1.1 Publish Message

Note: "//" indicates the comments part.

Property type:

GPS: ***//Global Positioning System***

```
{
    "version":      "1.1.1.0",
    "GPS":{
    "Latitude":     31.057020,
    "Longitude":    121.189590
    }
}
```

DI: ***//Digital Input***

```
{
    "version":      "1.1.1.0",
    "DI":       {
    "DI1":      false,
    "DI2":      true
    }
}
```

Read Coil Status / Input Status:

```
{
    "version":      "1.1.1.0",          //Version information
    "time":     "19/03/07,17:04:07",    //Time stamp
    "properties":{
    //When publishing messages in property type, the GT200-MQ-IE can publish 8 properties at most in one message. If more than 8 properties need to be transmitted, the GT200-MQ-IE will publish multiple messages of 8 or less properties.
    "BI3":      false,
    "BI4":      true,
    "BI5":      false,
```

```
    "BI6":    false

    }
}
```

Read Holding Register / Input Register:

```
{
    "version":      "1.1.1.0",
    "time":   "19/03/07,17:10:03",
    "properties":{
    "AI2":    999999.500000,     //The data type (for example, Float) consists with the property configuration
    "AI3":    888888.500000,
    "AI4":    777777.500000,
    "AI5":    666666.500000
    }
}
```

Register / Data only type:

GPS:

```
{
    "version":      "1.1.1.0",
    "type":   "GPS",
    "Latitude":     31.057243,
    "Longitude":    121.188759
}
```

DI:

When two DI are enabled:

```
{
    "version":      "1.1.1.0",
    "type":   "DI1,DI2",     //The data type
    "npoint": 2,             //The number of data
    "data":   "00,ff"
}
```

When only one DI is enabled:

```
{
    "version":      "1.1.1.0",
    "type":   "DI1",
    "npoint": 1,
    "data":   "ff"
}
```

Modbus Master:

Read Coil Status/Read Input Status:

Example (Read Coil Status):

```
{
    "version":      "1.1.1.0",
    "type":    "MODBUS_MASTER",
    "slave id":      1,              //The slave address of the field device
    "fc": 1,                         //Modbus function code
    " addr ":  0,                    //Modbus coil/register starting address, starting form 0
    "npoint": 2,                     //The number in data
    "data":    "42,81",              //The data type consists with that in Modbus protocol
    "time":    "19/02/27,16:05:14"//Time stamp. If disable, the message will not contain this line.
}
```

Read Holding Register/Read Input Register:

Example (Read Holding Register):

```
{
    "version":      "1.1.1.0",
    "type":    "MODBUS_MASTER",
    "slave id":      1,
    "fc": 3,
    " addr ":  0,
    "npoint":32,
    "data":"11,11,00,00,33,33,00,00,55,55,00,00,77,77,00,00,99,99,00,00,11,11,00,00,13,13,00,00,15,15,00,00"
}
```

Modbus Slave / Modbus TCP Server:

Coil Status:

When only one coil' status changes:

```
{
    "version":      "1.1.1.0",
    "type":    "MODBUS_TCP_SLAVE",
    "block":  0,        //Modbus Block. For example, "0" indicates the Coil Status block.
    " addr ":  2,       //Modbus status/register starting address, starting form 0
    "npoint":    1,
    "data":    "FF",    //00 or FF indicates 0 or 1.
    "time":    "19/02/27,16:13:40"
}
```

When multiple coils' status change:

Example 1:

```
{
    "version":      "1.1.1.0",
    "type":    "MODBUS_TCP_SLAVE",
    "block":   1,
    " addr ":  3,          //Modbus coil/register starting address, starting form 0
    "npoint":  2,
    //In this example, the 4th and 5th bits of the 1st byte are the two coils' statue.
    "data":    "18",      //It's HEX data.
    "time":    "19/02/27,16:23:35"
}
```

Example 2:

```
{
    "version":      "1.1.1.0",
    "type":    "MODBUS_TCP_SLAVE",
    "block":   0,
    " addr ":  6,          //Modbus coil/register starting address, starting form 0
    "npoint":  4,
    //In this example, the 1st bit of the 1st byte, the 7th and 8th bits of the 2nd byte is the three coils' status.
    "data":    "c0,03",
    "time":    "19/02/27,16:35:01"
}
```

Holding Register:

Example:

```
{
    "version":      "1.1.1.0",
    "type":    "MODBUS_TCP_SLAVE",
    "block":   4,
    " addr ":  1,
    "npoint":  4,
    "data":    "11,22,33,44",
    "time":    "19/02/27,16:48:39"
}
```

A.1.2 Subscribe Message

Note: "//" indicates the comments part.

Property type: (Up to 8 properties can be changed at one time)

DO: **//Digital Output**

```
{
    "DO":{
    "DO1":  false,
    "DO2":  true
    }
}
```

Modbus Master/Slave / Modbus TCP Server

Change Coil/Input Status:

```
{
    "properties":  {
    "BO3":  false,
    "BO4":  true,
    "BO5":  false
    }
}
```

Change Holding/Input Register:

```
{
"properties":  {
    "AO3":  123.5,
    "AO4":  321.5,
    "AO5":  555.6
    }
}
```

Register / Data Only Type:

DO:

```
{
    "type":  "DO1",
    "npoint": 1,
    "data":  "00"
}
```

Modbus Master:

Force Single Coil:

```
{
```

```
    "type":    "MODBUS_MASTER",
    "slave id":      1,
    "fc": 5,                //Modbus function code
    " addr ":  0,
    "npoint":  1,           //00 or FF indicates 0 or 1.
    "data":    "FF"
 }
```

Force Multiple Coils:

```
{
    "type":    "MODBUS_MASTER",
    "slave id":      1,
    "fc": 15,               //Modbus function code
    " addr ":  0,
    "npoint":  2,
    //The data contains multiple coils' status.
    "data":    "ff,aa"
}
```

Preset Single Register:

```
{
    "type":    "MODBUS_MASTER",
    "slave id":      1,
    "fc": 6,
    " addr ":  0,
    "npoint":  2,
    "data":    "aa,bb"
 }
```

Preset Multiple Registers:

```
{
    "type":    "MODBUS_MASTER",
    "slave id":      1,
    "fc": 16,
    "addr":    0,
    "npoint":32,
    "data":"11,11,00,00,22,22,00,00,33,33,00,00,44,44,00,00,00,00,00,00,00,00,00,00,00,00,88,8
    8,00,00,99,99"
}
```

Modbus Slave / Modbus TCP Server

Change single input status bit:

```
{
```

```
    "type":    "MODBUS_SLAVE",
    "block":   1,          //Modbus Block. "1" indicates the Input Status block.
    "addr ":   0,
    "npoint":  1,
    "data":    "FF"
 }
```

Change multiple input status bits:

```
{
    "type":    "MODBUS_SLAVE",
    "block":   1,
    "addr":    0,
    "npoint":  8,
    //In this example, 8 bits are sent in a byte.
    "data":    "aa"
 }
```

Change Input Register:

```
{
    "type":    " MODBUS_SLAVE ",
    "block":   3,
    "addr":    1,
    "npoint":  12,
    "data":    "02,22,03,33,04,44,05,55,06,66,07,77"
}
```

A.2 Custom Message Format

```
{
    "timestamp": |TIMESTAMP|,
    "values":    {
        |#each VALUES|
        "name": "|PROPERTIESNAME|", "v": |VALUE||#unless @last|,|/unless|
        |/each|
    }
}
```

(1) Variable description

|PROPERTIESNAME|: Property Name. For example, AI1.

|VALUE|: The value of the property.

|TIMESTAMP|: The time when the property was read.

(2) Grammar

|#each VALUES|
    |PROPERTIESNAME|, |VALUE|, |TIMESTAMP|,
|/each|

The "each" is to generate text for each property. Users just need to edit one single property template between "|#each VALUES|" and "|/each|" and it will be copied to every property. Up to 8 properties can be published in one message.

Template example:

```
{
    |#each VALUES|
        "|PROPERTIESNAME|_value": |VALUE|,
        "|PROPERTIESNAME|_timestamp": |TIMESTAMP|,
    |/each|
}
```

Message example:

```
{
    "AI1_value": 23,
    "AI1_timestamp": 1456150184825,
    "AI2_value": 17,
    "AI2_timestamp": 1456150184984,
    …
}
```

(3) Additional grammar description:

|#unless @last|,|/unless|

This line means "Must insert a comma at the end of each line, unless it's the last line". It cancels the comma in the last line. If users regard the comma in the last line as a syntax error, the template must contain this line.

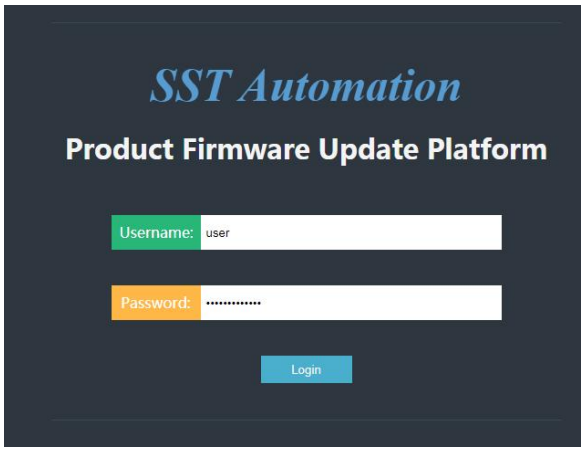
Appendix B: Change Product Firmware

SST Automation provides the built-in product firmware update platform to update the product firmware and change the running mode. Users can change the GT200-MQ-IE firmware by the following steps:

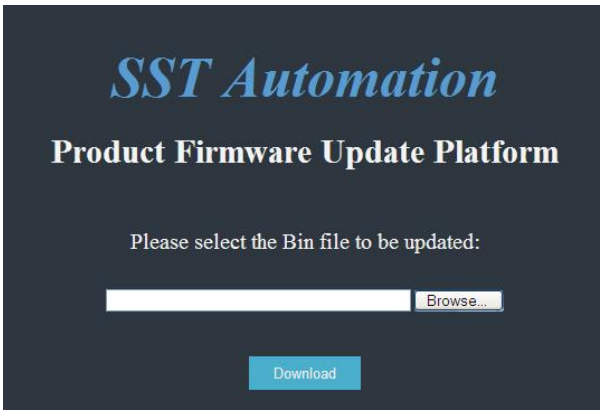
1. Change the IP address of PC as fixed 192.168.0.xxx.
2. Hold the button on the GT200-MQ-IE and power it on. The DI/DO LED will blink red and green, that indicates the GT200-MQ-IE is ready to update the firmware. Then release the button.
3. Enter 192.168.0.10 in your web browser to enter SST Automation Product Firmware Update Platform. Login with the following information:

   Username: user

   Password: sstautomation



4. Click “Browse...” to select a Bin file. The Bin files are located in the installation path of the SST-MQT-CFG software.



5. Click “Download” to download the specified firmware into the GT200-MQ-IE.

6. After download, it shows the firmware update is successful. Click “Reboot” to restart the GT200-MQ-IE.

7. Wait for the GT200-MQ-IE to restart.