PUSR®
www.pusr.com

USR-M100 Quick Start Guide with Azure

Be Honest & Do Best

Your Trustworthy Smart Industrial IoT Partner

1. Introduction ..... - 3 -
2. Azure ..... - 3 -
2.1. Basic Parameters ..... - 3 -
2.2. Publish Settings ..... - 4 -
2.3. Subscribe Settings ..... - 4 -
3. Azure Connection Test ..... - 4 -
3.1. Preparations ..... - 4 -
3.2. Azure Configuration ..... - 4 -
3.2.1. Login ..... - 4 -
3.2.2. Create IoT Hub ..... - 5 -
3.2.3. Register Device ..... - 8 -
3.3. Device Configuration ..... - 11 -
3.4. Data Transmission Test ..... - 12 -
3.4.1. Transparent Transmission ..... - 12 -
3.4.2. Edge Computing ..... - 15 -

1. Introduction

USR-M100 is a high-performance and scalable edge IOT gateway. This device integrates edge collection, data calculation, IO collection and control, linkage control, data transparent transmission, data encryption and other functions. M100 uses the Cortex-M7 core, with a main frequency of up to 400Mhz, supports both Ethernet ports and LTECat1 networks, making network communication methods more diversified. It supports TCP/UDP/MQTT(S)/HTTPS(S) protocol, supports Modbus RTU/TCP conversion and Modbus/Json protocol data reporting, also supports connecting to PUSR Cloud, Alibaba Cloud and AWS Cloud.

USR-M100 adopts an expandable design in structure, and can be combined by expanding modules with different functions to better meet the needs of different scenarios for the number of IOs and communication interfaces, making it convenient, fast and cost-saving.

2. Azure

USR-M100 supports connecting to Azure platform via MQTT, which can be achieved via simple parameter configuration. At the same time, the device's edge computing and Azure functions support combined configuration, which is very flexible and practical.

2.1. Basic Parameters

- Client ID: MQTT client identifier.
- Server Address: Azure MQTT connection domain name
- Cipher: Azure MQTT connection authentication password.
- Server Port NO.: Azure MQTT connection port, default to 8883, cannot be changed.
- KeepAlive: Heartbeat data sending interval after establishing MQTT connection.
- Reconnecting time without data: If MQTT PUBLISH data (application data) is not delivered to M100 within the set time, the device will reconnect to Azure.
- Reconnection Interval: The time interval between a failed connection and the next attempt to connect.
- Clear session: MQTT protocol connection flag, used to control the lifetime of session.

2.2. Publish Settings

●Topic: Publish topic name.

●QOS: Message quality of published topics.

●Retained message: MQTT retains the message flag, which is used by the server to store this application message and its quality of service (QoS).

2.3. Subscribe Settings

●Topic: Subscribe topics.

●QOS: Message quality of subscribed topics.

3. Azure Connection Test

In this case, we will show how to connect M100 to Azure.

3.1. Preparations

●USR-M100*1

●RS485 serial to USB cable*1

●Ethernet cable*1

●12V/1A power adaptor*1

●DeviceExplorer tool (Software tools required for Azure platform testing, the installation package name is: SetupDeviceExplorer.msi)

3.2. Azure Configuration

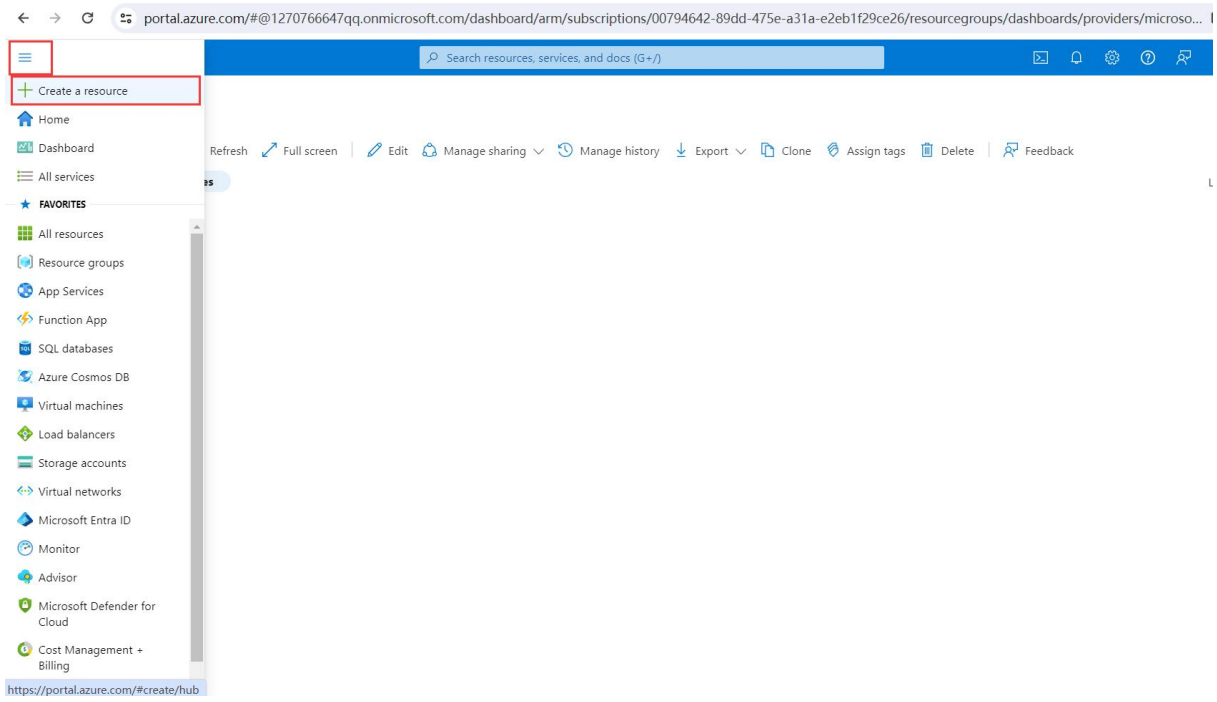
3.2.1. Login

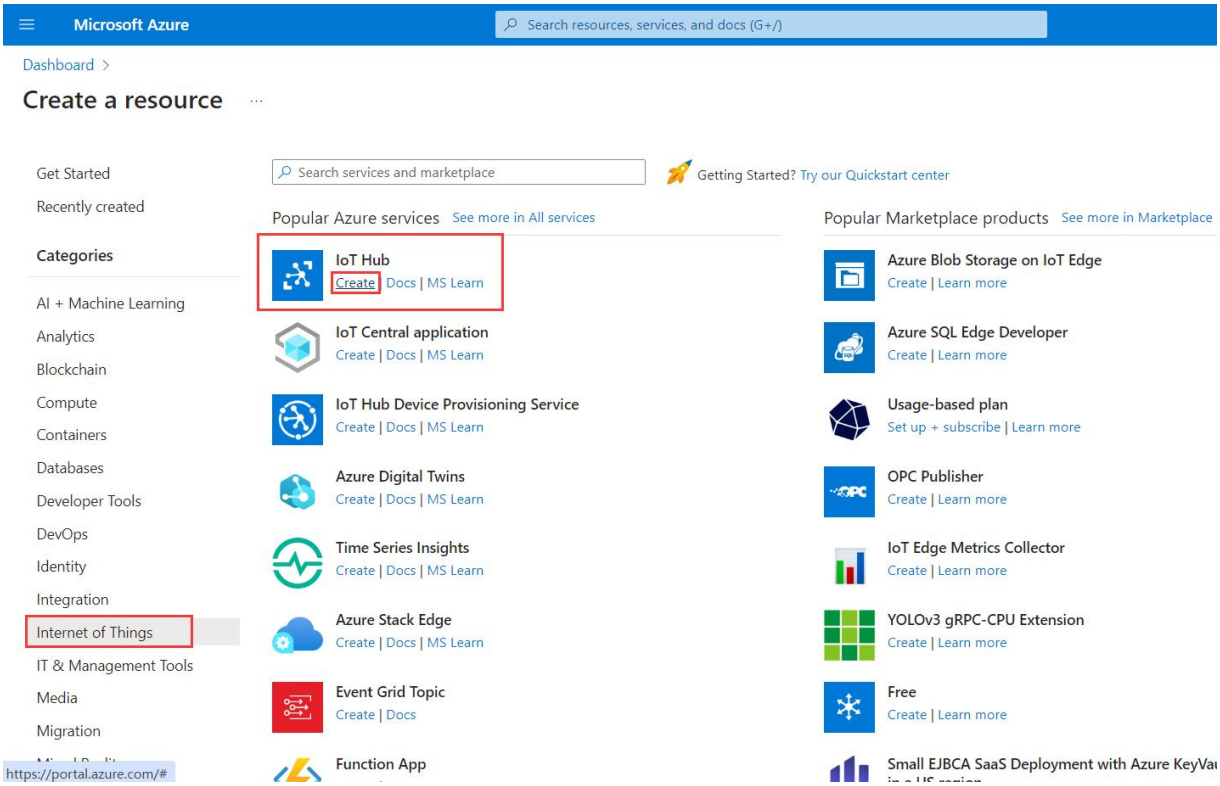
1.Log in to Azure platform: **https://azure.microsoft.com/en-us/products/iot-hub/**

2.Sign in (If you do not have an account, please register one first)

3.2.2. Create IoT Hub

1.After login, create an IoT hub. Find “Create a resource” >” Internet of Things” >” IoT Hub” , click “Create” .





2. In IoT hub interface, fill in below information.

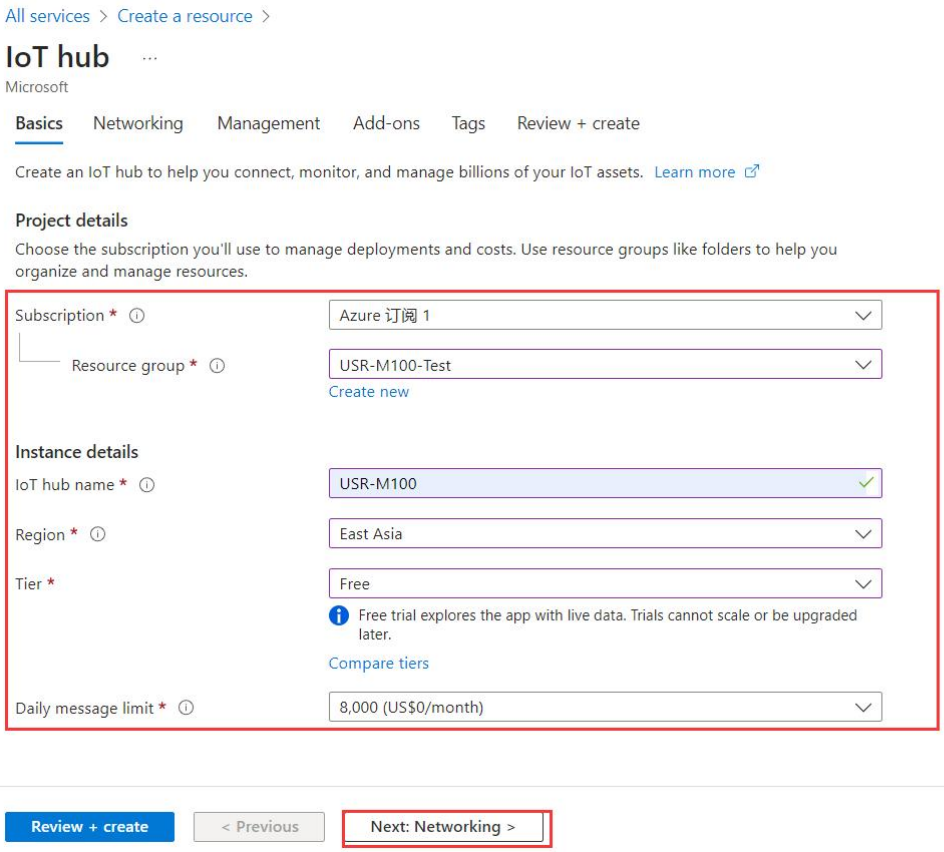
Subscription: Select the subscription that needs to be used to create this IoT Hub.

Resource Group: Create a resource group to host the IoT hub, or use an existing resource group. Fill in this field with an appropriate name.

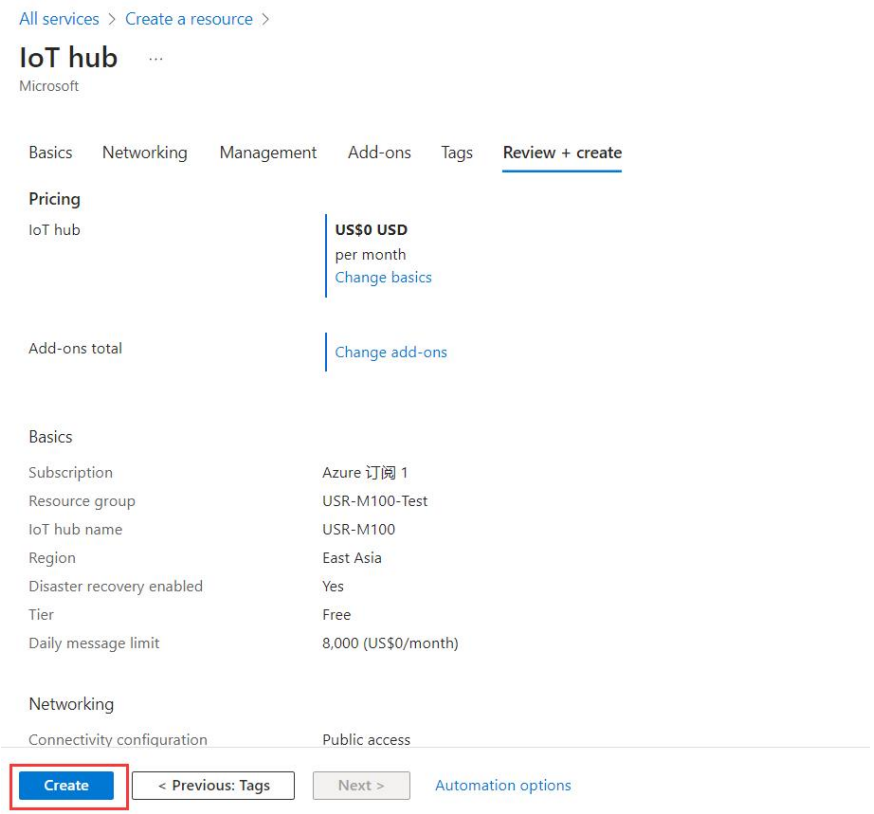
IoT Hub Name: Custom IOT hub name, the name is unique. If the entered name is available, there will be a green prompt.

Region: Select the appropriate regional ID according to your region.

Tier: Fee packages, choose according to your needs.

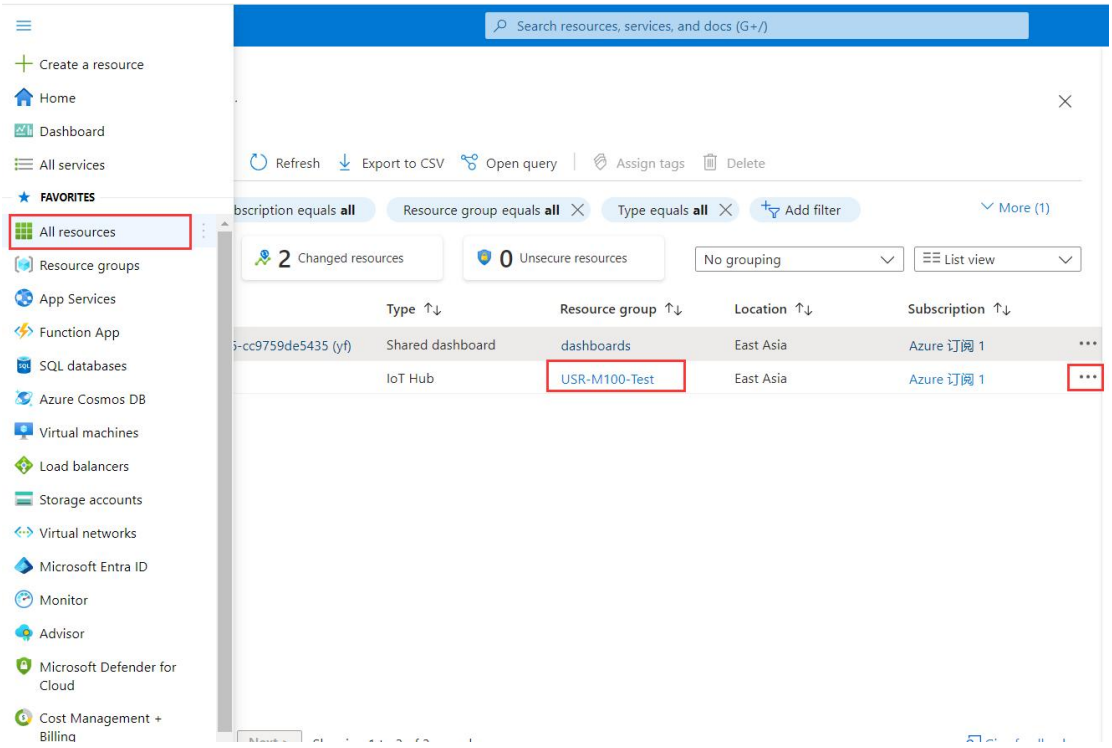


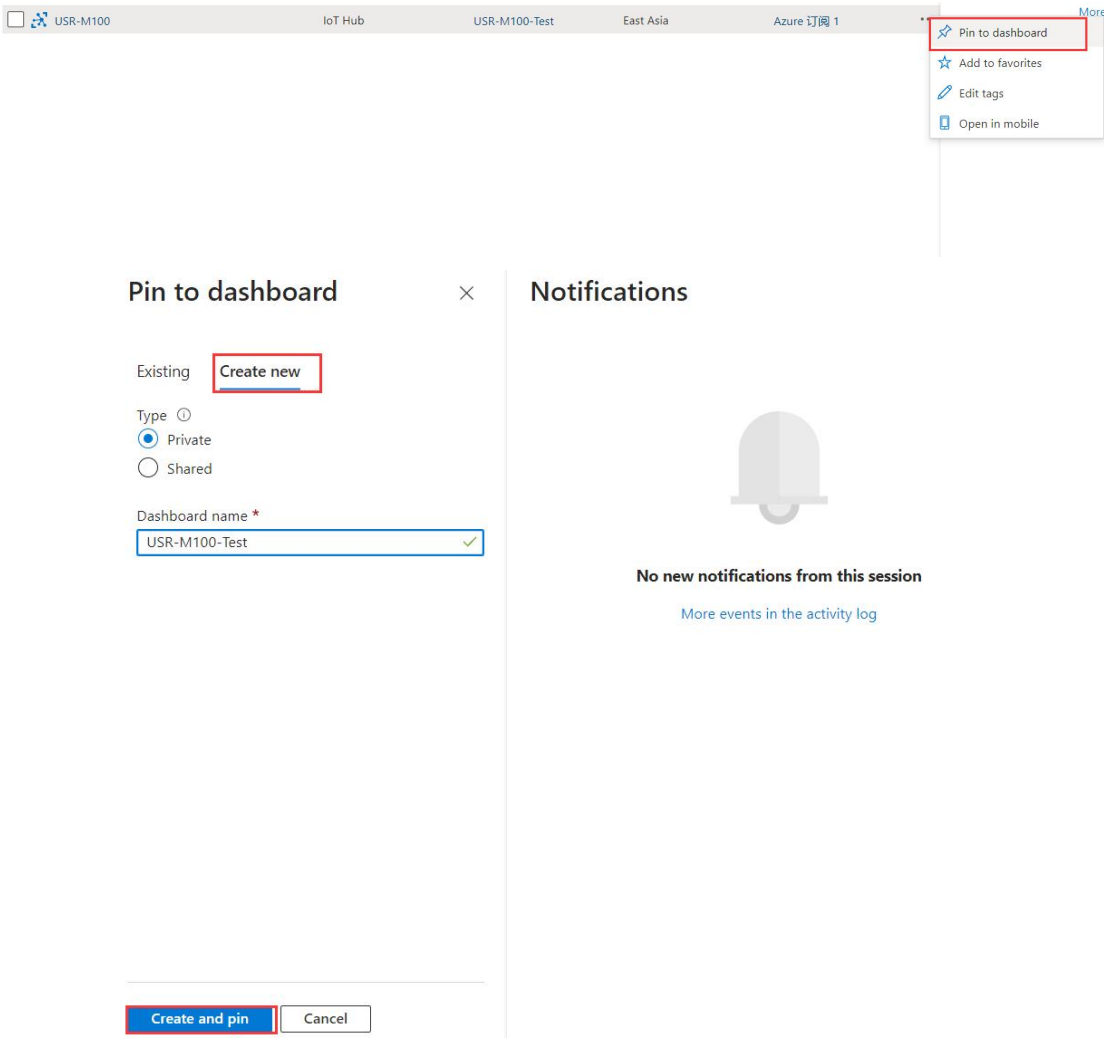
3.Keep clicking “Next” until you go to “Review + Create”, confirm the creation information then click “Create”.



4.You can check the creation progress in the "Notifications" window. After the creation is completed, register the device.

5.To facilitate subsequent testing, the created resources can be added to the dashboard manually.



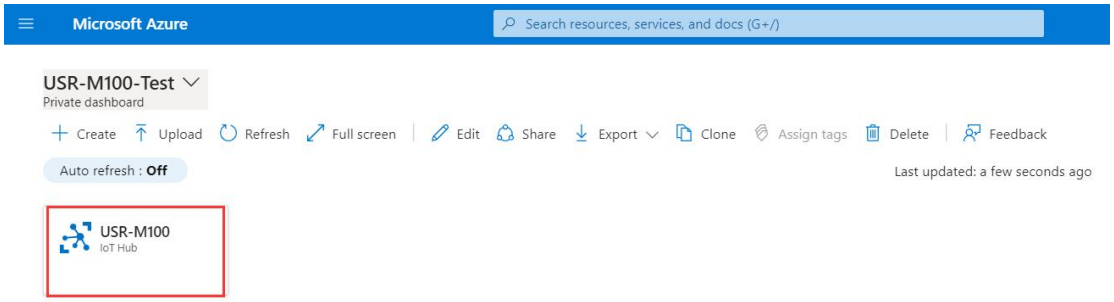


3.2.3. Register Device

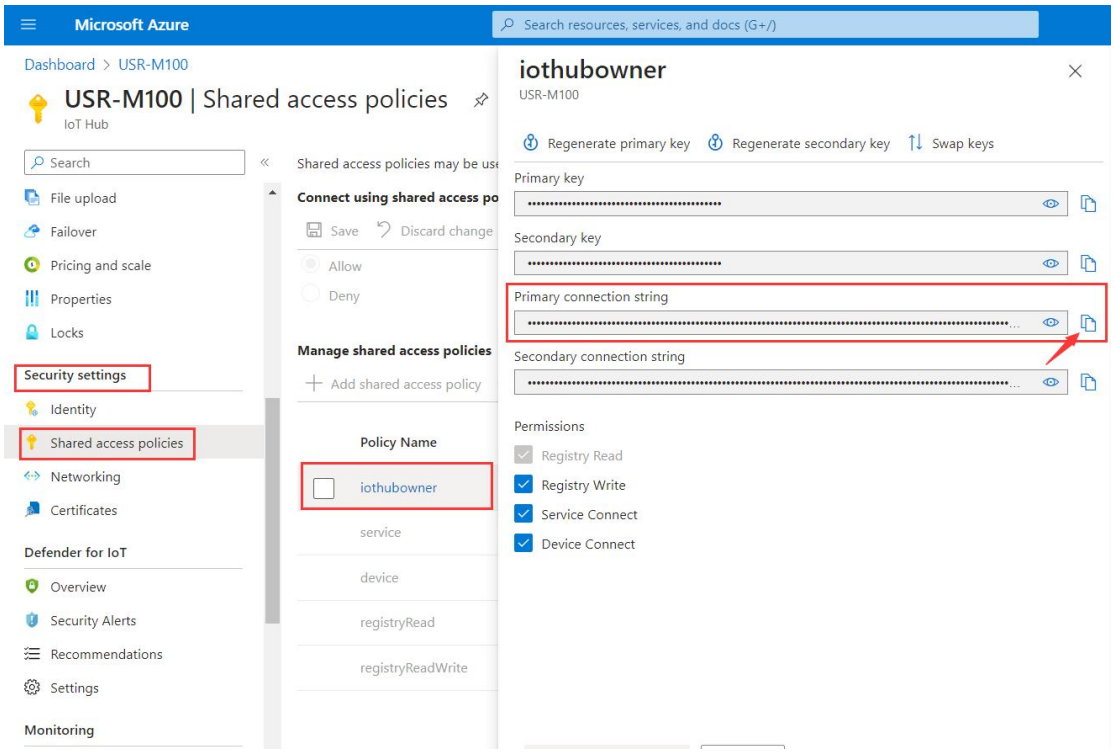
To run device-related examples, you need to register the device information into the IoT hub first, and then the device can connect to the IoT hub. In this example, the DeviceExplorer tool can be used to register the device.

1.Get the shared access policies in IoT hub.

a.Find the created IoT hub in the dashboard. Click it.

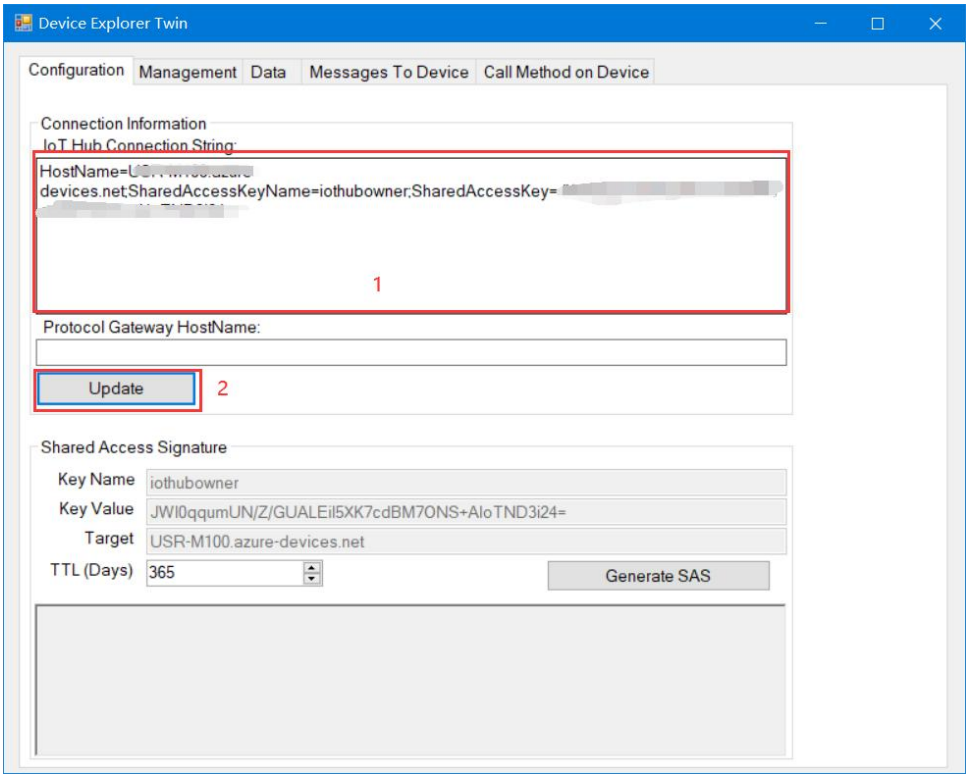


b.In “Security Settings”, choose “Shared access policies”, click “iothubowner”, find the “Primary connection string”, click the “Copy” button to get the key.

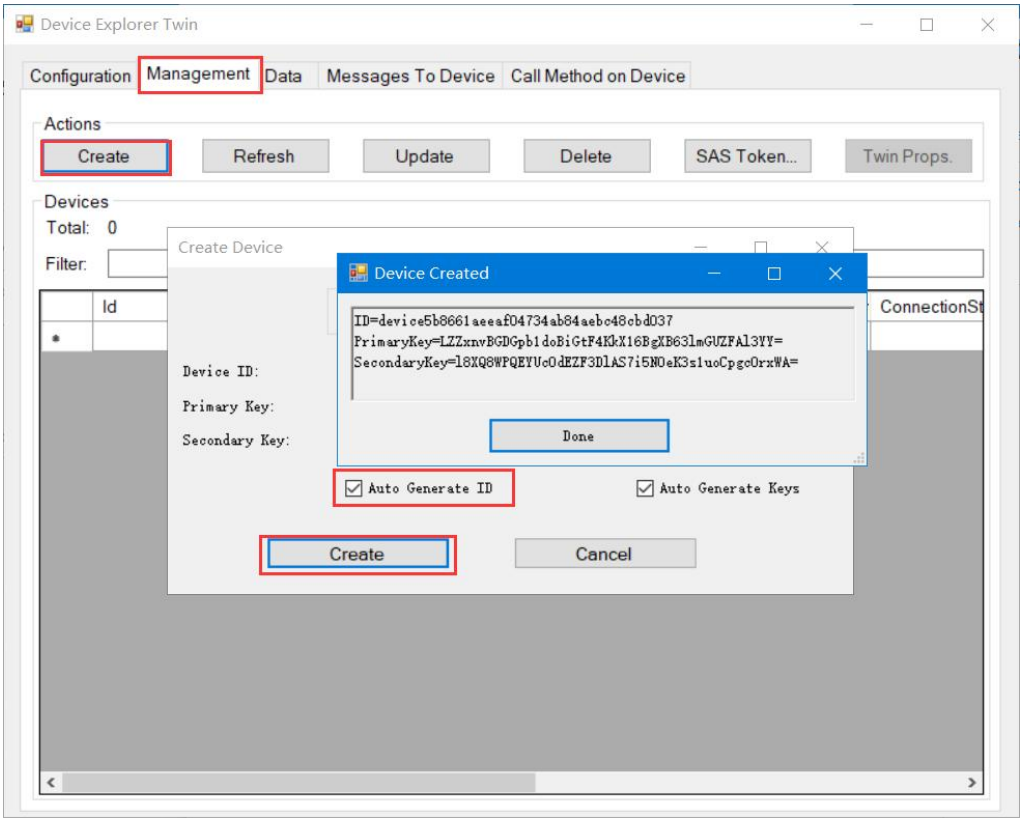


2.Open DeviceExplorer tool to create the device.

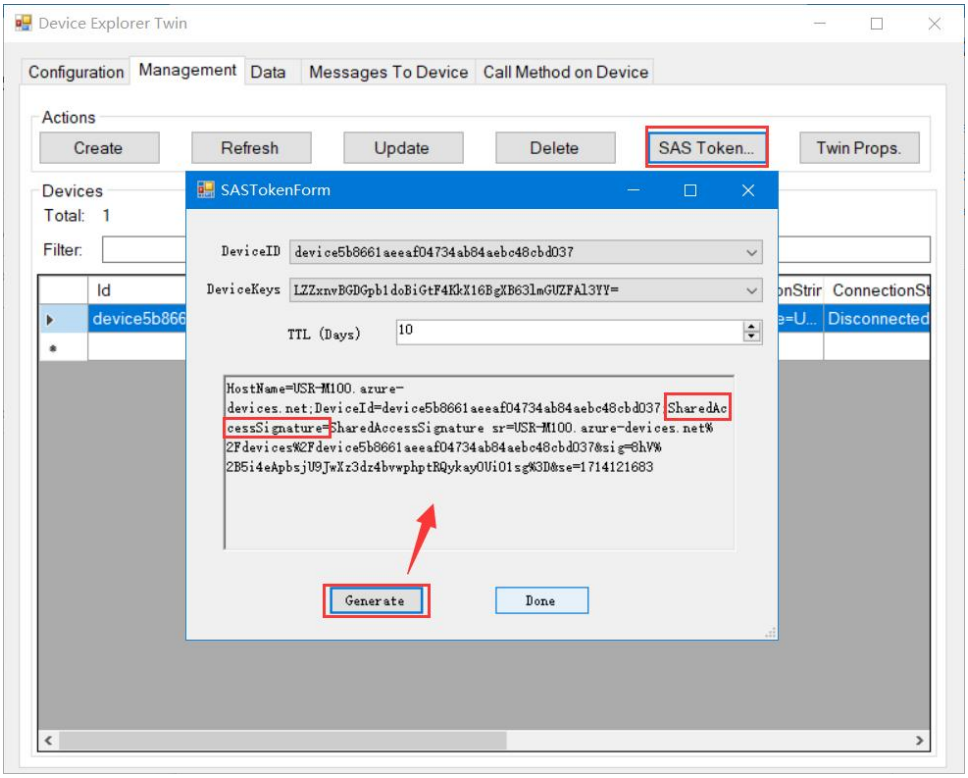
●Open DeviceExplorer, paste the copied key into the tool's “Configuration--IoT Hub Connection String” interface. Then click “Update” to update the configuration of the locally connected IoT hub.



- In “Management”, click “Create” to create the device.



- Choose “SAS Token...” button, In the pop-up dialog box, fill in the validity period in "TTL (Days)". Here we take 10 as an example. Click the "Generate" button to generate the connection password and find the "SharedAccessSignature=" field, copy and save the string information after the equal sign. To simplify our next configuration, please do not close this interface first.



3.3. Device Configuration

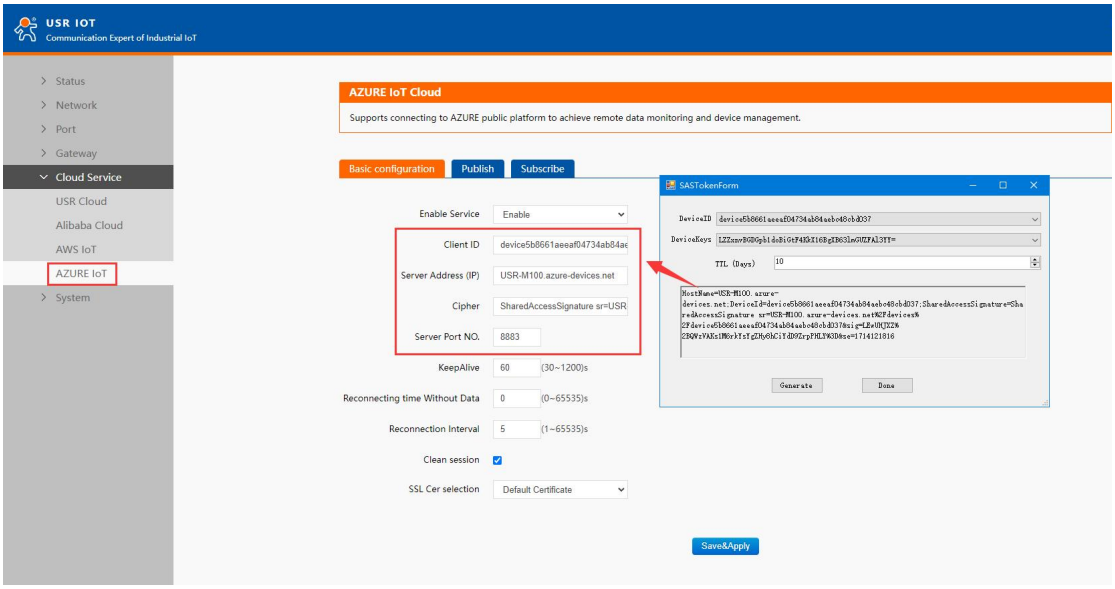
In the webpage of M100, configure parameters in “Cloud Service--AZURE IoT” interface.

Client ID: Fill in the Deviceid created in DeviceExplorer.

Server Address: Fill in the HostName shows in DeviceExplorer. You can also find it in “Configuration--Target” of DeviceExplorer.

Cipher: fill in the copied string information after the "SharedAccessSignature=" field in DeviceExplorer.

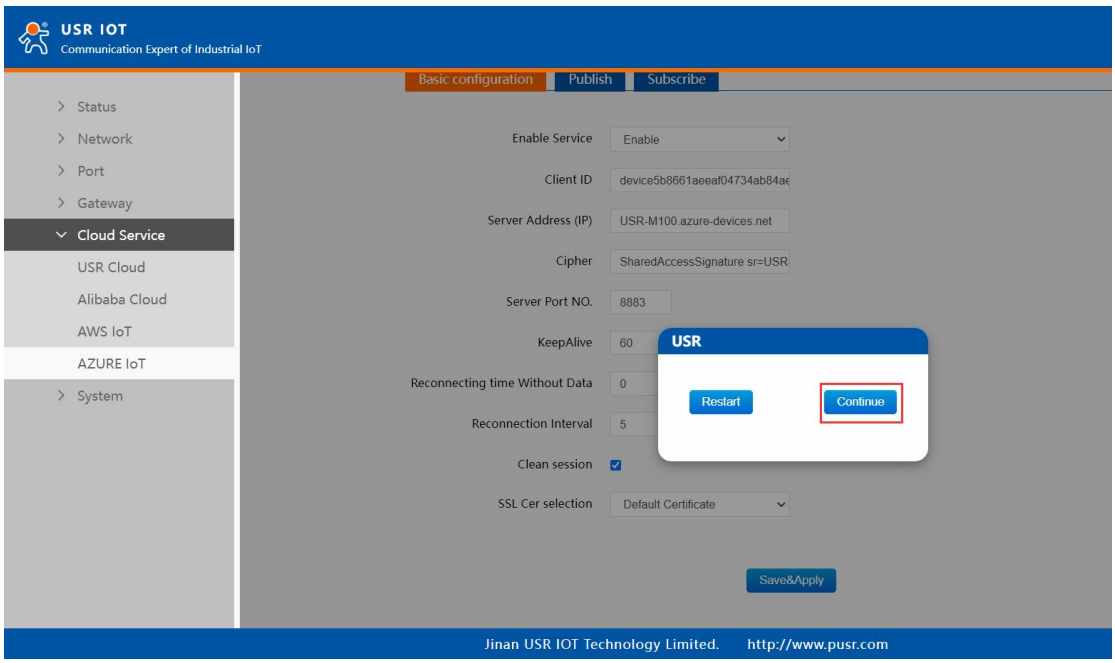
Server Port NO.: Default to 8883, cannot be changed.



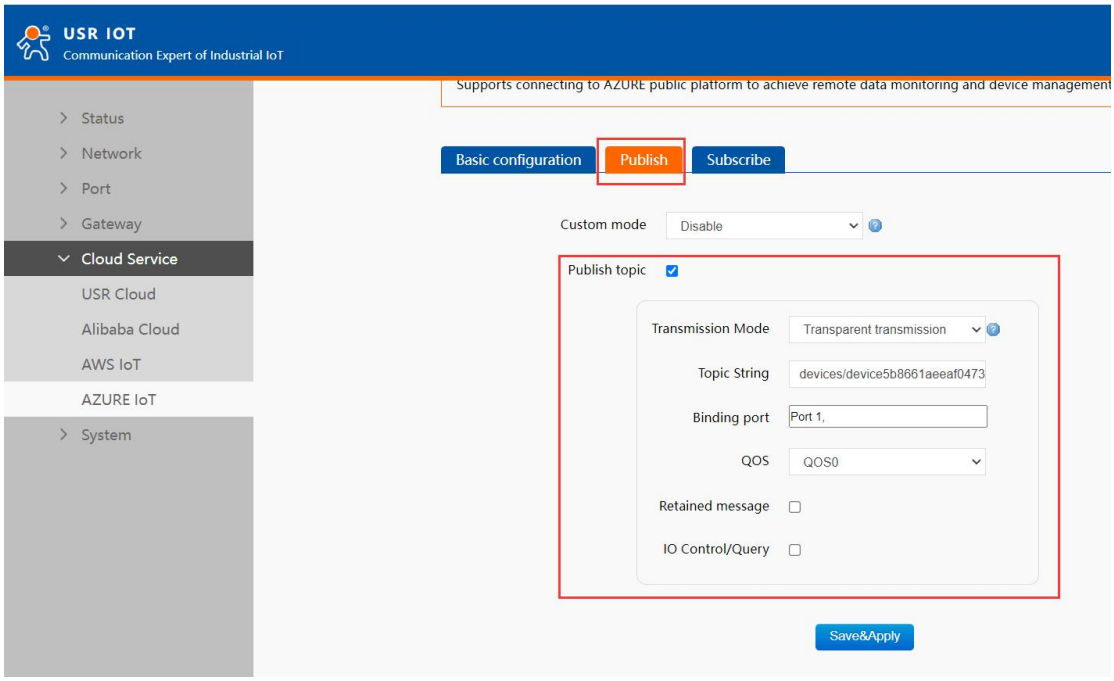
3.4. Data Transmission Test

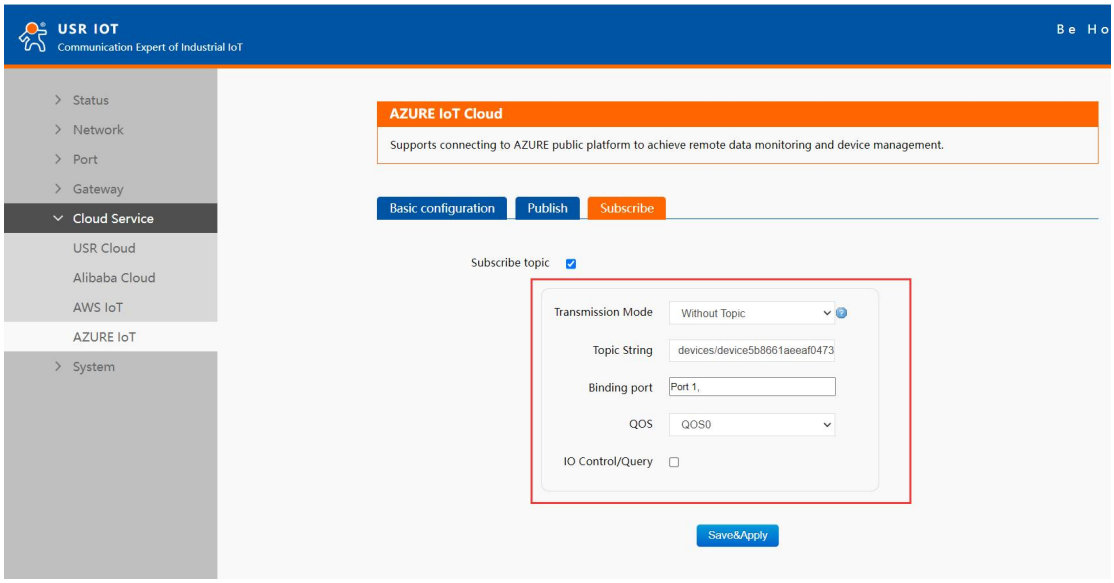
3.4.1. Transparent Transmission

1.After configuring the MQTT server parameters, click “Continue” to configure the topics.

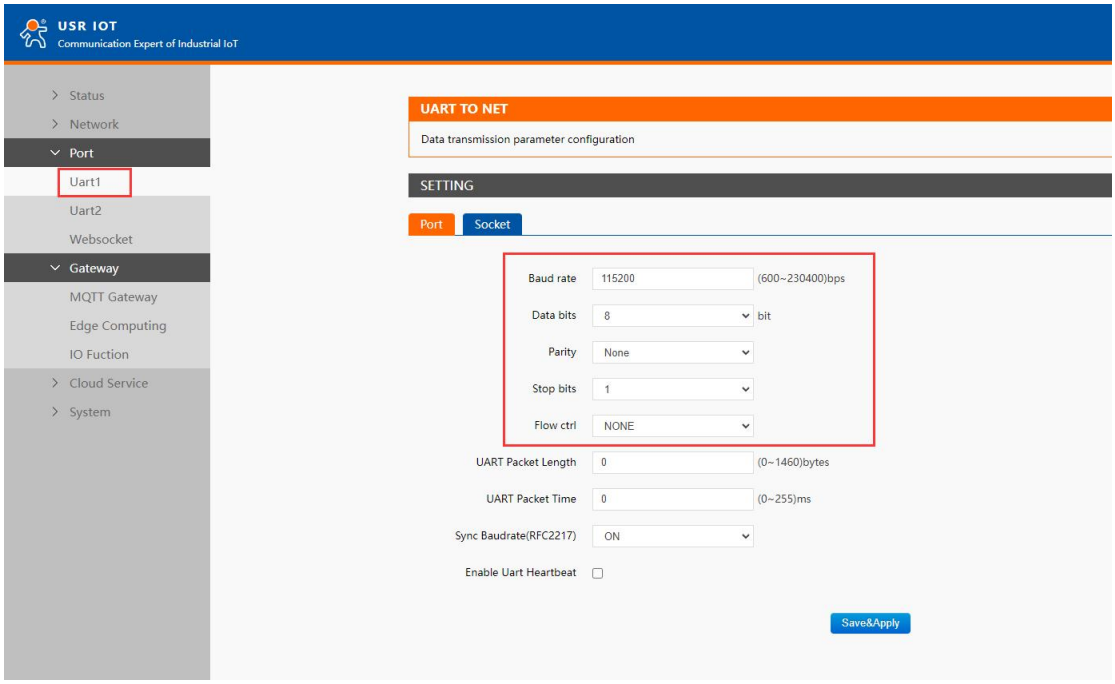


2.Enable “Publish topic” in “Publish”,the publish topic will be filled automatically by M100. Same with Subscribe topics.

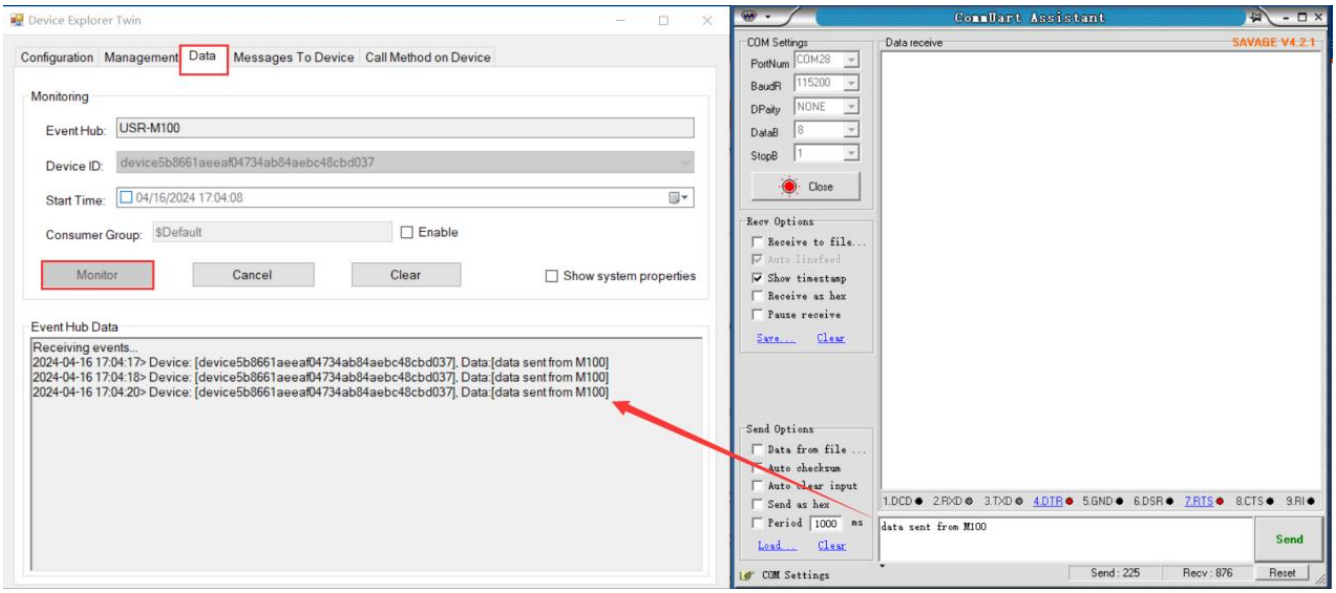




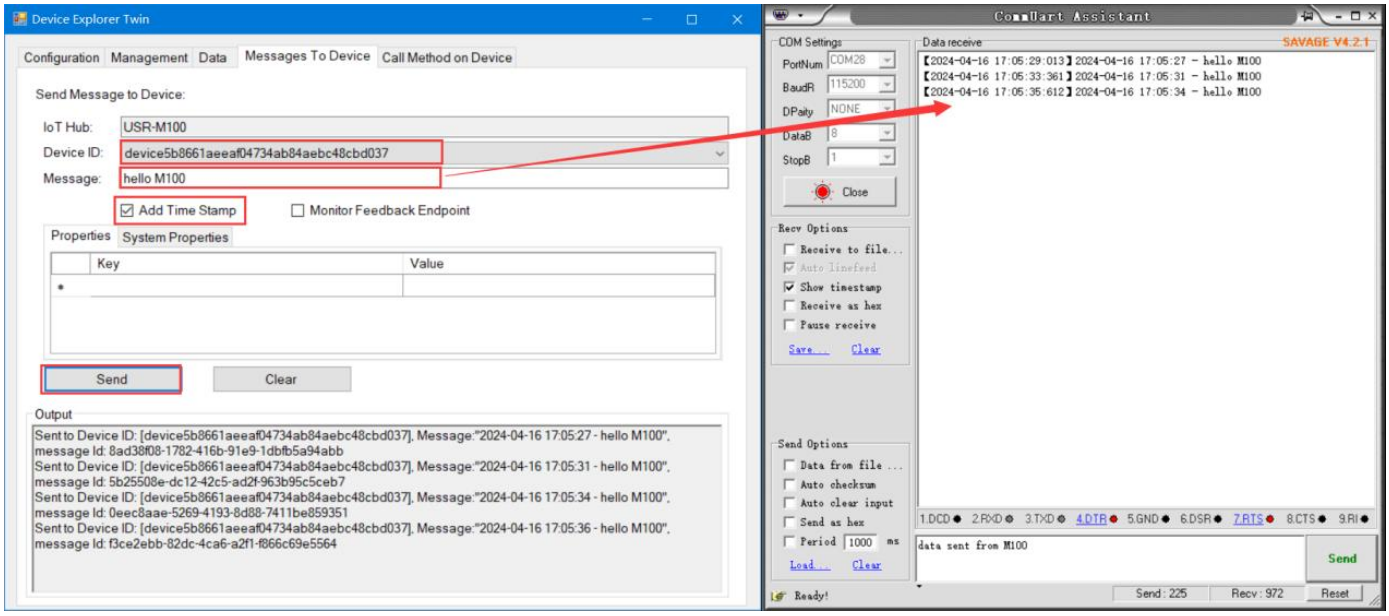
3.If you have connected serial RS232/RS485 devices to M100 device, also need to configure the serial parameters of M100 to be consistent with your device.



4. Here we connect the RS485 port of M100 to the PC via RS485 to USB cable. After the MQTT connection is established, we can achieve the data transmission between Azure and M100 serial port. Currently, Azure requires DeviceExplorer tool to achieve data receiving and sending debugging.
5. Receive data: Open the "Data' interface of the tool, choose the correct device ID, click "Monitor' to receive the data from M100.



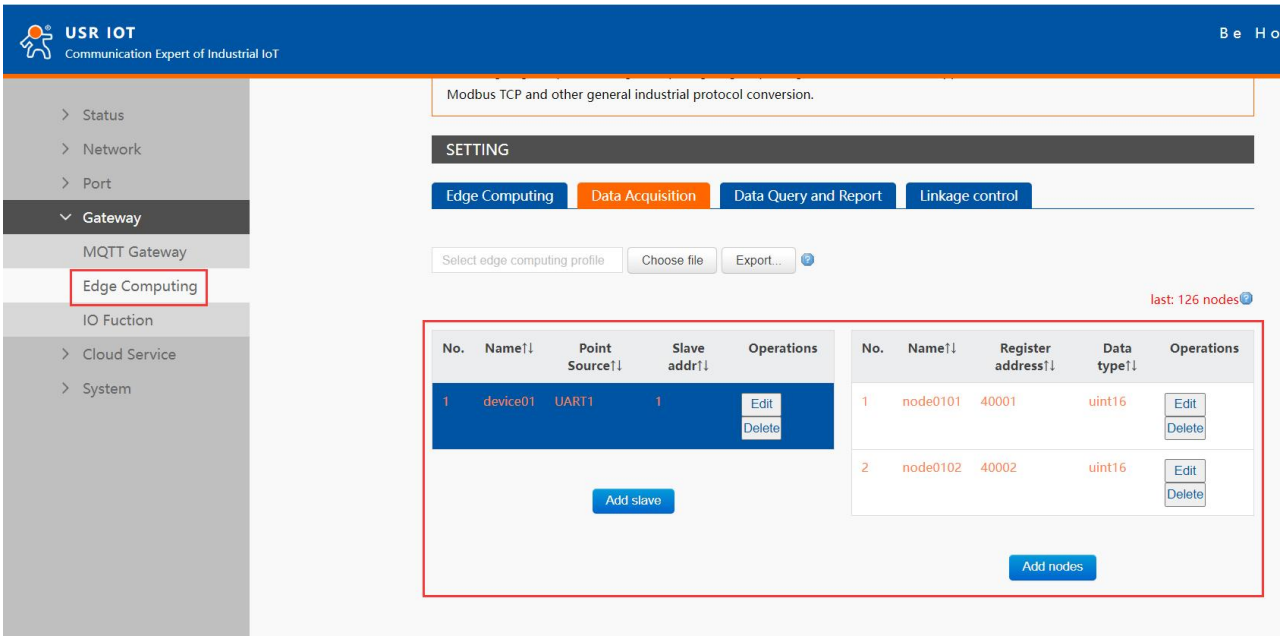
6. Send data: In "Messages To Devices" interface, choose the correct device ID, fill in the message needs to send, click "Send". We can receive the data on the serial port of M100.



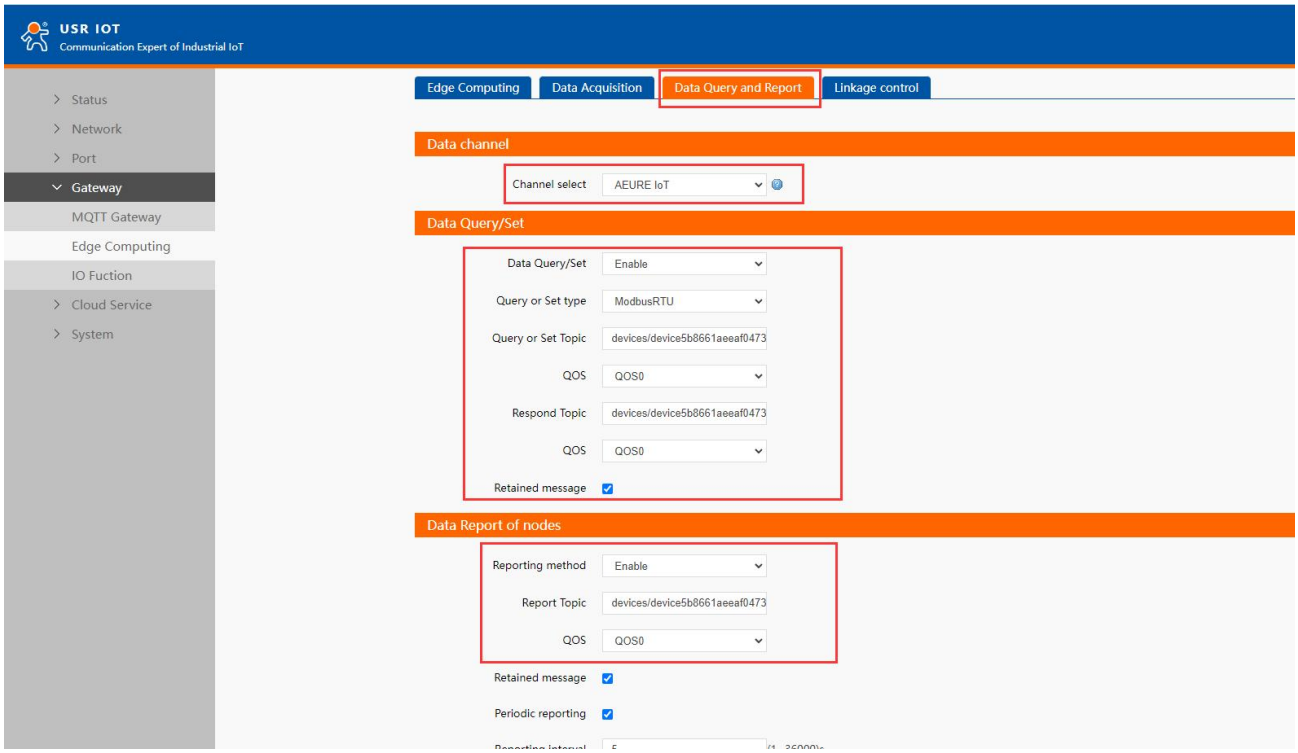
3.4.2. Edge Computing

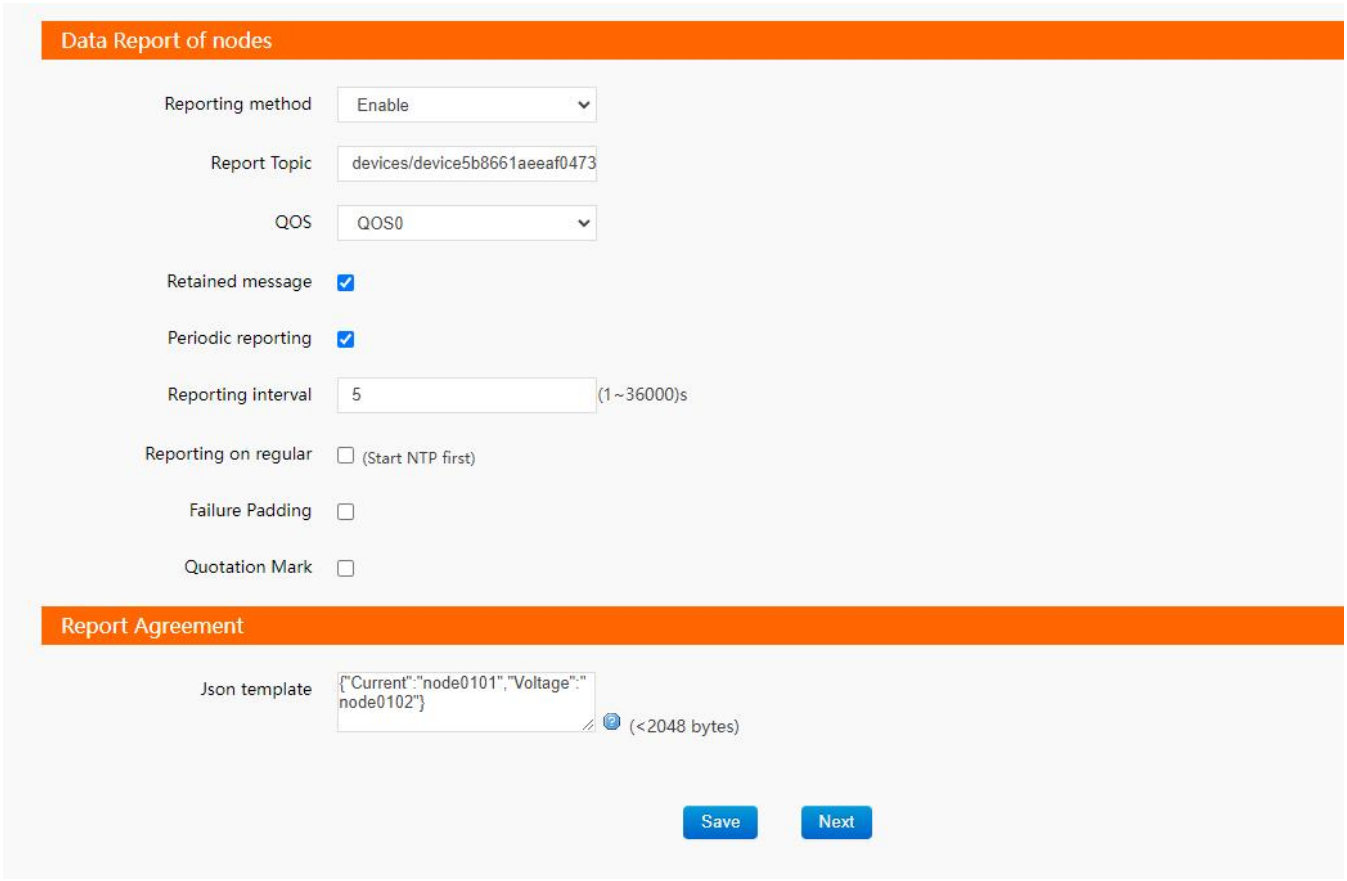
With edge computing function, M100 can read the serial Modbus RTU data and IO status automatically, and convert the response Modbus RTU data to JSON format, then send to the MQTT broker.

1.Enable "Edge Computing" function, configure the Modbus data points according to your serial device. Here you can also add the IO template to read the IO status of M100.

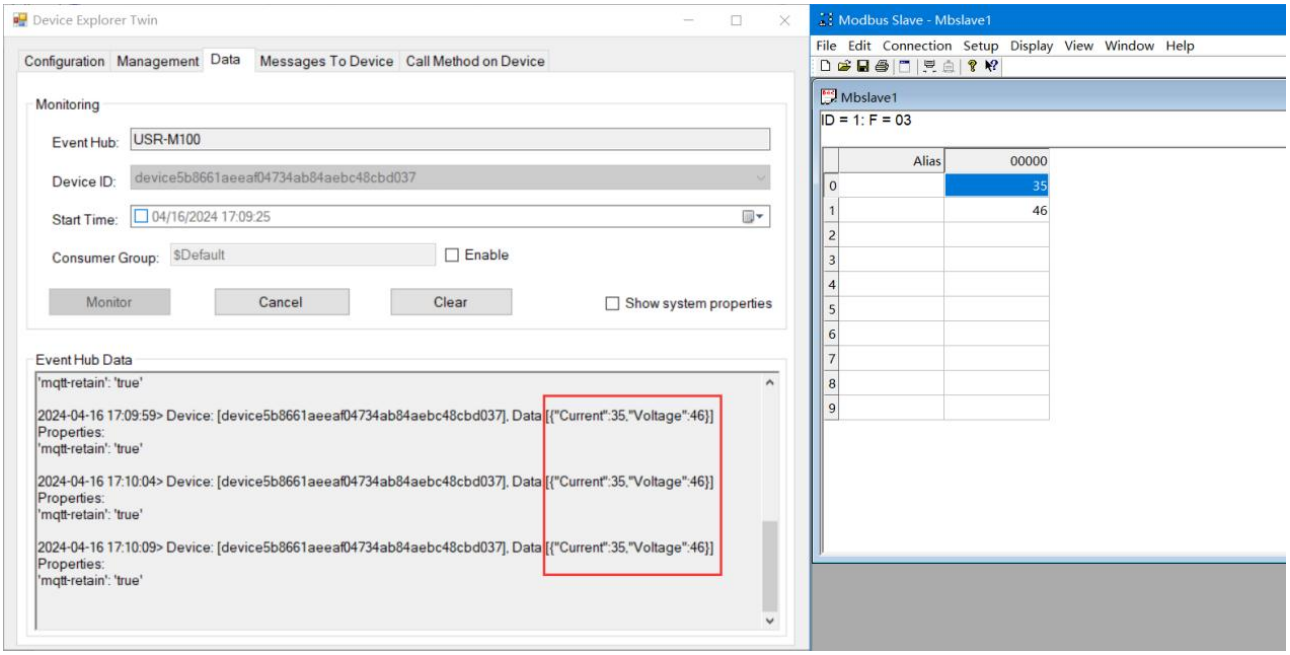


2. Change the data channel to AZURE IoT, enable data query/set function, topics will be filled automatically by M100.





3.Here we simulate a serial Modbus RTU device using Modbus Slave software. When we click "Monitor" in DeviceExplorer tool, we can receive the JSON format data from M100 device.


Device Explorer Twin
Configuration
Management
Data
Messages To Device
Call Method on Device
Monitoring
Event Hub:
USR-M100
Device ID:
device5b8661aeeaf04734ab84aebc48cbd037
Start Time:
04/16/2024 17:09:25
Consumer Group:
$Default
Enable
Monitor
Cancel
Clear
Show system properties
Event Hub Data
'mqtt-retain': 'true'
2024-04-16 17:09:59> Device: [device5b8661aeeaf04734ab84aebc48cbd037], Data:[{"Current":35,"Voltage":46}]
Properties:
'mqtt-retain': 'true'
2024-04-16 17:10:04> Device: [device5b8661aeeaf04734ab84aebc48cbd037], Data:[{"Current":35,"Voltage":46}]
Properties:
'mqtt-retain': 'true'
2024-04-16 17:10:09> Device: [device5b8661aeeaf04734ab84aebc48cbd037], Data:[{"Current":35,"Voltage":46}]
Properties:
'mqtt-retain': 'true'
Modbus Slave - Mbslave1
File Edit Connection Setup Display View Window Help
Mbslave1
ID = 1: F = 03
Alias
00000
0
35
1
46
2
3
4
5
6
7
8
9

Your Trustworthy Smart IOT Partner

PUSR®
www.pusr.com

Official Website: www.pusr.com

Official Shop: shop.usriot.com

Technical Support: h.usriot.com

Inquiry Email: inquiry@usriot.com

Skype & WhatsApp: +86 13405313834

Click to view more: Product Catalog & Facebook & Youtube