



USR IOT

Communication Expert of
Industrial IOT

Quick Start Guide with AWS IoT



Build a Smarter IOT world, Your Trustworthy Partner

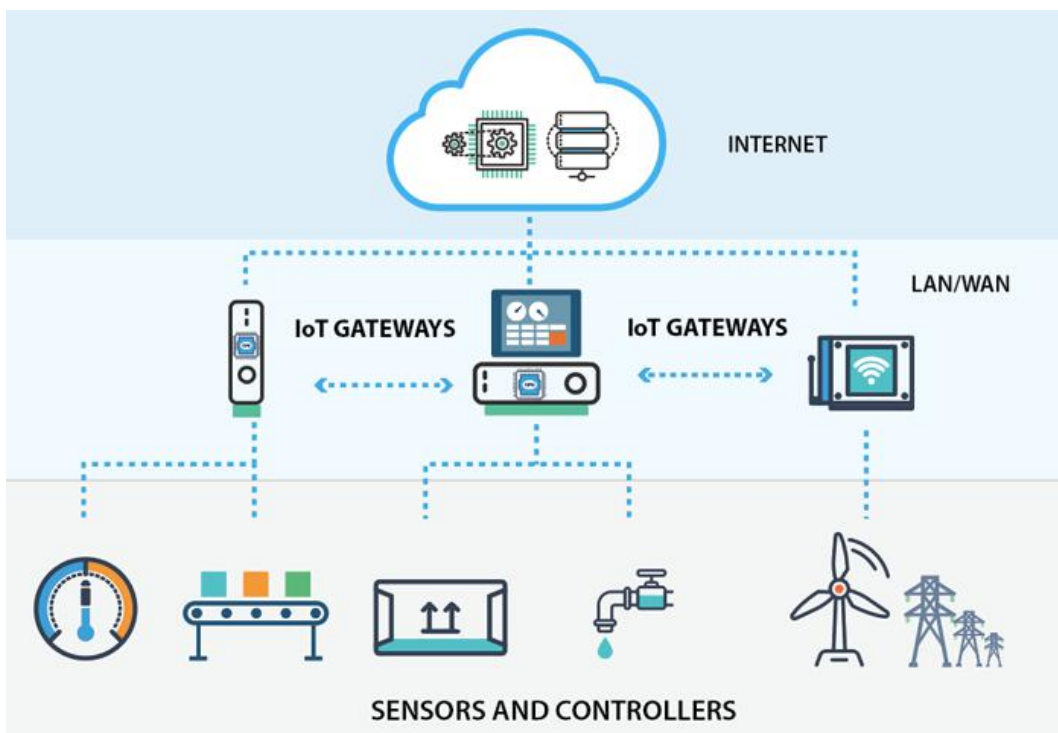
Content

Quick Start Guide with AWS IoT.....	1
Introduction.....	2
Prerequisites.....	2
1. Environment Preparation.....	3
1.1. Configuring the AWS IoT.....	3
1.1.1. Creating Things.....	4
1.1.2. Creating the Policy.....	8
1.1.3. Configuring the Certificate.....	9
1.2. Configuring the N510-H7 gateway.....	13
1.2.1. Hardware wiring.....	13
1.2.2. Basic Configuring.....	14
1.2.3. Upgrade the Software.....	18
1.2.4. Serial port configuration.....	19
2. Message Publishing and Subscription.....	19
2.1. Connecting to the AWS IoT.....	19
2.2. Publishing Messages to the AWS IoT.....	22
2.3. Subscribing to AWS IoT Messages.....	24
2.4. Data Collecting Configuration.....	25
2.4.1. Add a variable.....	26
3. Connecting a energy meter to AWS IoT.....	27
4. Factory reset.....	32

Introduction

AWS IoT Core lets you connect IoT devices to the AWS cloud without the need to provision or manage servers. AWS IoT Core can support billions of devices and trillions of messages, and can process and route those messages to AWS endpoints and to other devices reliably and securely. With AWS IoT Core, your applications can keep track of and communicate with all your devices, all the time, even when they aren't connected.

Whether it is in the context of industrial IoT or in connected homes, gateways are present in most architectural discussions. Now, we will show you how to implement the N510 MQTT gateway 'Bridge' capability to setup bi-directional exchange of data with AWS IoT Core through MQTT messages. This will enable your devices to communicate locally with the N510 gateway and with AWS IoT Core to benefit from the power of the AWS Cloud.



Prerequisites

- AWS cloud platform account
- USR-N510-H7 edge computing gateway
 - Firmware version V1.0.3 S or later
 - Hardware version N510-H7
- Network cable/serial port cable
- Setup software
 - EthernetTool-H7.exe

<https://www.pusr.com/support/downloads/USR-Ethernet-Tool.html>

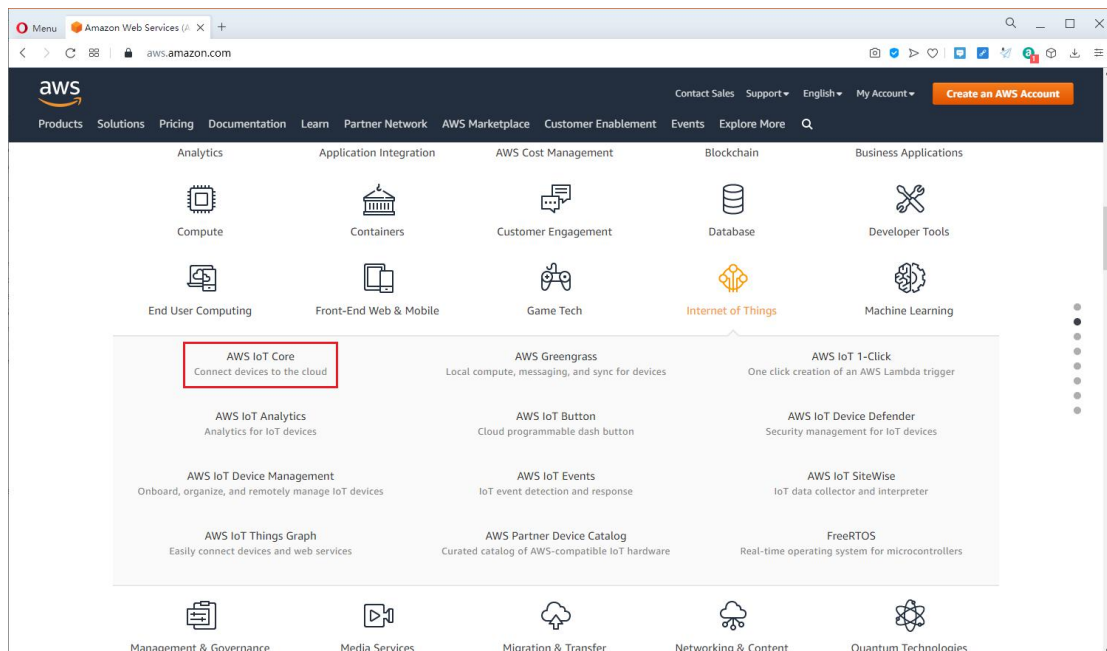
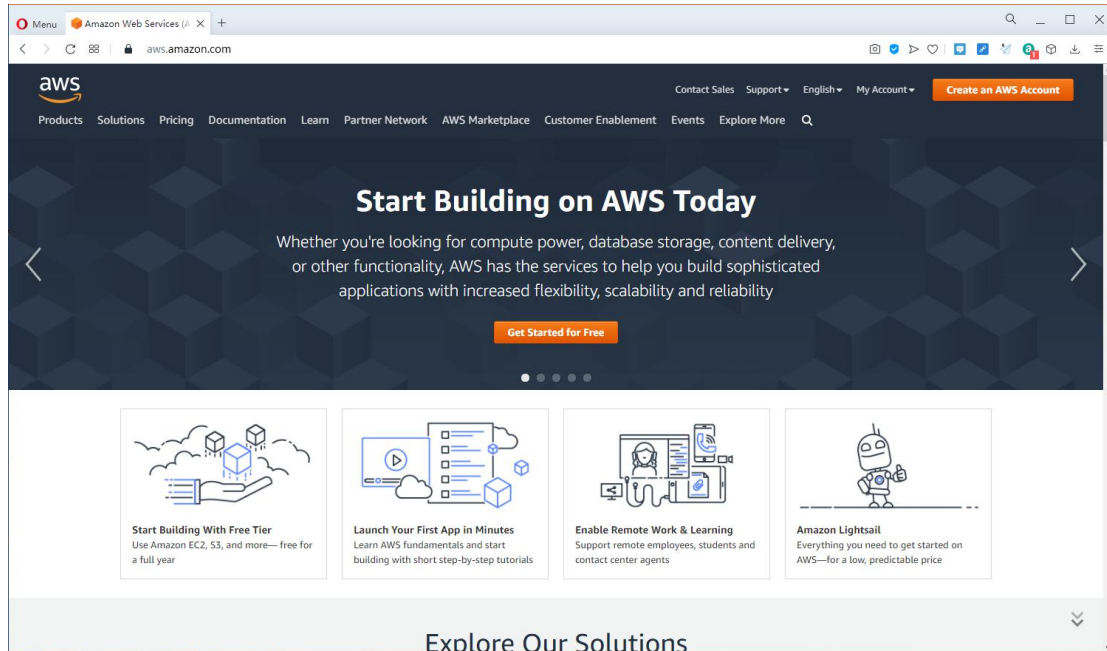
Please contact our sales for more ordering information.

1. Environment Preparation

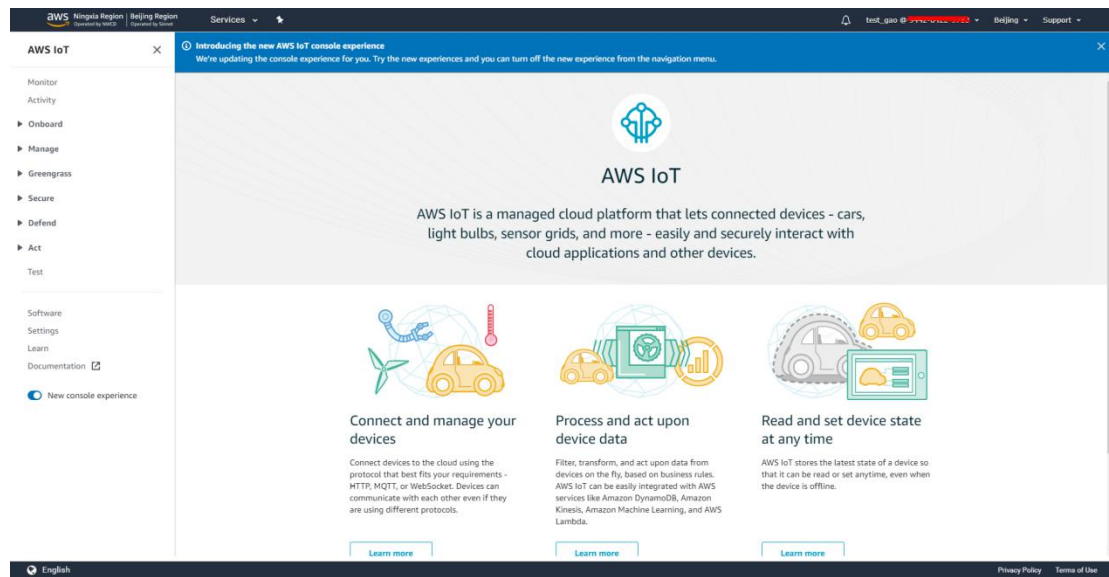
1.1. Configuring the AWS IoT

If you have configured the things, policy, and certificate in the AWS IoT console, go to [1.2 Configuring the N510-H7 gateway](#). Otherwise, perform the following steps to configure the AWS IoT console.

Visit <https://aws.amazon.com/>, log in to the IoT console, and choose **IoT Core**.

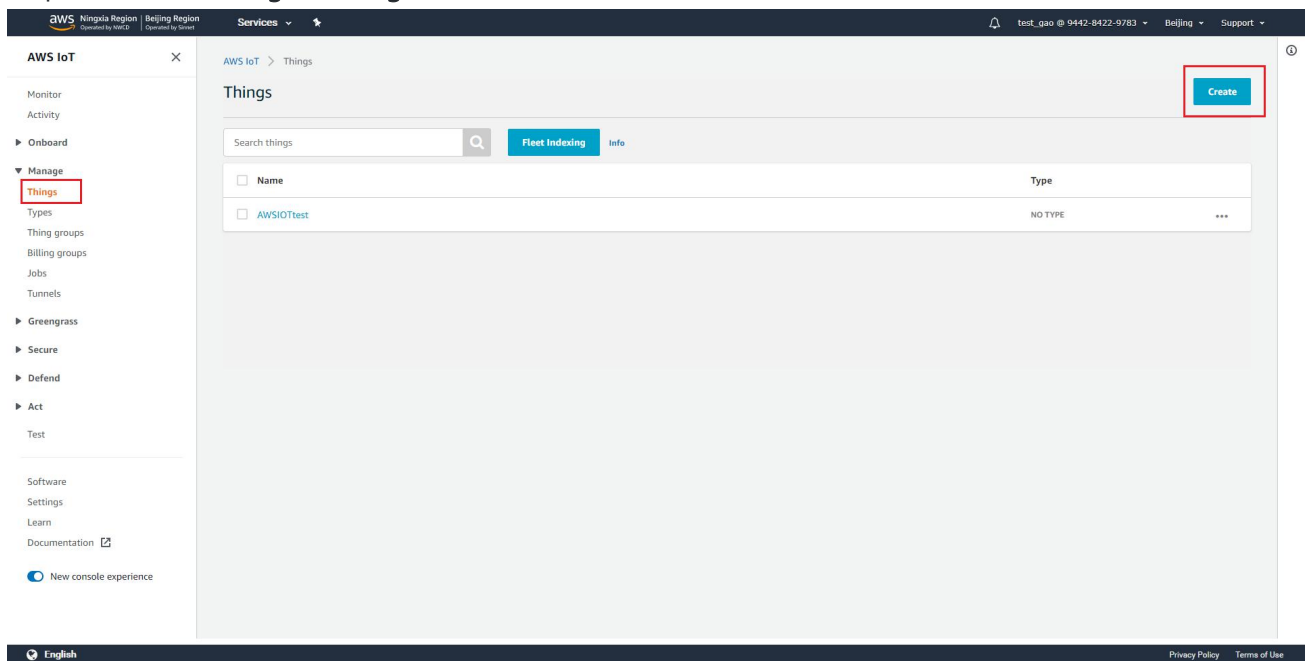


The following page appears after you log in to **IoT Core**



1.1.1. Creating Things

Step 1: Choose **Manage > Things** and click **Create**.



Step 2: Click **Create** a single thing.

AWS IoT > Things > Create things

Creating AWS IoT things

An IoT thing is a representation and record of your physical device in the cloud. Any physical device needs a thing record in order to work with AWS IoT. [Learn more.](#)

Register a single AWS IoT thing
Create a thing in your registry

Create a single thing

Bulk register many AWS IoT things
Create things in your registry for a large number of devices already using AWS IoT, or register devices so they are ready to connect to AWS IoT.

Create many things

Cancel **Create a single thing**

Step 3: Set the thing name, for example, USR-N510, retain the default values for other parameters and then click **Next**.

AWS IoT > Things > Create things

This step creates an entry in the thing registry and a thing shadow for your device.

Name
USR-N510

Apply a type to this thing
Using a thing type simplifies device management by providing consistent registry data for things that share a type. Types provide things with a common set of attributes, which describe the identity and capabilities of your device, and a description.

Thing Type
No type selected **Create a type**

Add this thing to a group
Adding your thing to a group allows you to manage devices remotely using jobs.

Thing Group
Groups / **Create group Change**

Set searchable thing attributes (optional)
Enter a value for one or more of these attributes so that you can search for your things in the registry.

Attribute key	Value
Provide an attribute key, e.g. Manufacturer	Provide an attribute value, e.g. Acme-Corporation

Clear

Add another

Show thing shadow

Cancel **Back** **Next**

Step 4: Click **Create certificate**.

CREATE A THING
Add a certificate for your thing
STEP 2/3

A certificate is used to authenticate your device's connection to AWS IoT.

One-click certificate creation (recommended)

This will generate a certificate, public key, and private key using AWS IoT's certificate authority.

Create certificate

Create with CSR

Upload your own certificate signing request (CSR) based on a private key you own.

Create with CSR

Use my certificate

Register your CA certificate and use your own certificates for one or many devices.

Get started

Skip certificate and create thing

You will need to add a certificate to your thing later before your device can connect to AWS IoT.

Create thing without certificate

Step 5: After the certificate is created, you need to **download** the certificate for this thing, the private key, and the root CA of the AWS IoT, **activate** the certificate and then click **Done**.

Note: Ensure the Certificate is "Active", otherwise activate it under ACTIONS in Things > Your Certificate > Security.

The screenshot shows the AWS IoT console interface. At the top, there are two green success banners: "Success Successfully created thing." and "Success Successfully generated certificate. Please download certificate files." Below these, a "Certificate created!" section provides instructions. It states: "Download these files and save them in a safe place. Certificates can be retrieved at any time, but the private and public keys cannot be retrieved after you close this page." It then lists three items for download in a table:

Item	File Name	Action
A certificate for this thing	9761b38626.cert.pem	Download
A public key	9761b38626.public.key	Download
A private key	9761b38626.private.key	Download

Below the table, it says: "You also need to download a root CA for AWS IoT: A root CA for AWS IoT" with a [Download](#) link. An [Activate](#) button is also present. At the bottom of the main content area, there are three buttons: "Cancel", "Done", and "Attach a policy". The "Done" button is highlighted with a red box.

Step 6: After you download "root CA", you will be lead to AWS IoT Developer Guide. Please find "Security > Authentication > Server Authentication" on the left. You should read through the guide. Then click "Amazon Root CA 1" and copy the code to create a `ca.pem`. **Keep these certificates in a folder for later use.**

Your Trust Worthy Industrial IoT Partner

6

www.pusr.com

CA certificates for server authentication

Depending on which type of data endpoint you are using and which cipher suite you have negotiated, AWS IoT Core server authentication certificates are signed by one of the following root CA certificates:

VeriSign Endpoints (legacy)

- RSA 2048 bit key: [VeriSign Class 3 Public Primary G5 root CA certificate](#)

Amazon Trust Services Endpoints (preferred)

Note
You might need to right click these links and select **Save link as...** to save these certificates as files.

- RSA 2048 bit key: [Amazon Root CA 1](#)
- RSA 4096 bit key: Amazon Root CA 2. Reserved for future use.
- ECC 256 bit key: [Amazon Root CA 3](#)
- ECC 384 bit key: Amazon Root CA 4. Reserved for future use.

These certificates are all cross-signed by the [Starfield Root CA Certificate](#). All new AWS IoT Core regions, beginning with the May 9, 2018 launch of AWS IoT Core in the Asia Pacific (Mumbai) Region, serve only ATS certificates.

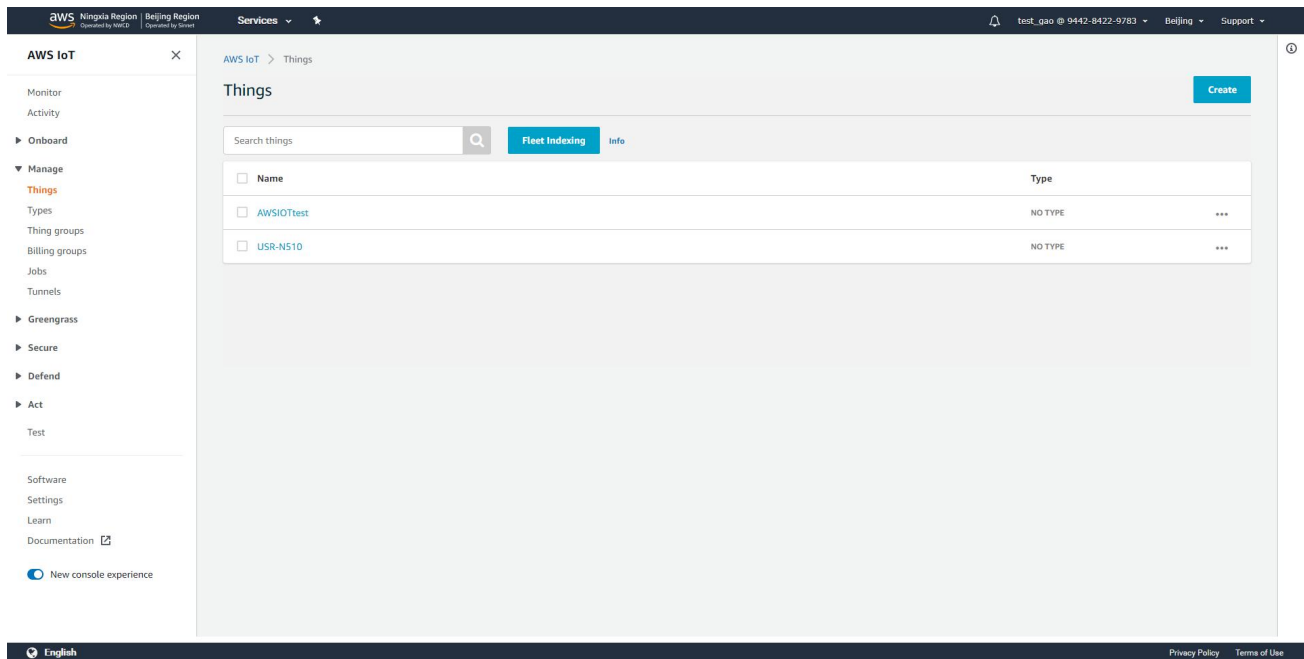
Server authentication guidelines

There are many variables that can affect a device's ability to validate the AWS IoT Core server authentication certificate. For example, devices may be too memory constrained to hold all possible root CA certificates, or devices may implement a non-standard method of certificate validation. For these reasons we suggest following these guidelines:

- We recommend that you use your ATS endpoint and install all supported Amazon Root CA certificates.

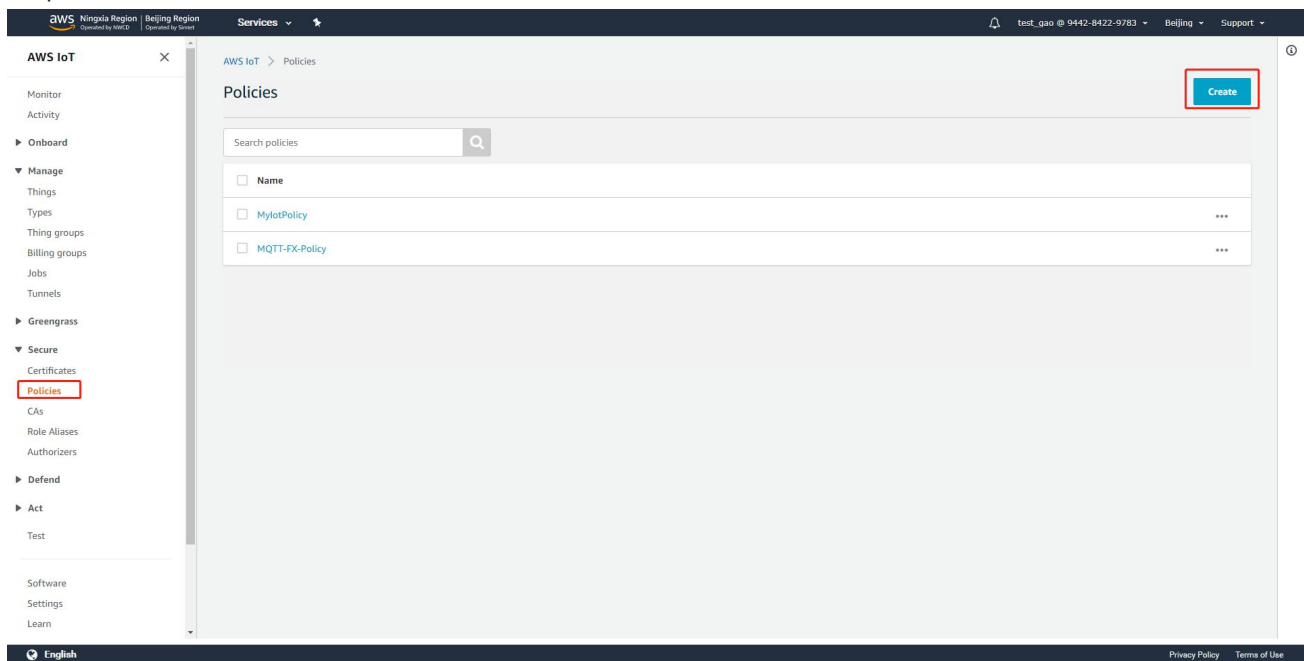
Name	Date modified	Type	Size
9761b38626-certificate.pem	3/10/2021 5:49 PM	Security Certificate	2 KB
9761b38626-private.pem.key	3/10/2021 5:49 PM	KEY File	1 KB
AmazonRootCA1.pem	5/11/2020 9:37 AM	PEM File	2 KB

After the thing is created, the following page is displayed



1.1.2. Creating the Policy

Step 1: Choose **Secure > Policies** and click **Create**.



Step 2: On the **Create a policy** page, enter the policy name, configure the policy by referring to the following settings, and then click **Create**. This policy allows all clients to connect to the AWS IoT.

- Enter `iot:*` in the Action text box.
- Enter `*` in the Resource ARN text box.
- Select **Allow** for **Effect**.

Create a policy

Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the [AWS IoT Policies documentation page](#).

Name:

Add statements
Policy statements define the types of actions that can be performed by a resource. Advanced mode

Action	iot*
Resource ARN	*
Effect	☒ Allow ☐ Deny Remove

Add statement Create

After the policy is created, the following page is displayed

Success
Successfully created a policy.

Policies Create

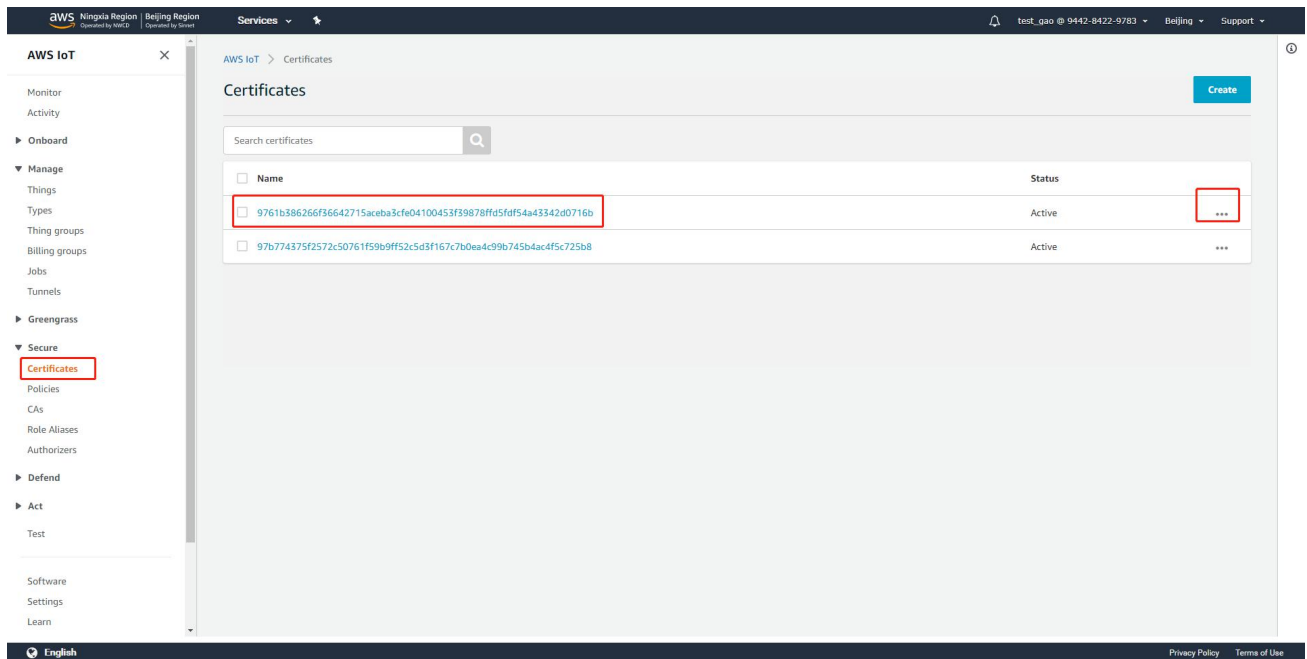
Search policies

Name	
☐ MyIotPolicy	...
☐ MQTT-FX-Policy	...
☒ AWSIoTTest	...

1.1.3. Configuring the Certificate

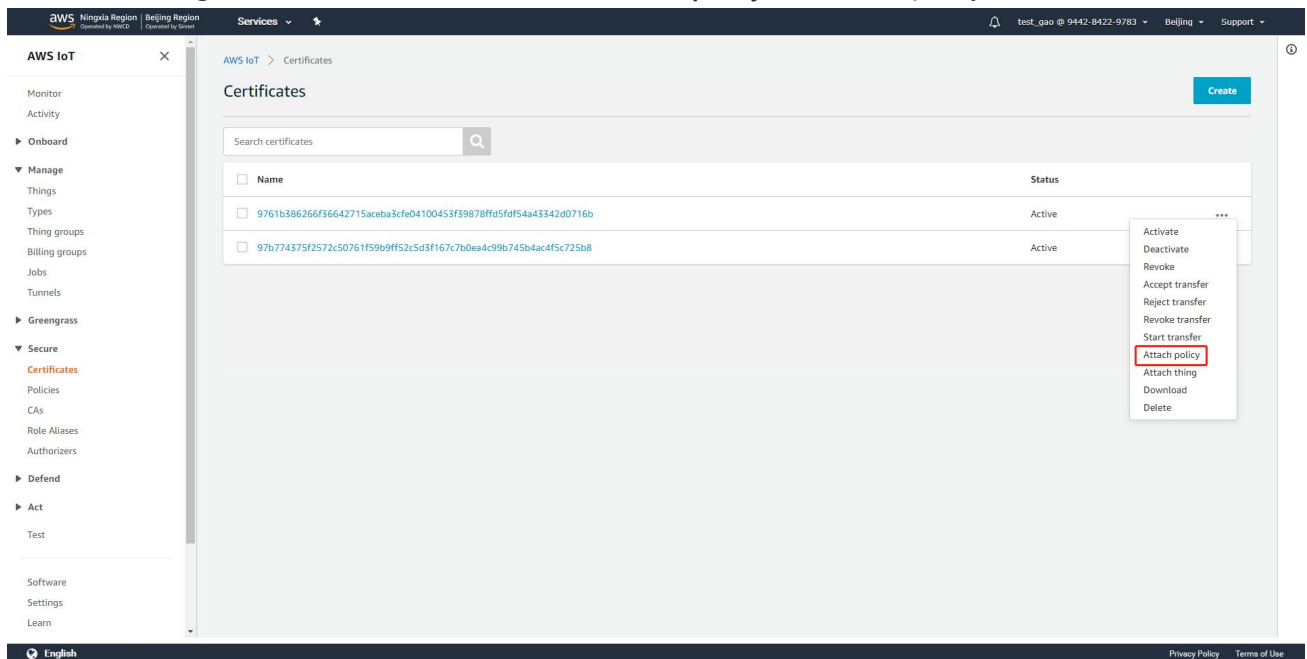
We already have a client certificate for our gateway, this certificate is associated with an IoT policy. This IoT policy will give permissions to the gateway to connect to AWS IoT Core and publish/subscribe to any topic (this policy must be further restricted to match your usage appropriately).

Choose **Secure > Certificates**. The following page is displayed



Step 1: Attaching a policy

Click ... to the right of the certificate and choose **Attach policy** to attach a policy to the certificate.



Select the policy that you have created, and click **Attach**.

Attach policies to certificate(s)

Policies will be attached to the following certificate(s):
9761b386266f36642715aceba3cfe04100453f39878ffd5fdf54a43342d0716b

Choose one or more policies

☐ MyIotPolicy	View
☐ MQTT-FX-Policy	View
☒ AWSIoTtest	View

1 policy selected

[Cancel](#) [Attach](#)

After the policy is attached, the following page is displayed

Success
Successfully attached policy.

AWS IoT > Certificates

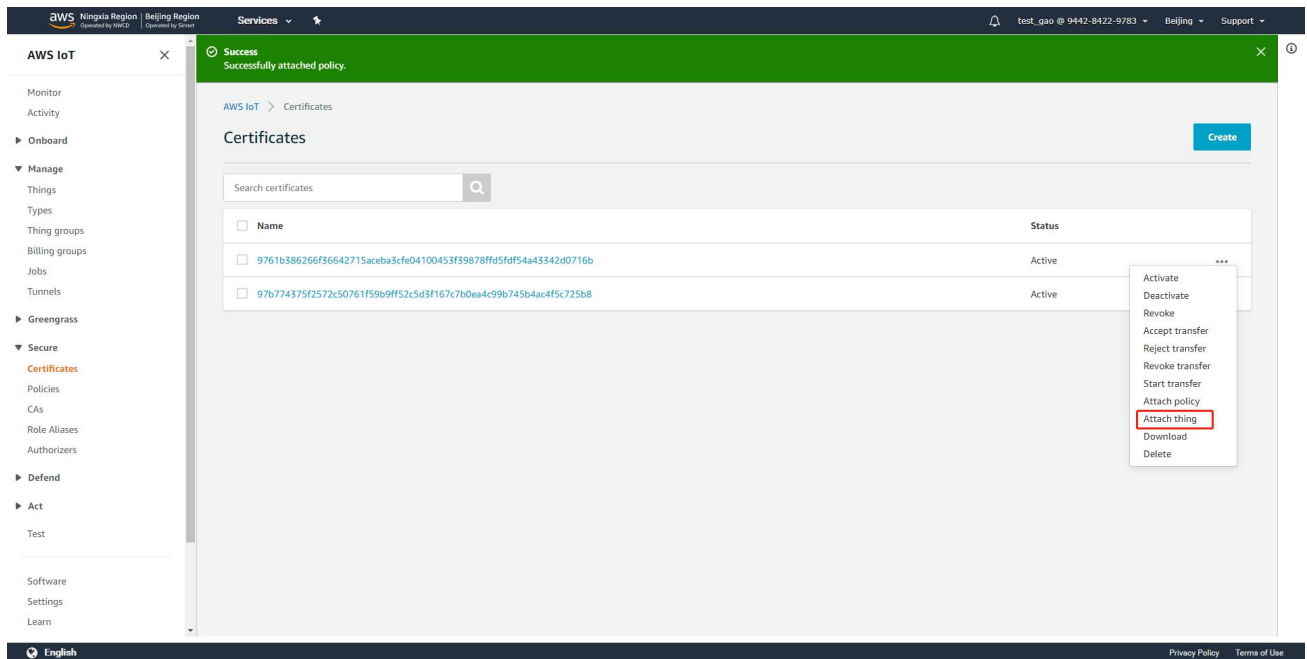
[Create](#)

Search certificates

☐ Name	Status	
☐ 9761b386266f36642715aceba3cfe04100453f39878ffd5fdf54a43342d0716b	Active	...
☐ 97b774375f2572c50761f59b9f52c5d3f167c7b0ea4c99b745b4ac4f5c725b8	Active	...

Step 2: Attaching a thing

Click ... to the right of the certificate and choose **Attach thing** to attach a thing to the certificate.



Select the thing that you have created, and click **Attach**.

Attach things to certificate(s)

Things will be attached to the following certificate(s):

9761b386266f36642715aceba3cfe04100453f39878ffd5fdf54a43342d0716b

Choose one or more things

Search things

☐ AWSIOTtest

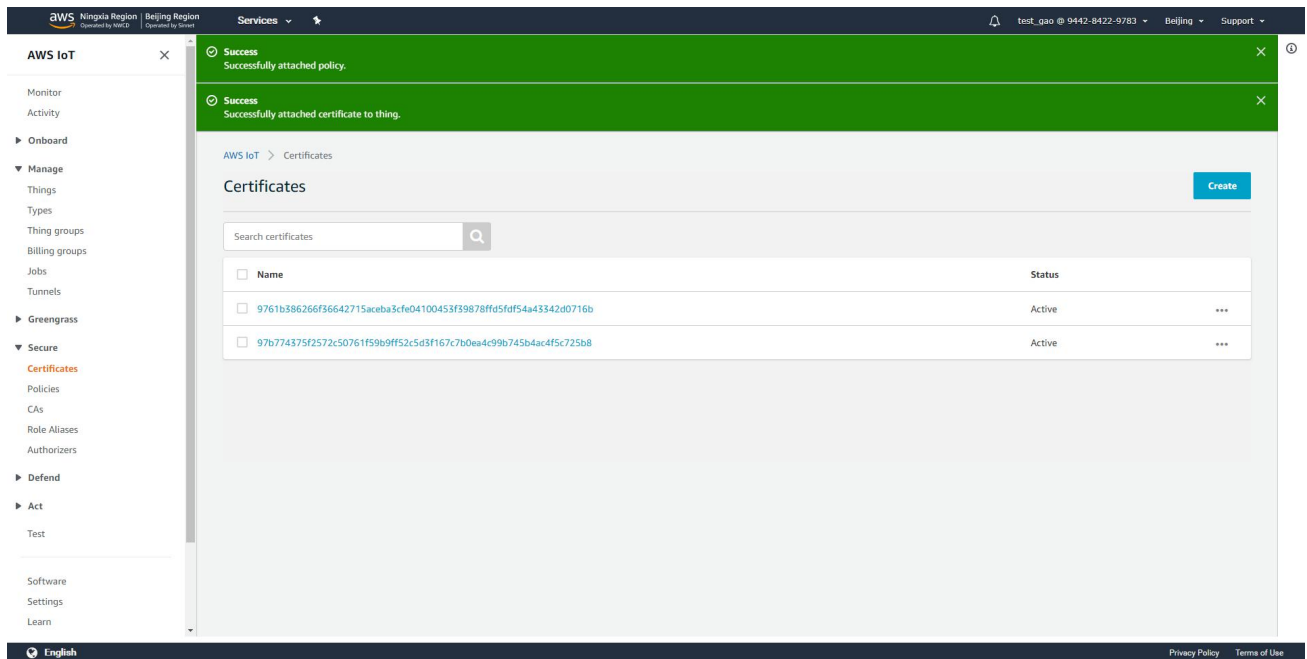
☒ USR-N510

1 thing selected

Cancel

Attach

After the thing is attached, the following page is displayed



The preparation of the AWS IoT environment is completed.

1.2. Configuring the N510-H7 gateway

1.2.1. Hardware wiring



Step 1: Connect N510 to PC via a net cable. Power on N510 and check the status of the device. The LED indicator would help:

Power: always on after power on

Work: blinking after system start.

Step 2: Connect N510 to PC via a serial port cable. The following figure describes how to connect serial port terminals of N510.

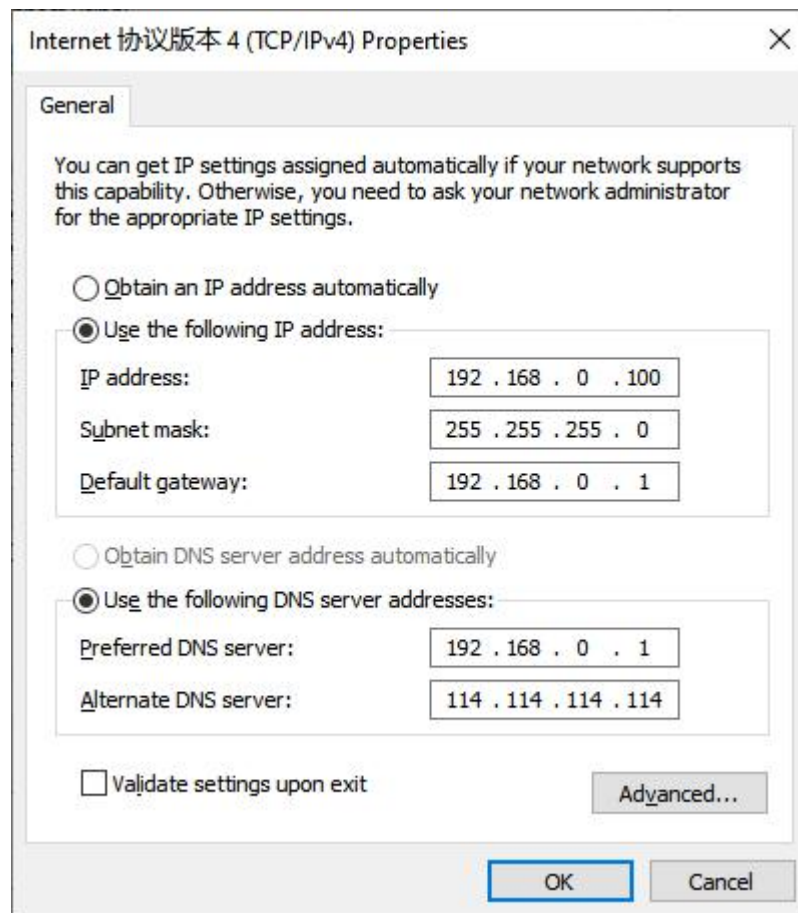
RS232/485 /422 Serial Port



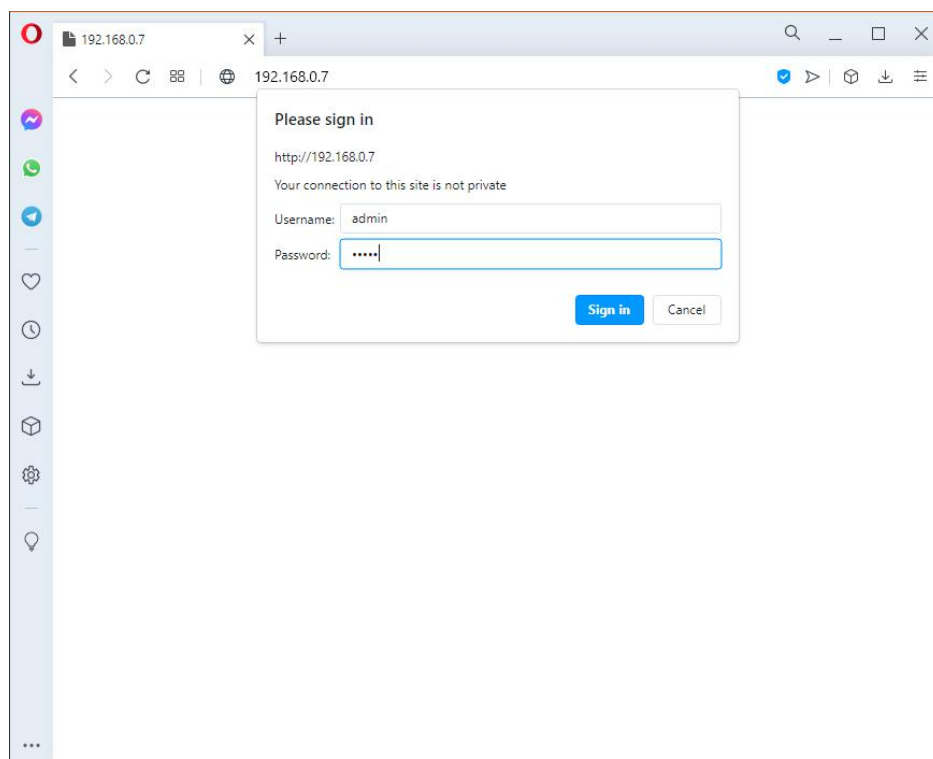
Item	Default Parameter
User name	admin
Password	admin
IP	192.168.0.7
Subnet mask	255.255.255.0
Network segment	192.168.0.1
Work mode	TCP Server
Local port	8899
Baud rate	115200
Parameter	None/8/1

1.2.2. Basic Configuring

Step 1: Set the PC's IP address to be on the same subnet with N510. By default, the IP address of N510 is 192.168.0.7. Set a fixed IP address. Select Use the following IP address, enter an IP address (By default, any from 192.168.0.2 to 192.168.0.254 except 192.168.0.7), subnet mask (By default, 255.255.255.0), default gateway (By default, 192.168.0.1), and DNS server address, and click OK.



Step 2: Launch the browser on the PC and access the IP address of N510 (192.168.0.7 for default) . Enter the login user name and password. The default user name and password are admin and admin respectively.



Step 3: After successful login, you can see the web page as shown below

Version: V1.0.3-S Type: H7 中文

USR IOT
-IOT Experts-

Be Honest, Do Best!

Overview	Overview	Tips
- Network Settings - Port1 - MQTT - Packet Forwarding - Websocket to Serial - Upload Certificates - Miscellaneous - System	Model: USR-N510 Firmware Version: V1.0.3-S Local IP Address: 192.168.0.7 Preferred DNS Server: 114.114.114.114 Alternate DNS Server: 223.5.5.5 MAC: 9C-A5-25-C4-F2-F9 System Uptime: 0day: 0hour: 0min Serial Port No.: port1 Link A: LISTEN TX Count of A: 0 bytes RX Count of A: 0 bytes Link B: IDLE TX Count of B: 0 bytes RX Count of B: 0 bytes	• System Uptime: The accumulative running time of the device after power on • TX/RX Count: Total number of bytes sent and received

Copyright © Jinan USR IOT Technology Limited. All Rights Reserved website: www.pusr.com

Step 4: To change the user name and password for logging in to the web management interface of N510, choose **Miscellaneous** page and set the new user name and password.

Version: V1.0.3-S Type: H7 中文

USR IOT
-IOT Experts-

Be Honest, Do Best!

Overview	Miscellaneous Settings	Tips
- Network Settings - Port1 - MQTT - Packet Forwarding - Websocket to Serial - Upload Certificates - Miscellaneous - System	Model Name: USR-N510 Websocket Port: 6432 Websocket interface: UART1 Web Server Port: 80 User Name: admin Password: admin Caching Serial Data: ☐ Interval of Restarting Without Data: 0 (60~65535) Save Cancel	• Model Name: The maximum length of the name is 32 bytes • Websocket Port: 6342(default). The server port number of the Websocket function. • Websocket Interface: The interface corresponding to the Websocket function, the default is UART1. • Web Server port: Web server port of this device. • User Name: admin (default) • Password: admin (default) • Caching Serial Data: Serial data will be

Copyright © Jinan USR IOT Technology Limited. All Rights Reserved website: www.pusr.com

Step 5: To change the IP address of N510, choose **Network Settings** page to configure network parameters. Now we have change the Method of IP obtaining to DHCP in order to connect N510 to the internet by a router or switch with net cable. **All parameter Settings take effect after restart.**

Version: V1.0.3-S Type: H7 中文

USR IOT
-IOT Experts-

Be Honest, Do Best!

Overview
Network Settings
Port1
MQTT
Packet Forwarding
Websocket to Serial
Upload Certificates
Miscellaneous
System

Network Settings

Method of IP Obtaining:

Method of DNS Setting:

IP Address:

Subnet Mask:

Gateway:

Preferred DNS Server:

Alternate DNS Server:

Tips

- **Method of IP Obtaining:** Static IP or DHCP
- **IP address :** Local IP address of this device
- **Subnet Mask:** 255.255.255.0 (default)
- **Gateway:** IP address of your router

Copyright © Jinan USR IOT Technology Limited. All Rights Reserved website: www.pusr.com

Version: V1.0.3-S Type: H7 中文

USR IOT
-IOT Experts-

Be Honest, Do Best!

Overview
Network Settings
Port1
MQTT
Packet Forwarding
Websocket to Serial
Upload Certificates
Miscellaneous
System

Firmware Update

Firmware File:

Restore Factory Defaults

Reboot

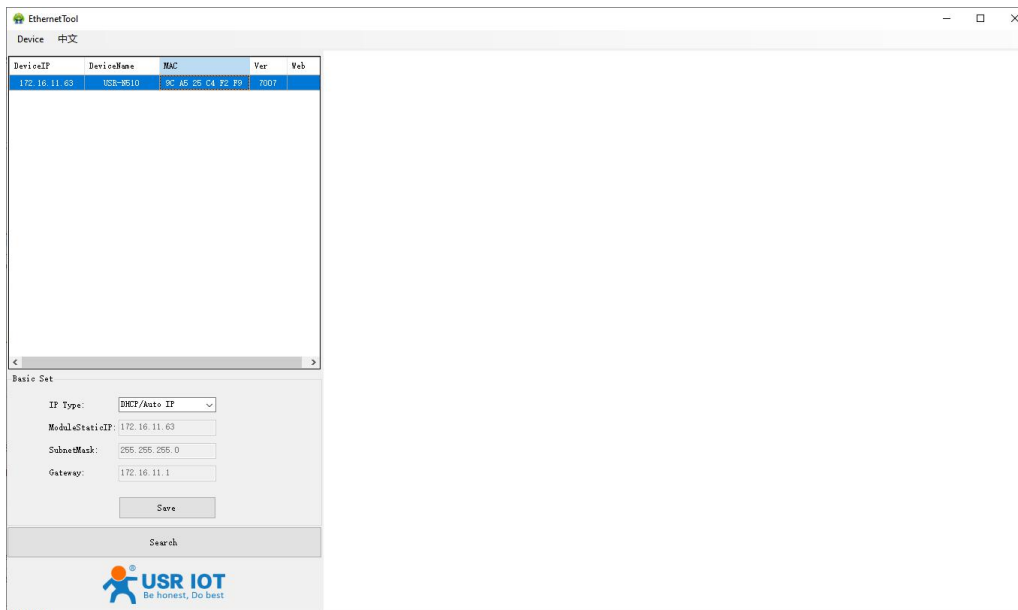
Tips

- **Firmware Update:** Flash the uploaded firmware file.
- **Restore Factory Defaults:** This function will reset all the parameters of the devices to their factory defaults. Take care.
- **Reboot:** The new configurations take effect after the device rebooted.

Copyright © Jinan USR IOT Technology Limited. All Rights Reserved website: www.pusr.com

Step 6: Put N510 and PC in the same subnet segment, e.g. under same router or switch. Open the setup software. You can download this software from <https://www.pusr.com/support/downloads/USR-Ethernet-Tool.html>

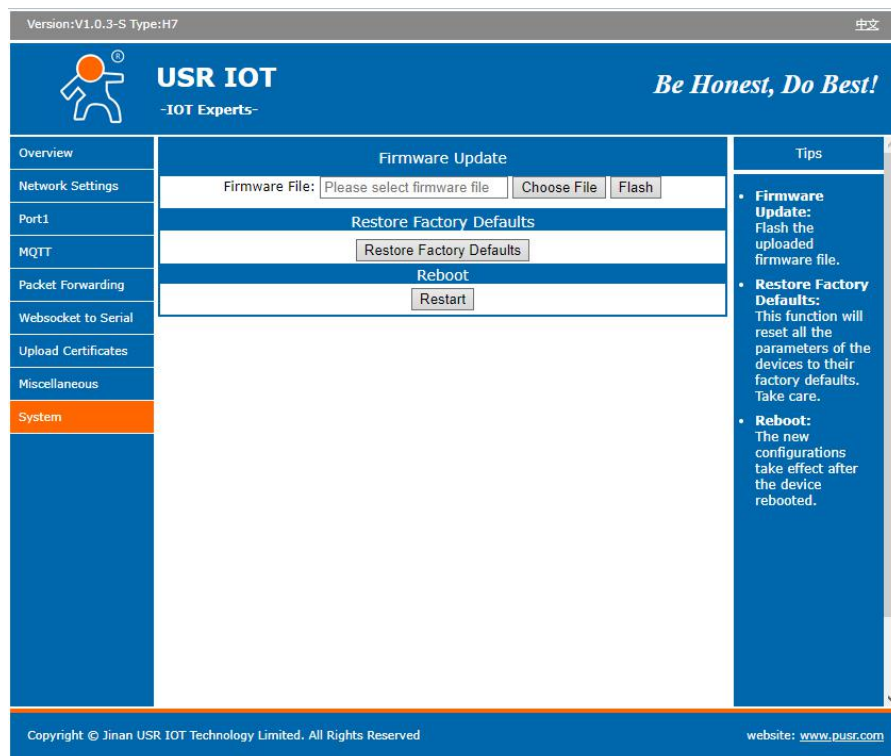
Click **Search**, you will find the N510, and configure it easily.



1.2.3. Upgrade the Software

To obtain the latest software version of N510 and updated functions, contact the technical support center or our sales. To update the N510 software version, do as follows.

Choose **System > Firmware Upgrade**. Select a firmware file and click Flash. After the update is completed, you are prompted to Restore Factory Defaults to apply the new firmware. Reopen the web server to continue.



1.2.4. Serial port configuration

In the serial port setting module, you can configure communication parameters for the serial ports RS485 and RS232, as shown in the figure below. We now configure the gateway's serial port to operate as required for the attached devices. Go to the Port1 Settings tab and enter the following settings.

Version: V1.0.4-S Type: H7 中文

USR IOT
-IOT Experts- *Be Honest, Do Best!*

Serial Port Settings

Baud Rate: 115200 bps(600~230400)
 Data Bits: 8 bit
 Parity: None
 Stop Bits: 1 bit
 Serial Port Selecting: Dial Switch
 Current Serial Port Type: RS232
 Flow control: NONE
 Packaging Time: 0 (0~255)ms
 Packaging Length: 0 (0~1460)chars
 RFC2217: ☒
 Serial Heartbeat Packet: ☐

Socket Parameters

Operating Mode: TCP Server None
 Max. No. of Client Connecting: 8 After Exceeding the Max. No. KICK
 Local Port: 23 (1~65535)
 Network Print Driver: ☐
 Modbus Poll: ☐ Timeout: 200 (10~9999)ms
 Modbus Abnormal response: ☐
 Network Heartbeat Packet: ☐
 Registry Packet format: None when: Connection String

Tips

- COM Port Settings:** Baud Rate, Data Bit, Parity, Stop Bit, Serial Operating Mode.
- IP Settings:** TCP Server, TCP Client, UDP Server, UDP Client, HTTP Client.
- Other Options:** Modbus TCP, Heartbeat Packet, RFC2217.

Copyright © Jinan USR IOT Technology Limited. All Rights Reserved website: www.pusr.com

Before you test the data, connect N510 and PC with a USB to serial port cable, and keep parameters consistent.

2. Message Publishing and Subscription

The topics that start with a dollar sign (\$) are reserved for the AWS IoT. You can subscribe to and publish messages to these topics. However, you cannot create topics with a "\$" prefix. Prohibited message publishing or subscription for the reserved topics may cause connection failures. For details about the topics reserved for the AWS IoT, see [Reserved Topics](#).

2.1. Connecting to the AWS IoT

Step 1: Open the built-in web server of N510, then choose **Upload Certificates** menu. Upload the files you downloaded in [1.1.1 Creating Things Step 5 and Step6](#).

Import "[AmazonRootCA1.pem](#)" to "SERVER CA File", "certificate.pem" to "Client CA File" and "private.pem.key" to "Client Private Key". Client Key Password: any string. Click "Save" to make changes take effect.

Version: V1.0.3-S Type: H7 中文


USR IOT
 -IoT Experts-

Be Honest, Do Best!

Overview	Upload Certificates: Client Key Password: 123456 Save Server CA File: AmazonRootCA1.pem Choose file Upload Client CA File: 8cc3acf9a6a67dbbb604d707 Choose file Upload Client Private Key: 8cc3acf9a6a67dbbb604d707 Choose file Upload	Tips • Client Key Password: Encrypt the client private key, it prevents someone from downloading and changing the key. Note CA file must be pem format
Network Settings		
Port1		
MQTT		
Packet Forwarding		
Websocket to Serial		
Upload Certificates		
Miscellaneous		
System		

Copyright © Jinan USR IOT Technology Limited. All Rights Reserved website: www.pusr.com

Step 2: MQTT broker profile settings.

The following is a configuration example

Version: V1.0.4-S Type: H7 中文


USR IOT
 -IoT Experts-

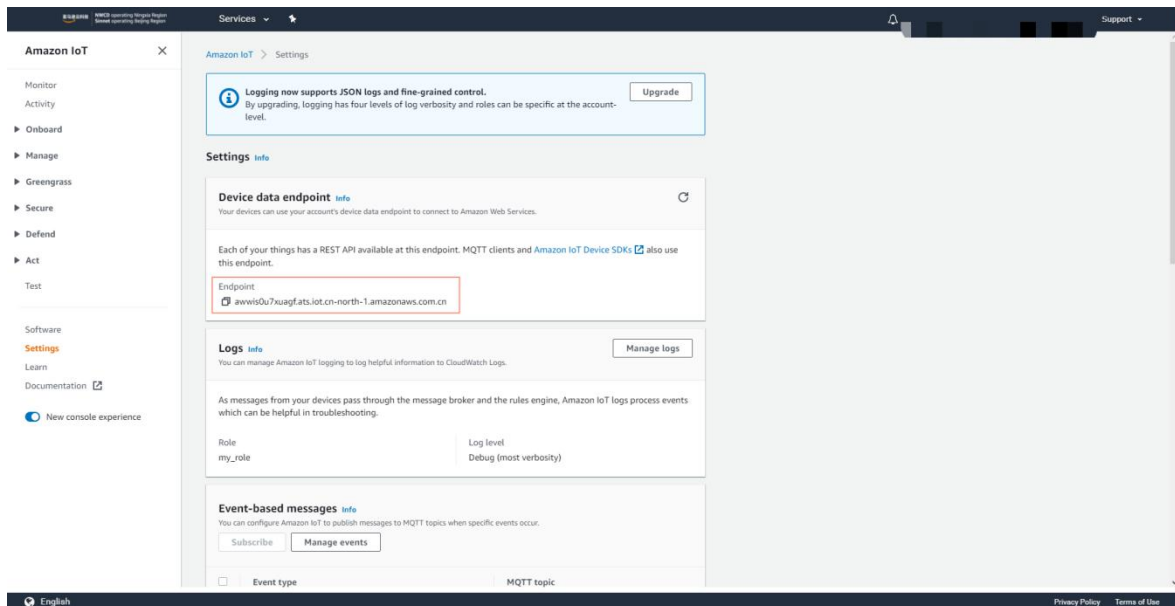
Be Honest, Do Best!

Overview	Basic Settings Topic Publish Topic Subscribe MQTT: Enable Disable MQTT Version: MQTT-3.1.1 Broker Address: awwis0u7xuagf.ats.iot.cn [N/A] Local/Remote Port NO.: 0 8883 (1~65535) Keep Alive Interval: 60 (1~65535)s Connection Timeout: 0 (0~99999)s User Credentials: ☒ User Name: admin Password: admin Client ID: 123456 Clean Session: ☒ Last-Will: ☐ Encryption mode: ssl1.2 verification method: Verify all Save Cancel	Tips • Keep Alive interval: Time interval for sending PINGREQ message. • User Credentials: Contains the Username, password, Client ID, and other information required for MQTT connection. • Clean session: Whether the broker continues to store/maintain the client's subscription status after the client disconnects. • Enable Last-Will: Closed by default, contains the Last-will topic, Last-will message, Last-will QoS, Last-will Retain
Network Settings		
Port1		
MQTT		
Packet Forwarding		
Websocket to Serial		
Upload Certificates		
Miscellaneous		
System		

Copyright © Jinan USR IOT Technology Limited. All Rights Reserved website: www.pusr.com

The parameters are described as follows:

Broker Address: Endpoint address of the AWS IoT, which can be obtained from the **Settings** page of the AWS IoT.



Remote Port:8883 default for AWS IoT.

User credentials: checked.

Username and password: any string. These items are not used in MQTT connection.

Client ID: any unique ID.

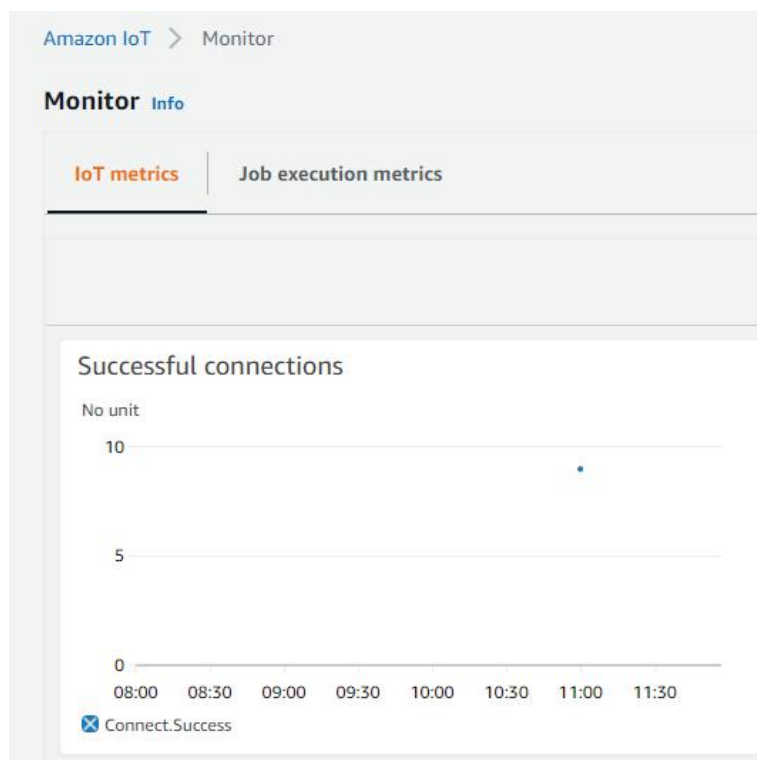
Encryption mode: select SSL V1.2

Verification method: verify all.

Use the default values for other parameters.

Click **"Save"** to make changes take effect. **All parameter Settings take effect after restart.**

The N510 gateway has now restarted and has already connected to AWS IoT Core in the background.

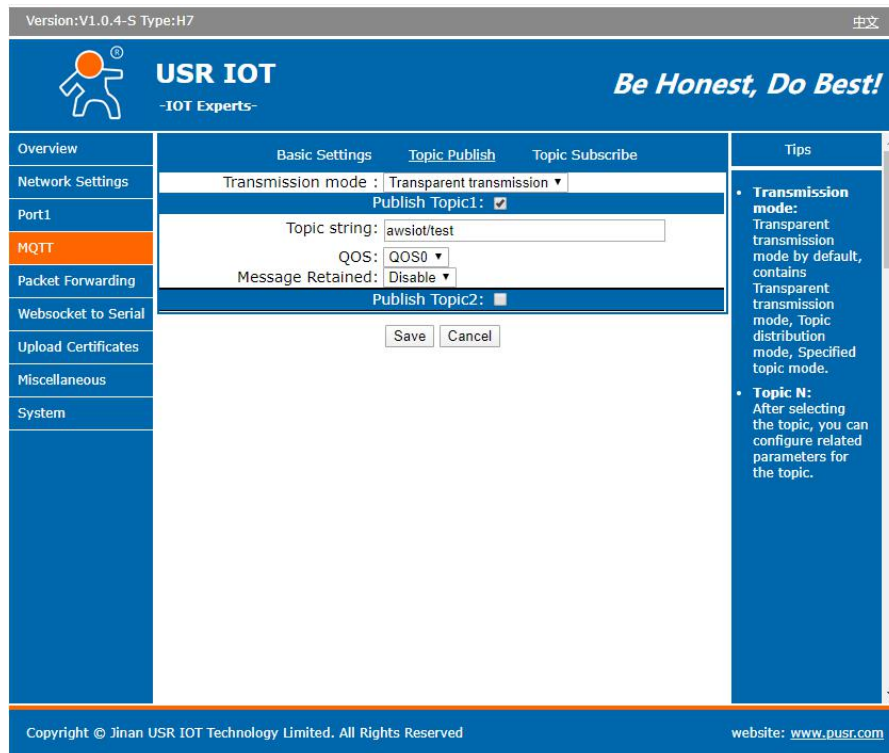


If something goes wrong, you can check N510 settings or contact USR IOT technical support for help.

2.2. Publishing Messages to the AWS IoT

Step 1: Configure the message to be published.

Choose **MQTT** -> **Topic Publish** and add the message to be published. The following figure shows the configuration



The message publishing parameters are described as follows:

Transmission mode:

① transparent transmission, just as its name implies, the gateway does not process any data, forward the data directly.

② Topic distribution, in this mode users should define a distribution header when configuring the topic. After receiving serial port data, the device pushes it to the associated topics based on the identifiers(headers). Different topics are allowed to have the same identifier(header). In this way, data is pushed to all topics with the same header. Header and payload are separated by comma. For example, header, {"message": "Hello from USR-N510 gateway"} as serial data.

③ Specified topic, add topic, Qos(0,1,2), retain(ON,OFF) message before payload. After receiving the serial data, N510 publishes the payload of the corresponding topic to cloud according to rules. This method can be used to publish any topic at any time. For example, awsiot/test,0,ON,{"message": "Hello from USR-N510 gateway"} as serial data.

Topic string: publication topic, which must be consistent with the topic that the MQTT server subscribes to.

QoS: publication QoS, which is recommended to be consistent with that of the MQTT server.

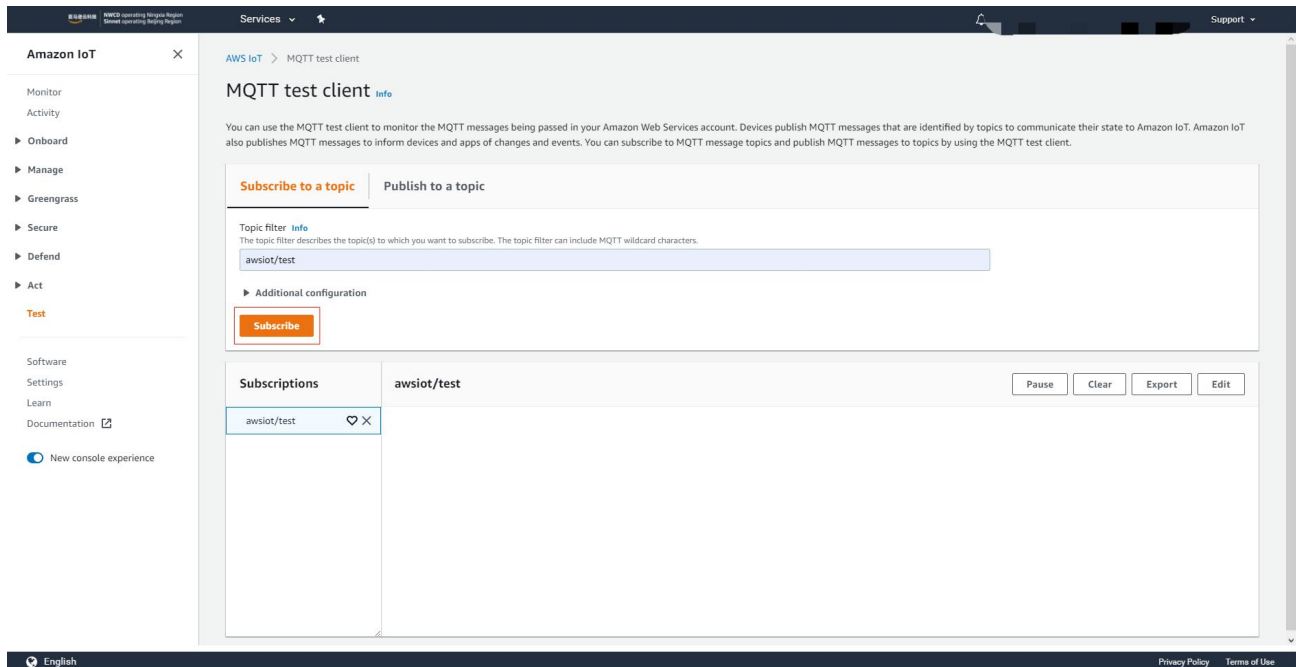
- 0: The message is sent only once, without retry.
- 1: The message is sent at least once to ensure that it reaches the MQTT server.
- 2: The message is sent exactly once. This level guarantees that each message is received only once by the intended recipients.

Message Retain: The broker stores the last retained message and the corresponding QoS for that topic. Each client that subscribes to a topic pattern that matches the topic of the retained message receives the retained message immediately after they subscribe.

Step 2: Subscribe to messages in the AWS IoT.

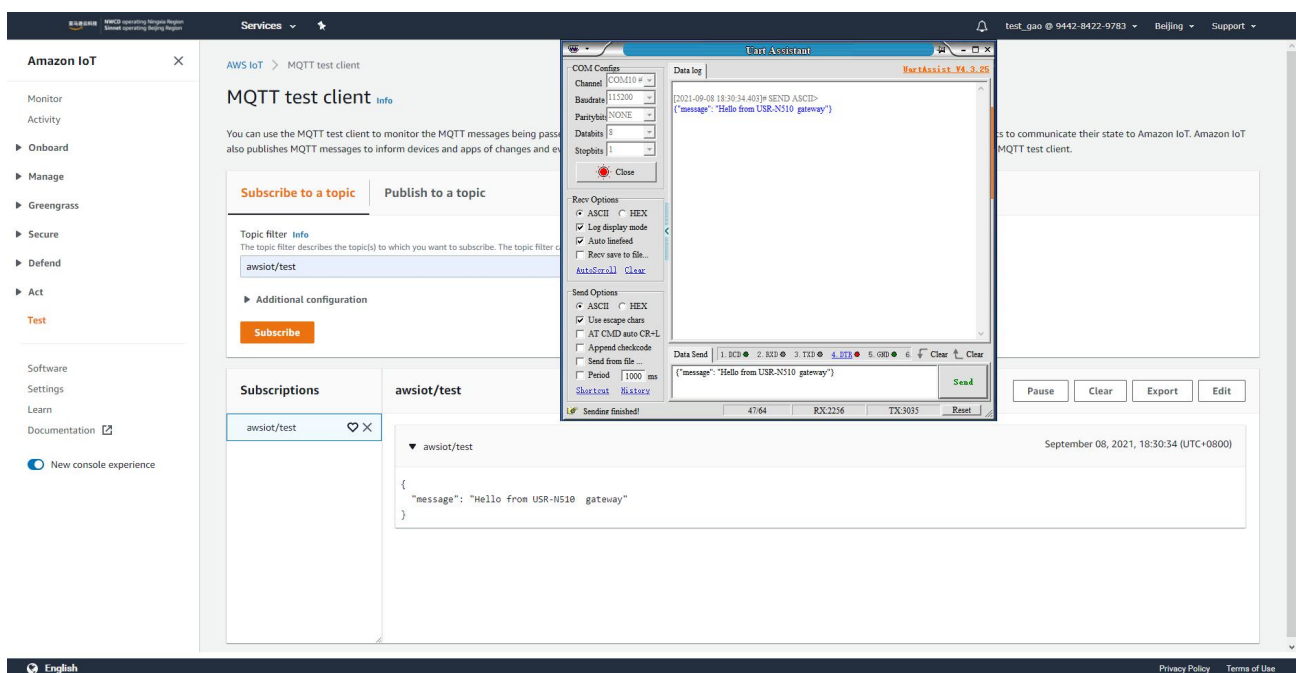
We will check that the topic is functioning properly, meaning that messages will be forwarded from the N510 gateway to AWS IoT Core.

Choose **AWS IoT > Test** and enter the N510 publication topic in the Subscription topic text box. As an example, the topic is **awsiot/test** and click on Subscribe to topic.



Step 3: View the messages that the AWS IoT receives.

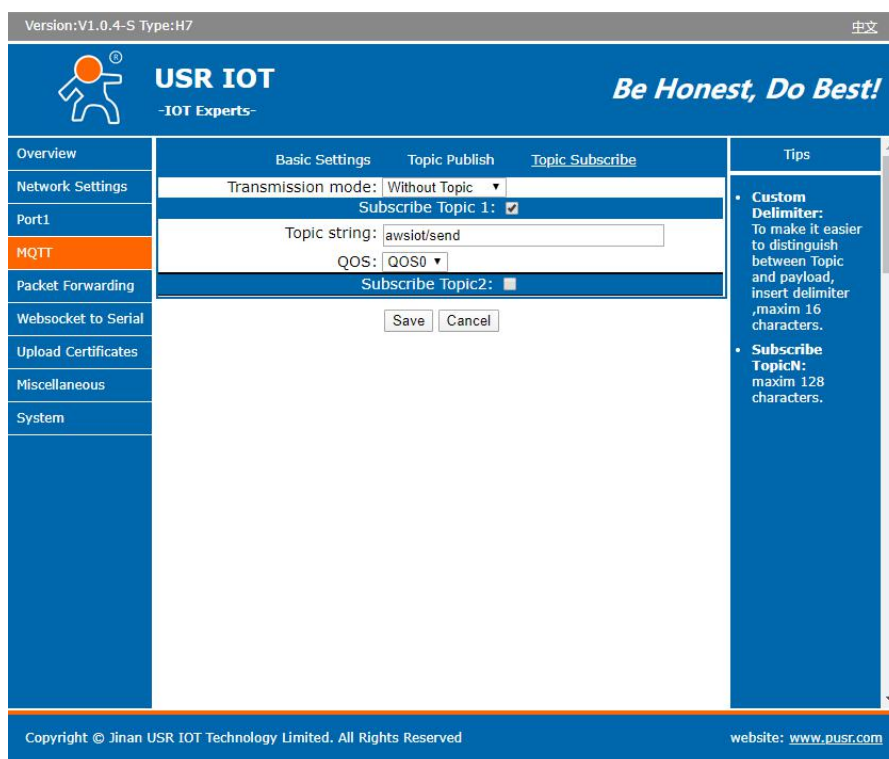
After subscribing to the topic, you can view the message content under the topic.



2.3. Subscribing to AWS IoT Messages

Step 1: Configure the message for subscription.

Choose **MQTT** -> **Topic Subscribe** and add the message to be published. The following figure shows the configuration:



The message subscription parameters are described as follows:

Transmissions mode: The data printed to the serial port can be set whether to carry the Topic

Topic string: Subscription topic, which must be consistent with the data topic published by the MQTT server.

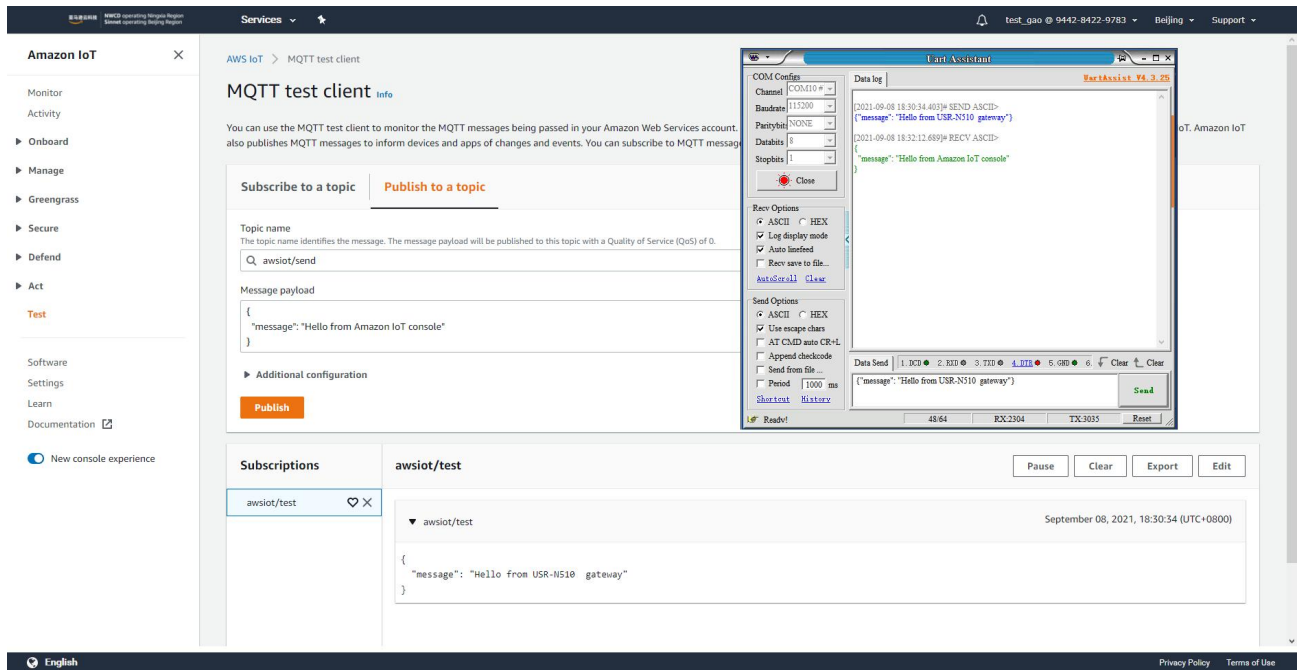
QoS: subscription QoS. The default value is 0.

Step 2: Publish messages in the AWS IoT.

Choose **AWS IoT** > **Test** and enter the N510 subscription topic in the **Publish to topic** text box. As an example, the topic is **awsiot/send**.

Step 3: View the messages that the AWS IoT publishes.

After the AWS IoT publishes messages, you can view the received messages in the data log of the **Uart Assistant**.



2.4. Data Collecting Configuration

N510 allows users to collect, process and upload data to the cloud conveniently and plays the role of transforming serial ModbusRTU data into Json format and MQTT protocol.

It defines how the serial data received by N510 be packaged when uploads it to AWS IoT Cloud.

Choose JSON and Enable the JSON functions. We can define payload structure in JSON Template.

Reporting method:

- ✓ Periodic: Collect the variable data at the poll interval and report the data at the report interval.
- ✓ Value update: Collect the variable data at the poll interval and report the data only when the data changes .

Version: V1.0.4-S Type: H7 中文


USR IOT
 -IOT Experts-

Be Honest, Do Best!

Overview	Node(1-5)	NODE(6-14)	NODE(15-23)	NODE(24-32)	Tips
Network Settings	Port1: JSON: ☐ Enable Add data header: ☐ Disable Operating mode: ☐ MQTT Topic: Topic1 Reporting method: ☐ Periodic Reporting interval: 10 s				JSON: Settings for Serial Message Format /RTU JSON Message Definition Node(1-5..) Operating mode: MQTT(default);Socket JSON template: The template should conform to the JSON format and contain key-value pairs of all nodes Device ID Contains the MAC, SN, and custom ID of the device. Timestamp Contains the Standard timestamps and UNIX timestamps Node N Configure the Modbus command
MQTT	JSON Template(<1000byte): <pre>{ "Voltage": "data1", "Current": "data2", "ActivePower": "data3", "Location": "bedroom", "DEVID": "MAC" }</pre>				
Packet Forwarding	Fixed Attribute: Device ID: ☒ Timestamp: ☒ Key-Value Pairs: "DEVID":"MAC" Key-Value Pairs: "TIME":"NTPTime" ID: MAC Timestamp Type: GMT Quotation Mark: ☒ Quotation Mark: ☒				
Websocket to Serial	NODE1: ☒ Key-Value Pairs: "Voltage":"data1" Poll interval: 500 (10~99999ms) Slave ID: 1 (0~255) Response Timeout: 2000 (10~99999ms) Register Address: 0 (0~65535) Unit: ☒ Function Code: 3 Unit content: V Data format: 32-bit float(A) String Filling Padding: ☒ Formula: %s String Filling: FF Decimal Digits 0-6: 2 Quotation Mark: ☒				
Upload Certificates					
Miscellaneous					
System					

Copyright © Jinan USR IOT Technology Limited. All Rights Reserved website: www.pusr.com

2.4.1. Add a variable

On the Packet Forwarding page, click NODE1, and configure the variable parameters in the pop-up box for adding variables.

Version: V1.0.4-S Type: H7 中文


USR IOT
 -IOT Experts-

Be Honest, Do Best!

Overview	Node(1-5)	NODE(6-14)	NODE(15-23)	NODE(24-32)	Tips
Network Settings	Port1: JSON: ☐ Enable Add data header: ☐ Disable Operating mode: ☐ MQTT Topic: Topic1 Reporting method: ☐ Periodic Reporting interval: 10 s				JSON: Settings for Serial Message Format /RTU JSON Message Definition Node(1-5..) Operating mode: MQTT(default);Socket JSON template: The template should conform to the JSON format and contain key-value pairs of all nodes Device ID Contains the MAC, SN, and custom ID of the device. Timestamp Contains the Standard timestamps and UNIX timestamps Node N Configure the Modbus command
MQTT	JSON Template(<1000byte): <pre>{ "Voltage": "data1", "Current": "data2", "ActivePower": "data3", "Location": "bedroom", "DEVID": "MAC" }</pre>				
Packet Forwarding	Fixed Attribute: Device ID: ☒ Timestamp: ☒ Key-Value Pairs: "DEVID":"MAC" Key-Value Pairs: "TIME":"NTPTime" ID: MAC Timestamp Type: GMT Quotation Mark: ☒ Quotation Mark: ☒				
Websocket to Serial	NODE1: ☒ Key-Value Pairs: "Voltage":"data1" Poll interval: 500 (10~99999ms) Slave ID: 1 (0~255) Response Timeout: 2000 (10~99999ms) Register Address: 0 (0~65535) Unit: ☒ Function Code: 3 Unit content: V Data format: 32-bit float(A) String Filling Padding: ☒ Formula: %s String Filling: FF Decimal Digits 0-6: 2 Quotation Mark: ☒				
Upload Certificates					
Miscellaneous					
System					

Copyright © Jinan USR IOT Technology Limited. All Rights Reserved website: www.pusr.com

Key-Value Pairs: The variable name. (The variable name must be unique in the same device.)The value must be the same as a key value pair in the template.

Register Address: The variable register address.The configuration is based on hexadecimal data.e.g.

Function code:supports reading coils and register,code 1,2,3,4

Data format: The variable data type, including:

- ✓ 16-bit unsigned character.
- ✓ 16-bit signed character.
- ✓ 32-bit unsigned character.
- ✓ 32-bit signed character.
- ✓ 32-bit floating point.
- ✓ STRING: The string length when the data type is STRING.A maximum of 32 can be supported.

Decimal Digits: The number of decimal places of the variable when the data type is FLOAT. The maximum value is 6.

Unit: The variable unit.

String Filling Padding:Data collection length error, verification error, slave address error, collection timeout will be filled with string set as key value into the template.

We use Modbus slave software to imitate the serial terminal device. In the cloud, we can see the packaged data in JSON format.

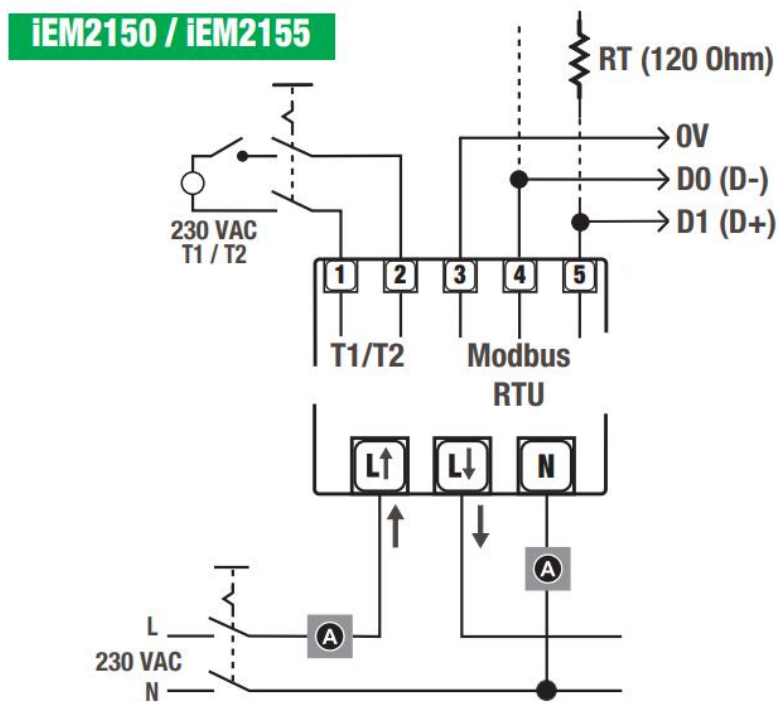
The screenshot displays two overlapping windows. The background window is the 'MQTT test client' interface within the AWS IoT console. It shows a sidebar with navigation options like Monitor, Activity, Onboard, Manage, Greengrass, Secure, Defend, Act, Test, Software, Settings, Learn, and Documentation. The main content area is titled 'MQTT test client info' and includes a 'Subscribe to a topic' button. Below this, a 'Topic filter' section shows 'awsiot/test' with a 'Subscribe' button. A 'Subscriptions' list on the right shows 'awsiot/test' with a heart icon and a JSON payload: { "voltage": "24.00V", "Current": "4.00A", "NAME": "bedroom", "DEVICE": "9CA525C4F2F0", "TIME": "2021-09-09 16:55:37" }. The foreground window is 'Modbus Slave' software. It has a 'Communication' tab with settings for Mode (TCP), Port (602), Port Name (COM0), Baud (115200), Data Bits (8 Bits), Stop Bits (1 Bit), and Parity (None). The 'Display Format' section shows Integer selected. The 'Slave ID' is 1 and 'Slave delay' is 100. A large table displays a grid of values from 0 to 99. At the bottom, a 'Communication Log' shows a series of hexadecimal data packets.

3. Connecting a energy meter to AWS IoT



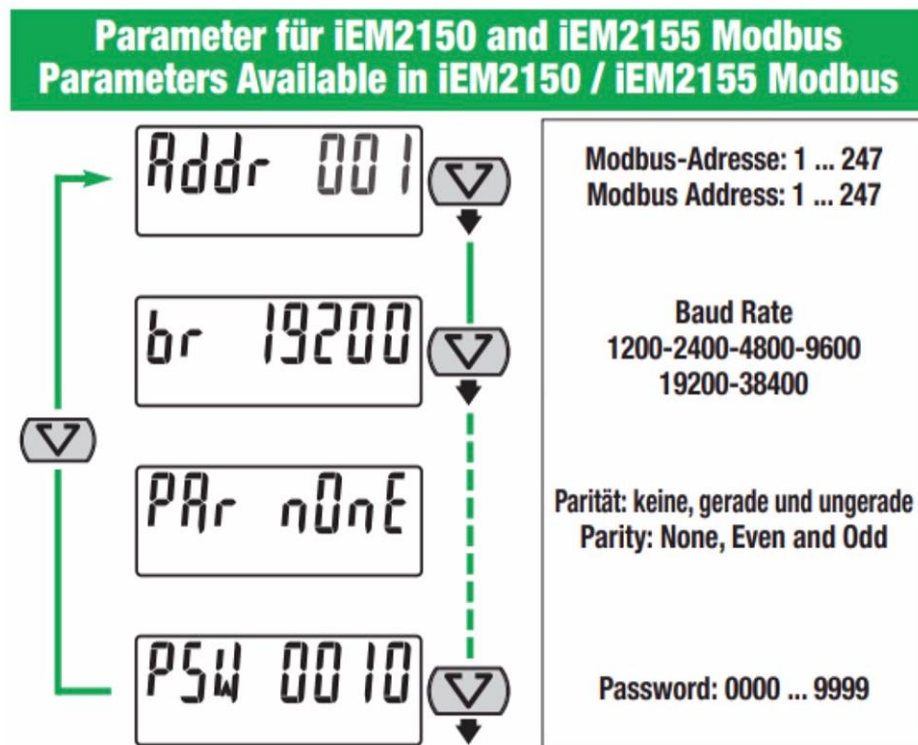
In this tutorial, we will set up a Energy Meter as a Modbus RTU slave and interface to our N510 gateway Modbus master.

The gateway is able to support up to 32 Modbus devices, including meters and remote I/O terminal. The standard industrial Modbus/RTU protocol runs on serial RS-485 interface, providing a robust & reliable interface.



Above shows the Serial RS-485 connection usually implemented using shielded twisted pair cable.

All slave devices connected on the RS-485 network must each have a unique address ranging from 1 to 247. Also make sure that the baudrate and parity for all slave devices and master must be the same. Use the front display/button to set the serial parameters.



Our gateway maps the required Modbus registers to be read during each polling cycle. This flexibility allows wide compatibility with most Modbus RTU devices from different manufacturers.

Meter Data

Current, voltage, power, power factor and frequency

Register Address	Action (R/W/WC)	Size	Type	Units	Description
Current					
3000	R	2	Float32	A	Current
Voltage					
3028	R	2	Float32	V	Voltage
Power					
3054	R	2	Float32	kW	Active Power
3068	R	2	Float32	kVAR	Reactive Power
3076	R	2	Float32	kVA	Apparent Power
Power Factor					
3084	R	2	Float32	—	Total Power Factor: -2 < PF < -1 = Quad 2, active power negative, capacitive -1 < PF < 0 = Quad 3, active power negative, inductive 0 < PF < 1 = Quad 1, active power positive, inductive 1 < PF < 2 = Quad 4, active power positive, capacitive
Frequency					
3110	R	2	Float32	Hz	Frequency

we will be referencing the Modbus register table from the product user manual (see sample above).we

are configuring to read

- Current (A) register from meter Modbus register address , 3000 (Schneider mandates a -1 offset) and name Key-Value Pairs, as "Current":"data2".
- Voltage (V) register from meter Modbus register address , 3028 (Schneider mandates a -1 offset) and name Key-Value Pairs, as "Voltage":"data1".
- Active Power(V) register from meter Modbus register address , 3054 (Schneider mandates a -1 offset) and name Key-Value Pairs, as "ActivePower":"data3".

The gateway is able to apply a customer formula to match the requirements of their cloud service.

We will now configure the gateway's serial port to operate as required for the attached Modbus devices. Go to the Port1 Settings tab and enter the following settings.

Version:V1.0.4-S Type:H7 中文

USR IOT
-IOT Experts-

Be Honest, Do Best!

Serial Port Settings

Baud Rate: 19200 bps(600~230400)

Data Bits: 8 bit

Parity: None

Stop Bits: 1 bit

Serial Port Selecting: RS485

Current Serial Port Type: RS232

Flow control: NONE

Packaging Time: 0 (0~255)ms

Packaging Length: 0 (0~1460)chars

RFC2217: ☒

Serial Heartbeat Packet: ☐

Socket A Parameters

Operating Mode: TCP Server

Max. No. of Client Connecting: 8 After Exceeding the Max. No. KICK

Local Port: 23 (1~65535)

Network Print Driver: ☐

Modbus Poll: ☐ Timeout : 200 (10~9999)ms

Modbus Abnormal response: ☐

Network Heartbeat Packet: ☐

Registry Packet format: None when: Connection String

Tips

- COM Port Settings:** Baud Rate, Data Bit, Parity, Stop Bit, Serial Operating Mode.
- IP Settings:** TCP Server, TCP Client, UDP Server, UDP Client, HTTP Client.
- Other Options:** Modbus TCP, Heartbeat Packet, RFC2217.

Then, go to the Packet Forwarding menu to enable Json and do the polling configuration as required by the user. Please refer to previous sections for how to connect N510 to AWT IOT service and data collecting.

Version: V1.0.4-S Type: H7 中文

USR IOT
-IOT Experts-

Be Honest, Do Best!

Overview	Basic Settings	Topic Publish	Topic Subscribe	Tips
Network Settings	MQTT: ☐ Enable			Keep Alive interval: Time interval for sending PINGREQ message. User Credentials: Contains the Username, password, Client ID, and other information required for MQTT connection. Clean session: Whether the broker continues to store/maintain the client's subscription status after the client disconnects. Enable Last-Will: Closed by default, contains the Last-will topic, Last-will message, Last-will QoS, Last-will Retain
Port1	MQTT Version: MQTT-3.1.1			
MQTT	Broker Address: awwis0u7xuagf.ats.iot.cn- [N/A]			
Packet Forwarding	Local/Remote Port NO.: 0 8883 (1~65535)			
Websocket to Serial	Keep Alive Interval: 60 (1~65535)s			
Upload Certificates	Connection Timeout: 0 (0~99999)s			
Miscellaneous	User Credentials: ☒			
System	User Name: admin			
	Password: admin			
	Client ID: 123456			
	Clean Session: ☒			
	Last-Will: ☐			
	Encryption mode: ssl1.2			
	verification method: Verify all			
	Save Cancel			

Version: V1.0.4-S Type: H7 中文

USR IOT
-IOT Experts-

Be Honest, Do Best!

Overview	Node(1-5)	Node(6-14)	Node(15-23)	Node(24-32)	Tips
Network Settings	Port1: JSON: ☐ Enable				JSON: Settings for Serial Message Format /RTU JSON Message Definition Node(1-5..) Operating mode: MQTT(default);Socket JSON template: The template should conform to the JSON format and contain key-value pairs of all nodes Device ID Contains the MAC, SN, and custom ID of the device. Timestamp Contains the Standard timestamps and UNIX timestamps Node N Configure the Modbus command
Port1	Add data header: ☐ Disable				
MQTT	Operating mode: MQTT				
Packet Forwarding	Topic: Topic1				
Websocket to Serial	Reporting method: Periodic				
Upload Certificates	Reporting interval: 10 s				
Miscellaneous	Fixed Attribute:				
System	Device ID: ☒	Timestamp: ☒			
	Key-Value Pairs: "DEVID":"MAC"	Key-Value Pairs: "TIME":"NTPTim"			
	ID: MAC	Timestamp Type: GMT			
	Quotation Mark: ☒	Quotation Mark: ☒			
	NODE1: ☒				
	Key-Value Pairs: "Voltage":"data1"	Poll interval: 500 (10~99999ms)			
	Slave ID: 1 (0~255)	Response Timeout: 2000 (10~99999ms)			
	Register Address: 2999 (0~65535)	Unit: ☒			
	Function Code: 3	Unit content: V			
	Data format: 32-bit float(A)	String Filling Padding: ☒			
	Formula: %s	String Filling: FF			
	Decimal Digits 0-6: 2	Quotation Mark: ☒			

Copyright © Jinan USR IOT Technology Limited. All Rights Reserved website: www.pusr.com

Version: V1.0.4-S Type: H7 中文

USR IOT
-IoT Experts-

Be Honest, Do Best!

Overview

Network Settings

Port1

MQTT

Packet Forwarding

Websocket to Serial

Upload Certificates

Miscellaneous

System

Decimal Digits 0-6: 2 Quotation Mark: ☒

NODE2: ☒

Key-Value Pairs: "Current"."data2" Poll interval: 500 (10~99999ms)

Slave ID: 1 (0~255) Response Timeout: 2000 (10~99999ms)

Register Address: 3027 (0~65535) Unit: ☒

Function Code: 3 Unit content: A

Data format: 32-bit float(A) String Filling Padding: ☒

Formula: %s String Filling: FF

Decimal Digits 0-6: 2 Quotation Mark: ☒

NODE3: ☒

Key-Value Pairs: "ActivePower"."d" Poll interval: 500 (10~99999ms)

Slave ID: 1 (0~255) Response Timeout: 2000 (10~99999ms)

Register Address: 3053 (0~65535) Unit: ☒

Function Code: 3 Unit content: kW

Data format: 32-bit float(A) String Filling Padding: ☒

Formula: %s String Filling: FF

Decimal Digits 0-6: 2 Quotation Mark: ☒

NODE4: ☐

Save Cancel

Copyright © Jinan USR IOT Technology Limited. All Rights Reserved website: www.pusr.com

Node 2: Configure the Modbus command and key-value pair parameters of the corresponding node

Function Code: Support 1, 2, 3, 4 function codes, formula calculations can be done on the data collected by 3, 4 function codes.

After these settings are done, restart the gateway. Now that the connectivity between the Schneider meter and the N510 gateway has been set, we will then look at getting the data onto a designated cloud service. Next go back to go your AWS IoT console and subscribe to the Topic to "Test" that data is being received.

Amazon IoT

Monitor

Activity

Onboard

Manage

Greengrass

Secure

Defend

Act

Test

Software

Settings

Learn

Documentation

New console experience

AWS IoT > MQTT test client

MQTT test client

You can use the MQTT test client to monitor the MQTT messages being passed in your Amazon Web Services account. Devices publish MQTT messages that are identified by topics to communicate their state to Amazon IoT. Amazon IoT also publishes MQTT messages to inform devices and apps of changes and events. You can subscribe to MQTT message topics and publish MQTT messages to topics by using the MQTT test client.

Subscribe to a topic Publish to a topic

Topic filter Info

The topic filter describes the topic(s) to which you want to subscribe. The topic filter can include MQTT wildcard characters.

awsiot/test

Additional configuration

Subscribe

Subscriptions

awsiot/test

Pause Clear Export Edit

awsiot/test

September 15, 2021, 15:26:48 (UTC+0800)

```
{
  "Voltage": "110.90V",
  "Current": "8.68A",
  "ActivePower": "953.73kW",
  "Location": "bedroom",
  "DEVID": "9C4525C4F2F9",
  "TIME": "2021-09-15 15:26:48"
}
```

English Privacy Policy Terms of Use

4. Factory reset

There are two ways to restore the N510 to factory settings: hardware factory reset and software factory

reset.

➤ Hardware factory reset

Step 1: Find the RESET button

Step 2: After the device is powered on, hold down the RESET button 3-15s and release the RESET button

➤ Software factory reset

Choose System, click the Restore Factory Defaults and select OK. N510 will complete the factory reset operation by itself.

