

NPort IA5000A-I/O or NPort IAW5000A-I/O to AWS via MQTT Broker

Moxa Technical Support Team
support@moxa.com

Contents

- 1. Introduction..... 2**
- 2. System Topology..... 2**
 - 2.1 Create AWS IoT and Thing 3
 - 2.2 Set Up the NPort IA5000A-I/O and NPort IAW5000A-I/O to Connect to AWS IoT Via MQTT Broker 12
 - 2.3 Set Up MQTT.fx(client) Connect to AWS IoT..... 15
- 3. Upload/Download Serial Patterns and I/O Status With the Cloud 18**
 - 3.1 Sending Serial Patterns From the Device to the Cloud..... 20
 - 3.2 Sending Serial Data From the Cloud to the Device..... 24
 - 3.3 Sending the NPorts’ DI and DO Status to the Cloud 27
 - 3.4 Control the NPort’s DO Status Through the Cloud..... 30

Copyright © 2019 Moxa Inc.

Released on June 06, 2019

About Moxa

Moxa is a leading provider of edge connectivity, industrial networking, and network infrastructure solutions for enabling connectivity for the Industrial Internet of Things. With over 30 years of industry experience, Moxa has connected more than 50 million devices worldwide and has a distribution and service network that reaches customers in more than 70 countries. Moxa delivers lasting business value by empowering industry with reliable networks and sincere service for industrial communications infrastructures. Information about Moxa’s solutions is available at www.moxa.com.

How to Contact Moxa

Tel: +886-2-8919-1230
Fax: +886-2-8919-1231



1. Introduction

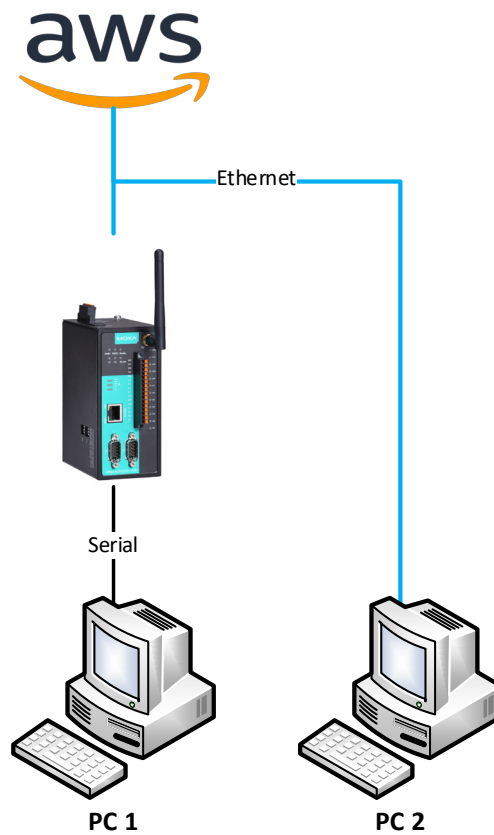
The NPort IA5000A-I/O and NPort IAW5000A-I/O serial device servers, which have built-in digital I/Os, provide maximum flexibility when you need to integrate serial equipment in the field with an Ethernet network or cloud platform. From Firmware Version 2.0 onwards, they support communications with IIoT applications, using generic MQTT or third-party cloud services, such as Azure and Alibaba Cloud.

This document demonstrates how to use the NPort IA5000A-I/O or NPort IAW5000A-I/O serial device connecting to AWS IoT. We also demonstrate how to publish serial or I/O data messages to AWS IoT and subscribe messages from AWS IoT.

2. System Topology

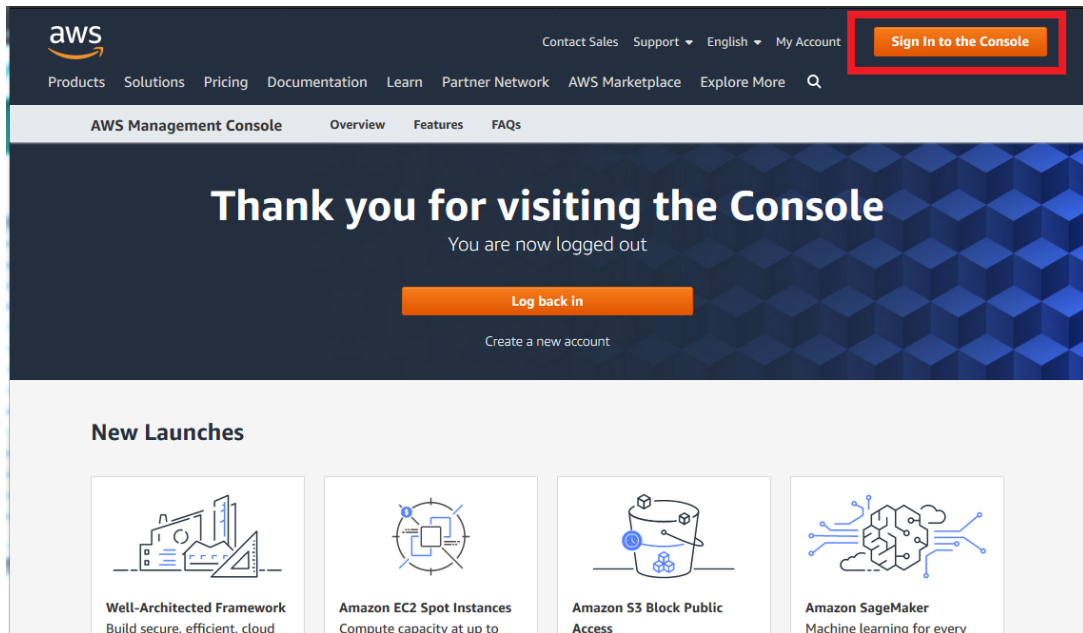
Figure 1 illustrates the system topology. PC1 runs PComm Lite to act as a serial device. It connects to Port 1 of the NPort IA5000A-I/O or NPort IAW5000A-I/O serial device. The NPort IA5000A-I/O or NPort IAW5000A-I/O serial device acts as a MQTT Broker and connects to AWS IoT. PC2 runs MQTT.fx MQTT Client. The MQTT.fx published messages to AWS IoT and subscribes topics from AWS IoT.

< Figure 1. System Topology >

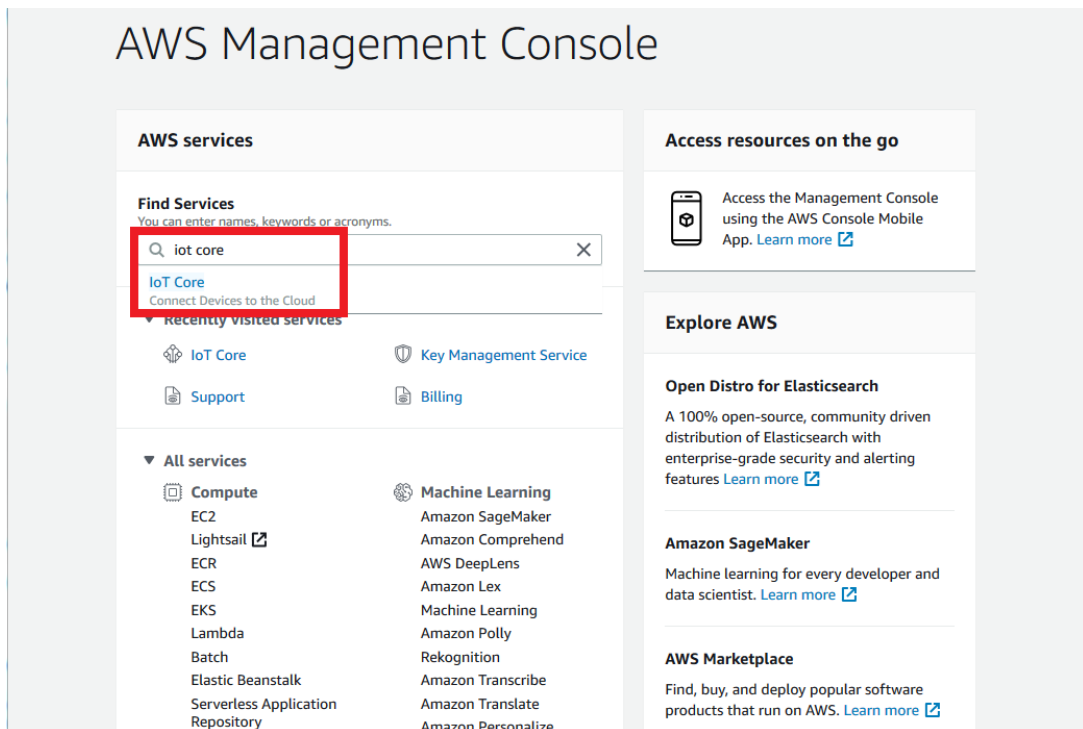


2.1 Create AWS IoT and Thing

1. Use AWS user account to log in to AWS Console.
Website: <https://aws.amazon.com/console/>

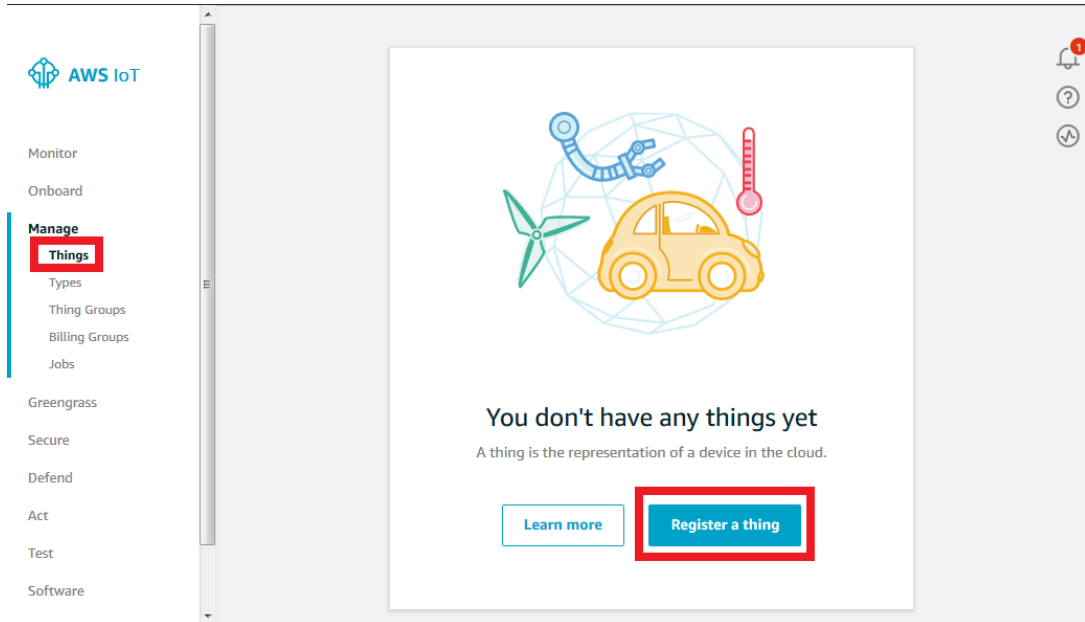


2. Fill in "iot core" key word under **Find Services**, or find "Internet Of Things → IoT Core" under the **All services** section:

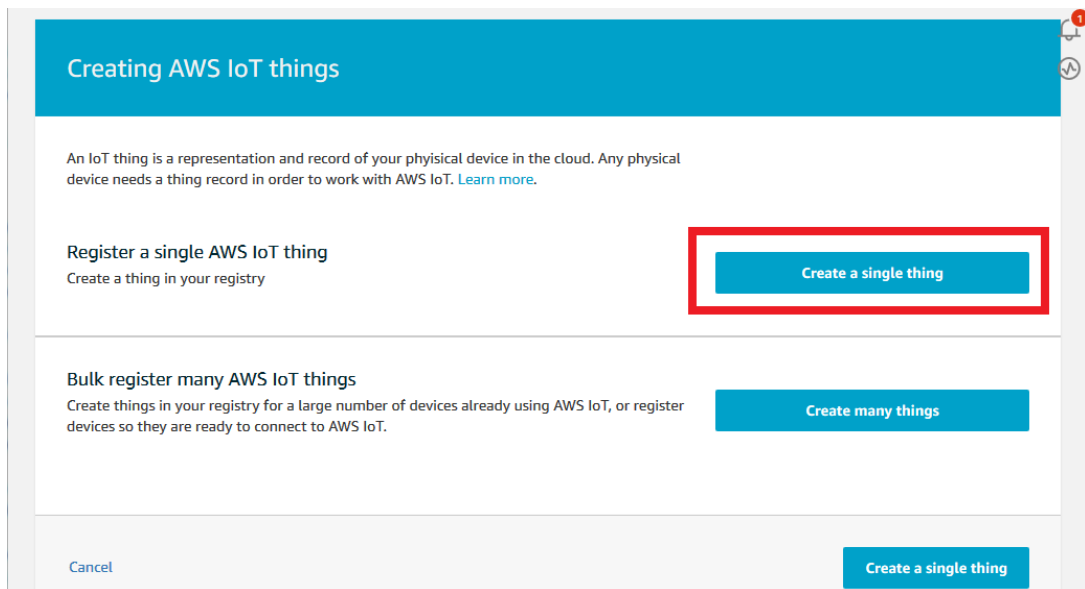


3. To register a thing:

On the menu, select **Manage → Things**, and click the **Register a thing** button.



On the **Creating AWS IoT things** page, click **Create a single thing**.



For the first step, on the **Add your device to the thing registry** page, fill in "NPort" under **Name** and click **Next**.

CREATE A THING

Add your device to the thing registry

STEP 1/5

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

Name

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.

For step 2, on the **Add a certificate for your thing** page, click **Create certificate**.

← CREATE A THING

Add a certificate for your thing

STEP 2/5

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

For the last step, on the **Certificate created!** page, download the files below:

- A certificate for this thing
- A public key
- A private key
- A root CA for AWS IoT

Then, click the **Activate** button, which will change to **Deactivate**; lastly, click **Attach a policy**.

Certificate created! Successfully deactivated certificate. x

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.

In order to connect a device, you need to download the following:

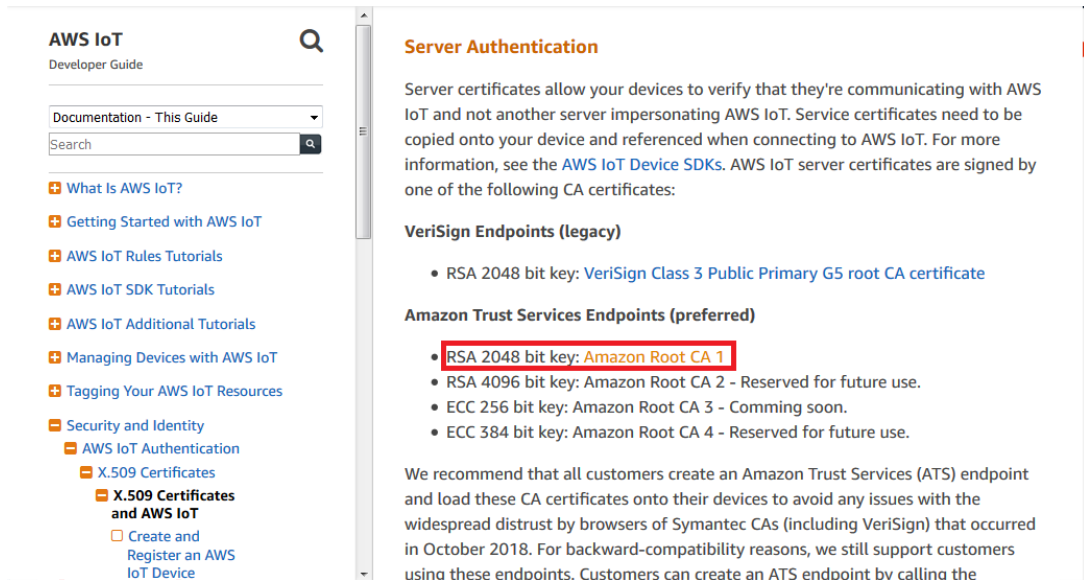
A certificate for this thing	XXXXXXXXXX.cert.pem	Download
A public key	XXXXXXXXXX.public.key	Download
A private key	XXXXXXXXXX.private.key	Download

You also need to download a root CA for AWS IoT:
A root CA for AWS IoT: Download

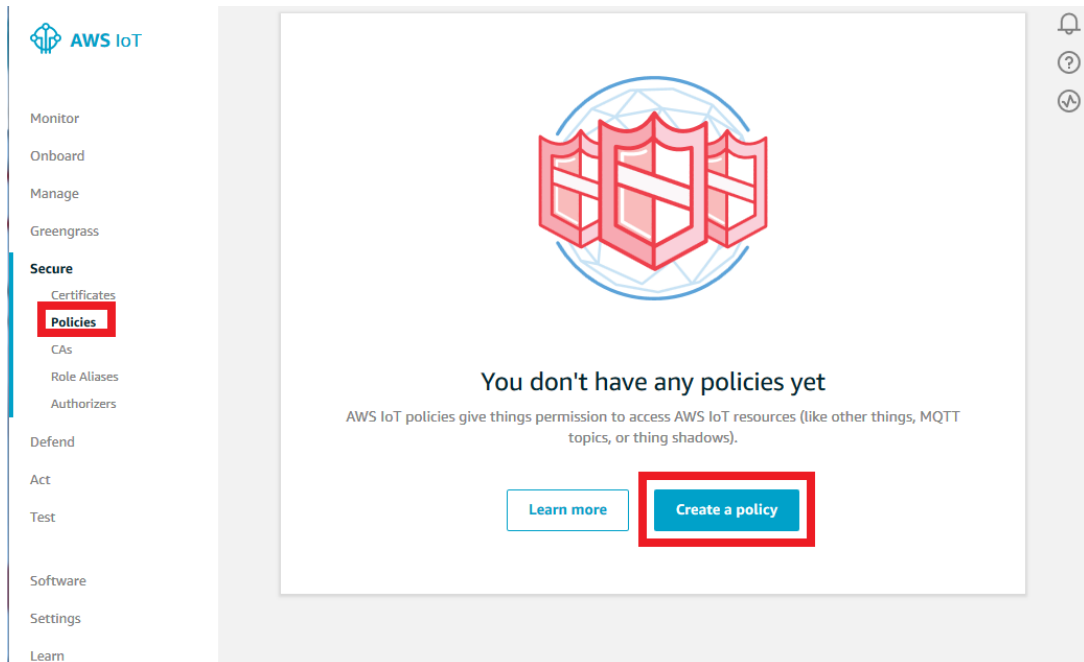
Activate

Cancel Done Attach a policy

When you download a root CA file, a new web page (as below) will pop up for you to download a root CA file. In this demonstration, we use Amazon Root CA 1 (RSA 2048 bit key).



4. To create a policy, select **Secure** → **Policies**, and click the **Create a policy** button.



On the **Create a policy** page, set the following settings as below and click the **Create** button:

- Name: MOXA_IoT_Policy
- Action: iot:*
- Resource ARN: *
- Effect: Allow

← 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
MOXA_IoT_Policy

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

Now, you have a new policy, namely MOXA_IoT_Policy.

AWS IoT

Monitor
Onboard
Manage
Greengrass
Secure
Certificates
Policies
CAs
Role Aliases
Authorizers

Policies

Successfully created a policy.

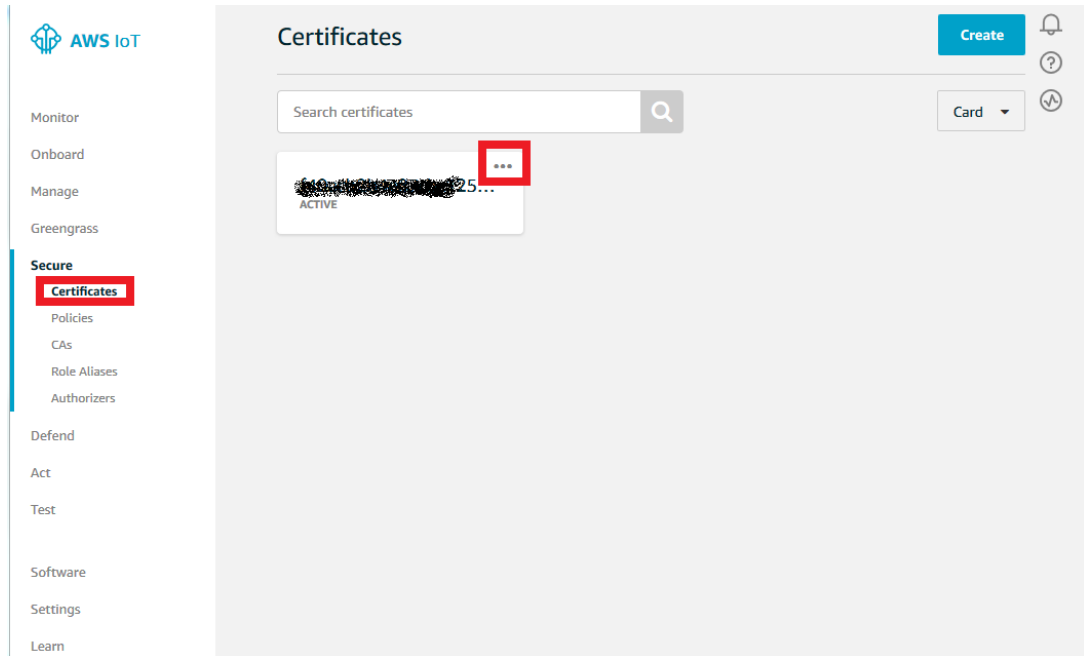
Search policies

MOXA_IoT_Policy

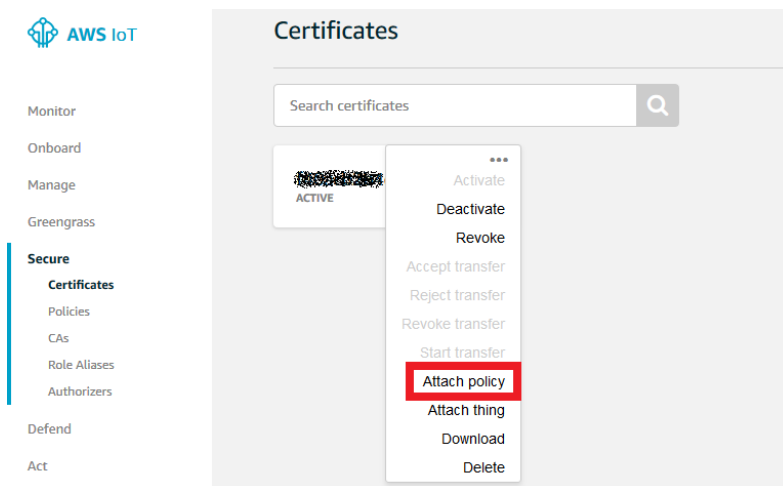
After clicking **OK**, it will show "Device Certificate". We can now click **Close** to close the window.

Next, in the Device List, the device name will appear as "NPort".

5. To attach the Device Certificate to an AWS IoT Policy and a thing:
Select **Secure** → **Certificates** and left-click the ellipsis (...).



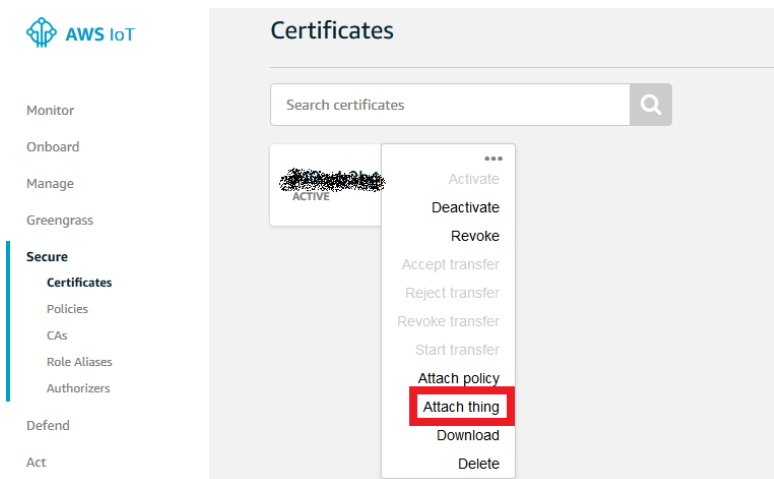
In the drop-down menu, click **Attach policy**.



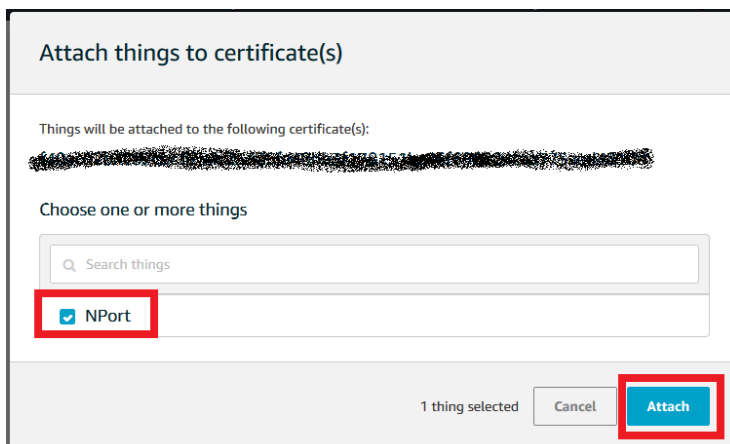
In the **Attach policies to certificate(s)** dialog box, check "MOXA_IoT_Policy", and then click **Attach**.



Go back to the drop-down menu and select **Attach thing**.



In the **Attach things to certificate(s)** dialog box, check "NPort" and click **Attach**.



2.2 Set Up the NPort IA5000A-I/O and NPort IAW5000A-I/O to Connect to AWS IoT Via MQTT Broker

1. Log in to the NPort IA5000A-I/O or NPort IAW5000A-I/O's web console and set the correct time setting on the **Basic Settings** page. Also, you can fill in **Time server** to correct the NPort's time-on period. You will find the NTP service at <https://www.pool.ntp.org/zone/>

Time Settings	
Time zone	(GMT+08:00)Taipei
Local time (24-hour)	2019 / 05 / 17 22 : 23 : 49
Time server	asia.pool.ntp.org

2. Click **Main Menu** → **IoT Management** → **IoT Mode** to set **IoT platform** as "MQTT Broker".

IoT Mode

Basic Settings	
IoT platform	MQTT Broker

The IoT Mode is running with MQTT Broker. It will show more settings regarding MQTT as below:

MQTT Connection Settings	
Host address	<input type="text"/>
Host port	1883
Username	<input type="text"/>
Password	<input type="text"/>
Client ID	<input type="text"/> <input type="button" value="Generate"/>
Keep alive	60 (1 - 65535 sec.)
Clean session	<input type="checkbox"/> enable
TLS (Transport Layer Security)	
TLS mode	Disable

- 3. Under the **MQTT Connection Settings** → **Host address** string, fill in the HTTPS link, and "8883" for **Host port**.

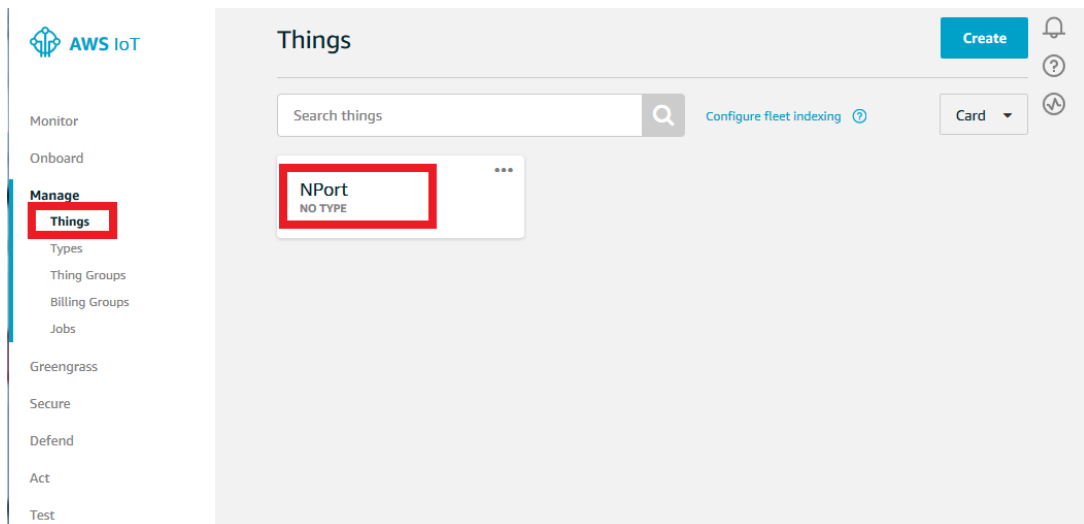
Client ID identifies the MQTT session; it must be unique. Broker doesn't accept the same **Client ID** connection twice. You can fill in an identifiable ID or click **Generate** to generate a random ID.

IoT Mode

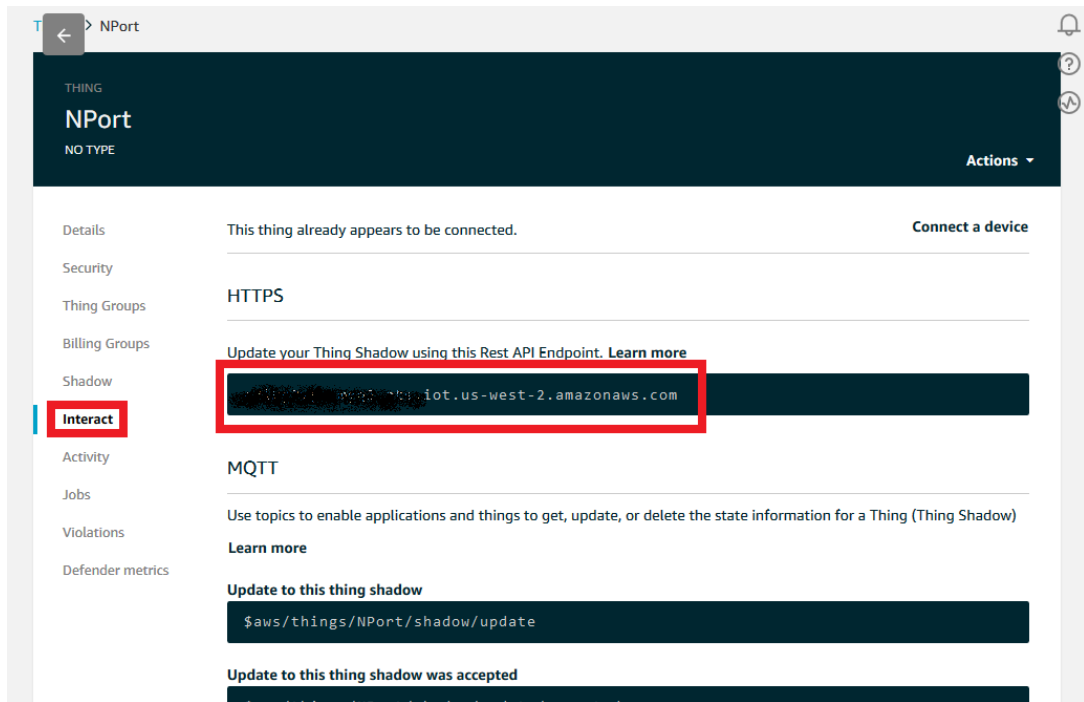
The screenshot shows the 'IoT Mode' configuration page. Under 'MQTT Connection Settings', the following fields are visible: 'IoT platform' is set to 'MQTT Broker'; 'Host address' is 'myq3-ats.iot.us-west-2.amazonaws.com'; 'Host port' is '8883'; 'Username' and 'Password' are empty; 'Client ID' is 'fad08b22-a1a4-48fd-8e1b-73d5406832!' with a 'Generate' button; 'Keep alive' is '60 (1 - 65535 sec.)'; and 'Clean session' is checked 'enable'.

You can follow the following steps to find the HTTPS link:

Go back to **IoT core** on the AWS Console. Select **Manage** → **Things** and left-click to show the details.



Under **Thing**, select **Interact** in order to reveal the HTTP link.



- To enable a TLS transmission, set **TLS mode** as "TLS v1.2". We need to upload CA certificate, client certificate, and client key file before you can download ("Register a Thing" step). The certificates and key file must be PEM encoded. If your key file has a passphrase, fill in the correct passphrase when uploading a key file as below:

TLS (Transport Layer Security)

TLS mode	TLS v1.2		
CA file	AmazonRootCA1.pem	Browse...	No file selected. Upload Delete
Client certificate file	XXXXXXXXXX-certificate.pem.crt	Browse...	No file selected. Upload Delete
Client key file	XXXXXXXXXX-private.pem.key	Browse...	No file selected. Upload Delete
Client key password			

5. Uncheck **Retain** under **MQTT Publish** so that AWS IoT does not support retain messages. For more detailed information, please reference AWS Documentation » AWS IoT » Developer Guide » Message Broker for AWS IoT » Protocols » MQTT (<https://docs.aws.amazon.com/iot/latest/developerguide/mqtt.html>)

MQTT Publish			
Serial port 1	Topic: NPortIO/JSON/Sport1/Pub/Data	QoS: 1	Retain <input type="checkbox"/>
Serial port 2	Topic: NPortIO/JSON/Sport2/Pub/Data	QoS: 1	Retain <input type="checkbox"/>
I/O	Topic: NPortIO/JSON/DIO/Pub	QoS: 1	Retain <input type="checkbox"/>
MQTT Subscribe			
Serial port 1	Topic: NPortIO/JSON/Sport1/Sub/Data	QoS: 1	
Serial port 2	Topic: NPortIO/JSON/Sport2/Sub/Data	QoS: 1	
I/O	Topic: NPortIO/JSON/DIO/Sub	QoS: 1	

After clicking the **Submit** button, the NPort IA5000A-I/O or NPort IAW5000A-I/O will connect to AWS IoT, and you can check whether connection status states **Connected** under **IoT Connection Monitoring** as below:

IoT Connection Monitoring

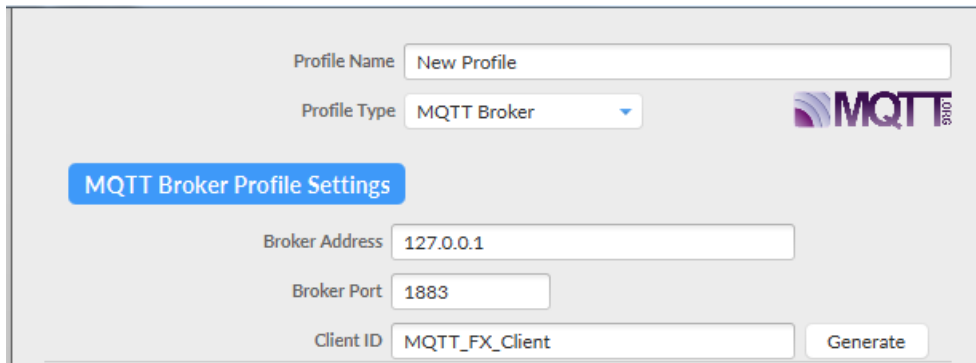
MQTT Client Connection Information	
Target	██████████.iot.us-west-2.amazonaws.com
Connection status	Connected
Diagnostics log	2019/05/24 21:18:58 Connecting... 2019/05/24 21:18:59 Connected successfully!

2.3 Set Up MQTT.fx(client) Connect to AWS IoT

MQTT.fx is a MQTT Client written in Java based on Eclipse Paho. It is published under Apache License, Version 2.0.

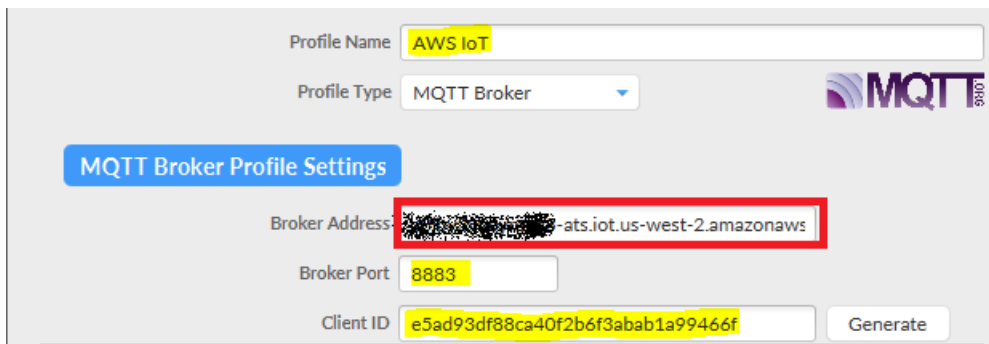
1. Download MQTT.fx and install it on **PC 2**, which can be download from <https://mqttfx.jensd.de/>
2. To launch MQTT.fx and configuration profile with MQTT Broker default settings, click the "gear" icon, or on the toolbar, select **Extras** → **Edit Connection Profile** to modify the profile settings.

The **Edit Connection Profile** window will pop up.



The screenshot shows the 'MQTT Broker Profile Settings' window. It features a 'Profile Name' field with 'New Profile', a 'Profile Type' dropdown menu set to 'MQTT Broker', and an MQTT logo. Below these is a blue button labeled 'MQTT Broker Profile Settings'. The 'Broker Address' field contains '127.0.0.1', the 'Broker Port' field contains '1883', and the 'Client ID' field contains 'MQTT_FX_Client'. A 'Generate' button is located at the bottom right.

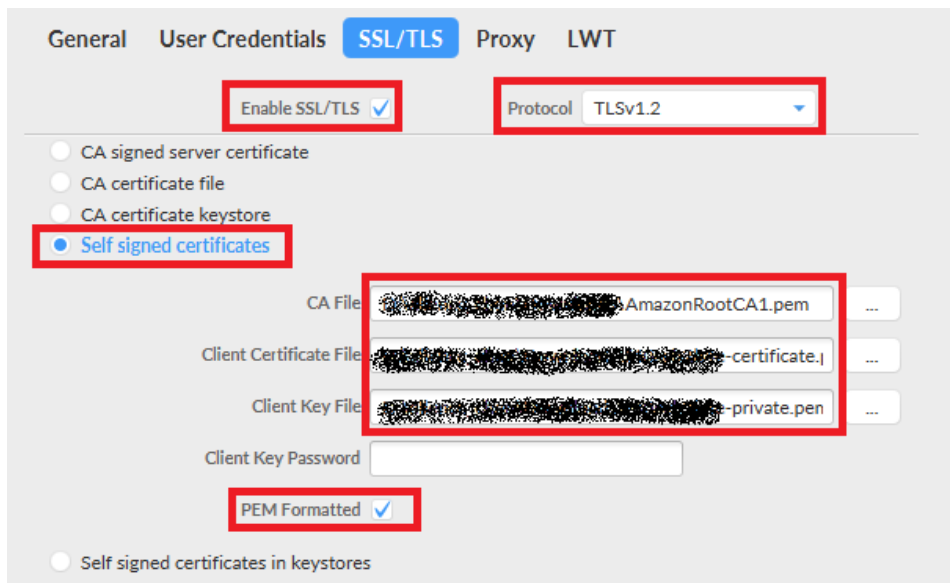
To configure the AWS IoT Profile Settings session, set "AWS IoT" as the **Profile Name** and "MQTT Broker" as the **Profile Type**. For **Broker Address**, fill in the HTTPS link registered for "thing", and for **Broker Port** fill in "8883". The **Client ID** identifies the MQTT session; it must be unique. Broker doesn't accept the same **Client ID** connection twice. You can fill in an identifiable ID or click **Generate** to generate a random ID.



The screenshot shows the 'MQTT Broker Profile Settings' window configured for AWS IoT. The 'Profile Name' field is highlighted in yellow and contains 'AWS IoT'. The 'Profile Type' dropdown menu is set to 'MQTT Broker'. The 'MQTT Broker Profile Settings' button is highlighted in blue. The 'Broker Address' field is highlighted with a red box and contains a redacted address followed by '-ats.iot.us-west-2.amazonaws.com'. The 'Broker Port' field is highlighted in yellow and contains '8883'. The 'Client ID' field is highlighted in yellow and contains a long alphanumeric string. A 'Generate' button is located at the bottom right.

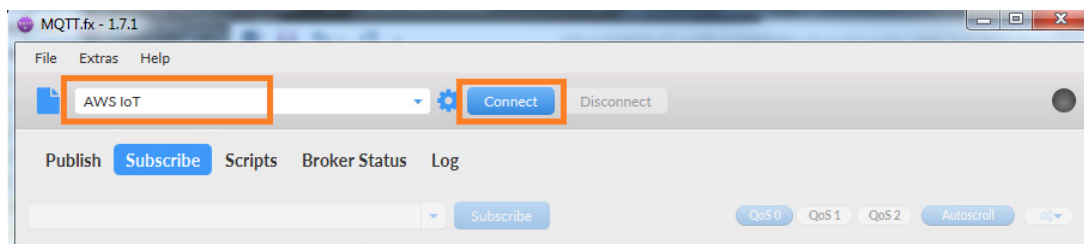
Select the **SSL/TLS** tab and set the settings as below:

- Enable SSL/TLS: Check
- Protocol: TLSv1.2
- Select **Self signed certificates** to upload the CA certificate, client certificate, and client key file before you can download ("Register a thing" step). If your key file has a passphrase, fill in the correct passphrase.
- PEM Formatted: Check

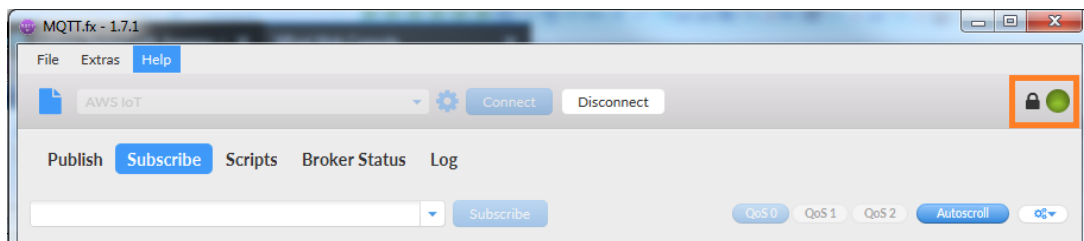


Click **OK** to confirm and close the window.

3. Select **AWS IoT** for the profile name and then click the **Connect** button.



After clicking **Connect**, the lamp icon will change to green if the connection has been successfully established.



3. Upload/Download Serial Patterns and I/O Status With the Cloud

In this section, we will instruct you on how the NPort IA5000A-I/O and NPort IAW5000A-I/O (in the following chapters, referenced as NPort or NPorts) send serial patterns to the cloud and receive patterns from the cloud. If a DI is triggered, the NPorts will publish the I/O status to the cloud, and receive a message from the cloud to the NPorts to change the I/O status. The NPorts support three types of message formats: JSON, RAW with header, and RAW. In this demonstration, we use the JSON format. For **Message format**, we select JSON, and for **I/O publish trigger mode**, we select "Specific I/O change" along with "DI-00".

Serial and I/O Message Format Settings

Message format JSON Raw

Serial and I/O JSON message definition

I/O publish trigger mode

MQTT Publish

Serial port 1	Topic	NPortIO/JSON/SPort1/Pub/Data	QoS	1	Retain	<input type="checkbox"/>
Serial port 2	Topic	NPortIO/JSON/SPort2/Pub/Data	QoS	1	Retain	<input type="checkbox"/>
I/O	Topic	NPortIO/JSON/DIO/Pub	QoS	1	Retain	<input type="checkbox"/>

MQTT Subscribe

Serial port 1	Topic	NPortIO/JSON/SPort1/Sub/Data	QoS	1	
Serial port 2	Topic	NPortIO/JSON/SPort2/Sub/Data	QoS	1	
I/O	Topic	NPortIO/JSON/DIO/Sub	QoS	1	

For the purpose of this demonstration, we will show you the text content of the data we upload to the cloud platform. Click the **Serial JSON** button to uncheck the **enable Base64 Encode/Decode for serial data** checkbox. JSON format does not support any special characters. If needed, set **Encode/Decode for serial data**. For more about JSON format rules, please reference <http://json.org/>

Serial JSON Message Definition

Publish JSON Message

```

{
  "msgVer"      :      "1.0",
  "gwID"       :      "NPortIAW5250A-12I/O_2647",
  "devID"      :      "SerialPort1",
  "dateTime"   :      "2018-08-27T15:43:14+08:00",
  "msgNumber"  :      0-65535,
  "msgType"    :      "Data",
  "msgFrame"   :      "Raw data from serial port"
}

```

port 1 port 2 (devID is referred to Alias in Ser
 enable
 enable
 enable Base64 Encode/Decode for serial data

Note: You must fill in serial **alias name**, which is an identifiable ID for serial data on the **Serial Parameter** page.

Serial Parameter

* Modifying "Serial Parameter" settings will cause the serial ports to restart connections.

Port	Alias	Alias code	Baud rate	Parity	Data bit	Stop bit	Flow control	FIFO	Interface
1	SerialPort1	p1	115200	None	8	1	RTS/CTS	Enable	RS-232
2	SerialPort2	p2	115200	None	8	1	RTS/CTS	Enable	RS-232

In this demonstration, we use the NPort's DO-00 to trigger DI-00 (connect DO-00 to DI-00 by wire).



3.1 Sending Serial Patterns From the Device to the Cloud

In this section, we will instruct you on how to send serial data to the cloud. First, use MQTT.fx to subscribe to the topic of Serial Port 1 of the NPort from the cloud; second, send a serial pattern from PC 1 through the NPort to the cloud, and MQTT.fx can receive a message from the NPort.

1. Log in to the NPorts' web console and change Serial Port 1's **Operation mode** to "IoT" and **Force transmit** to "500". The NPorts support several **Data Packing** combinations. In this demonstration, we use Force transmit. If needed, set the correct Data Packing method.

Operation Mode

Port Settings

Port: 1

Operation mode: IoT

Sniffer mode: Enable (Subscribed messages will be dropped)

Data Packing

Packet length: (0 - 2880)

Delimiter 1: (HEX) Enable

Delimiter 2: (HEX) Enable

Delimiter process: Do Nothing (Processed only when the packet length is 0)

Force transmit: 500 (0 - 65535 ms)

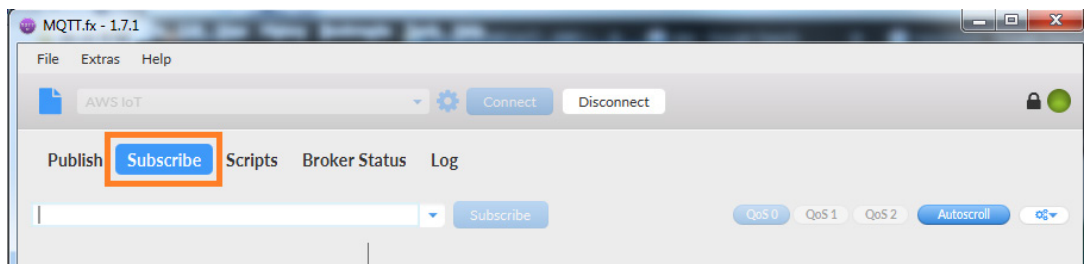
Apply the above settings to all serial ports

Click **Submit** to activate configuration.

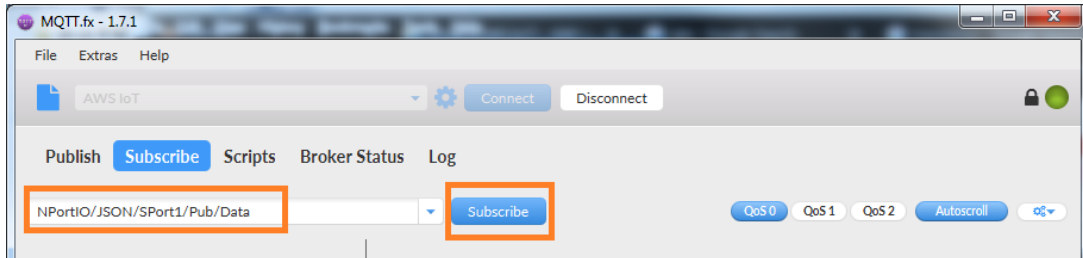
Operation Mode

Port	Operating mode	Packet length	Delimiter 1	Delimiter 2	Delimiter process	Force transmit
1	IoT	0	00 (Disable)	00 (Disable)	Do Nothing	500
2	Real COM	0	00 (Disable)	00 (Disable)	Do Nothing	0
Max connection:			1			

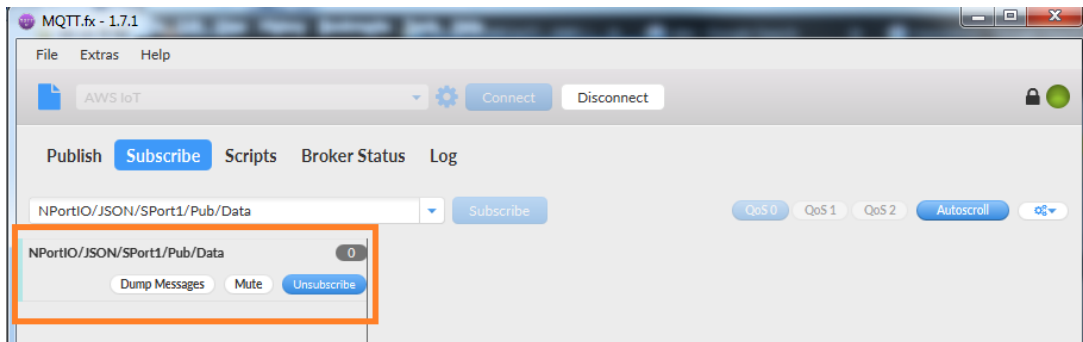
2. In the **MQTT.fx** window, click the **Subscribe** tab.



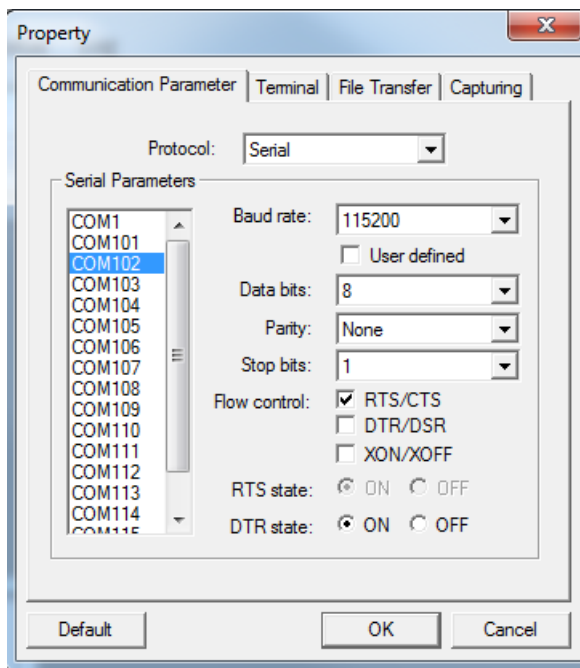
3. To subscribe a topic from the NPort's Serial Port 1, click the **Subscribe** tap, fill in the topic string as "NPortIO/JSON/SPort1/Pub/Data" in the drop-down field, and click the **Subscribe** button to the right.



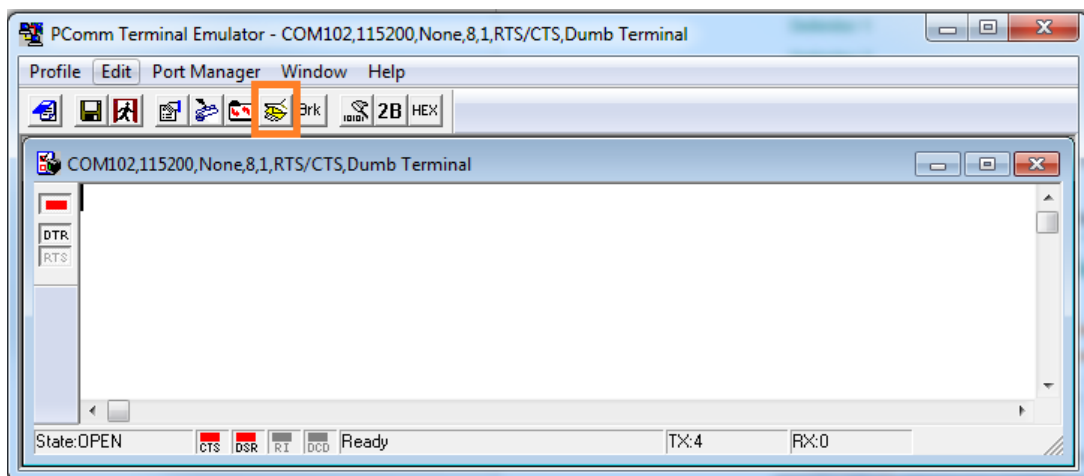
Registered topics are listed in fill-in box to the left of the Subscribe tab and can be unsubscribed by clicking **Unsubscribe**.



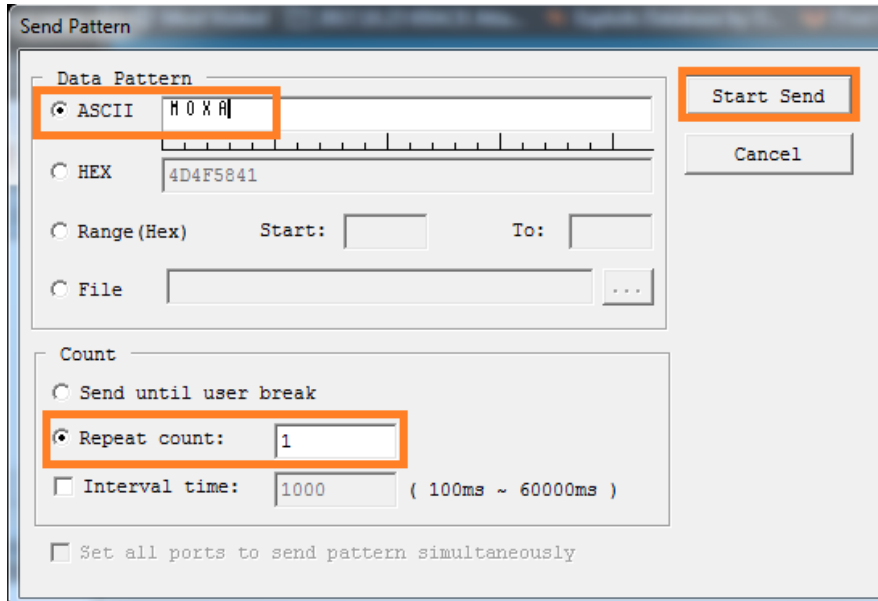
4. Launch **PComm Terminal Emulator** on PC 1, and open COM Port with the NPort's serial default settings as below:
 - Port number: PC 1's native COM port connecting to the NPort's Port 1
 - Baud rate: 115200
 - Data bits: 8
 - Parity: None
 - Stop bits: 1
 - Flow control: RTS/CTS



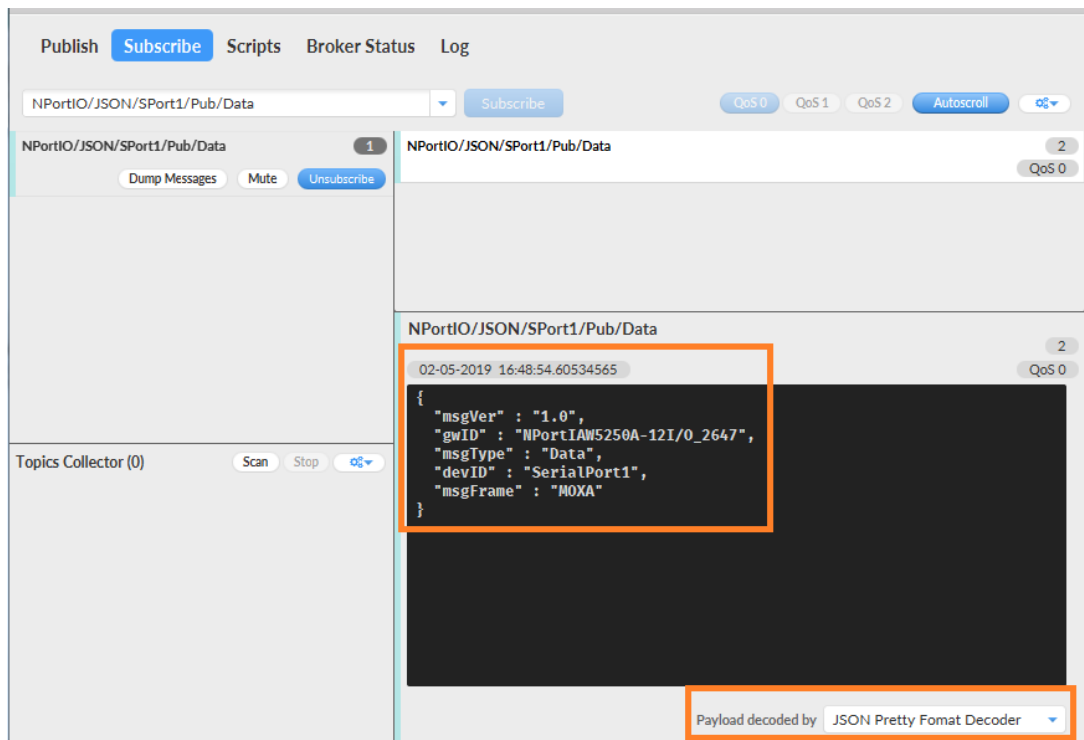
5. Click the **Send Pattern** button, or on the toolbar select **Port Manager** → **Send Pattern** to send a serial pattern.



In the Send Pattern window, select **ASCII** and fill in "MOXA"; then, select **Repeat count** and enter "1". Click **Start Send** to send the pattern.



6. On the **MQTT.fx** page, you will receive a message from the cloud that was sent from the NPorts. For **Payload decoded by** select "JSON Pretty Format Decoder" to show the message.



The serial data pattern will be filled in the msgFrame.

```
NPortIO/JSON/SPort1/Pub/Data  
02-05-2019 16:48:54.60534565  
{  
  "msgVer" : "1.0",  
  "gwID" : "NPortIAW5250A-12I/O_2647",  
  "msgType" : "Data",  
  "devID" : "SerialPort1",  
  "msgFrame" : "MOXA"  
}
```

3.2 Sending Serial Data From the Cloud to the Device

In this section, we will instruct you on how to send serial data to PC 1 from the cloud. We use MQTT.fx to publish the topic of Serial Port 1 of the NPort to the cloud. We will receive a serial pattern from PC 1 through the cloud to the NPort.

1. Click **Serial JSON**.

IoT Mode

Basic Settings

IoT platform: MQTT Broker

MQTT Connection Settings

Host address: [redacted]-ats.iot.us-west-2.amaz

Host port: 8883

Username: [input field]

Password: [input field]

Client ID: 1136d71d-ead0-4dde-ae64-bb8ef9986e

Keep alive: 60 (1 - 65535 sec.)

Clean session: enable

TLS (Transport Layer Security)

TLS mode: TLS v1.2

CA file: AmazonRootCA1.pem No file selected.

Client certificate file: [redacted]certificate.pem.crt1 No file selected.

Client key file: [redacted]-private.pem.key No file selected.

Client key password: [input field]

MQTT Will Message

Enable Will message: enable

Serial and I/O Message Format Settings

Message format: JSON Raw

Serial and I/O JSON message definition: I/O JSON

I/O publish trigger mode: Specific I/O change DI-00

Copy **Subscribe JSON Message**:

Serial JSON Message Definition

Publish JSON Message

```
{  
  "msgVer"      :      "1.0",  
  "gwID"        :      "NPortIAW5250A-12I/O_2647",  
  "devID"       :      "SerialPort1",  
  "dateTime"    :      "2018-08-27T15:43:14+08:00",  
  "msgNumber"   :      0-65535,  
  "msgType"     :      "Data",  
  "msgFrame"    :      "Raw data from serial port"  
}
```

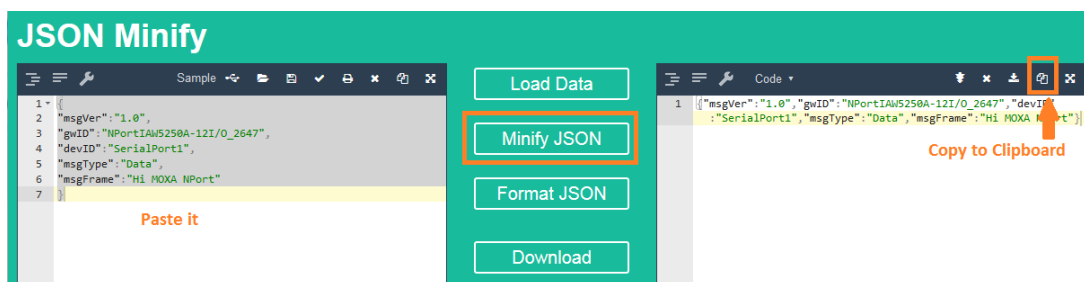
Subscribe JSON Message

```
{  
  "msgVer"      :      "1.0",  
  "gwID"        :      "NPortIAW5250A-12I/O_2647",  
  "devID"       :      "SerialPort1",  
  "msgType"     :      "Data",  
  "msgFrame"    :      "Raw data to serial port"  
}
```

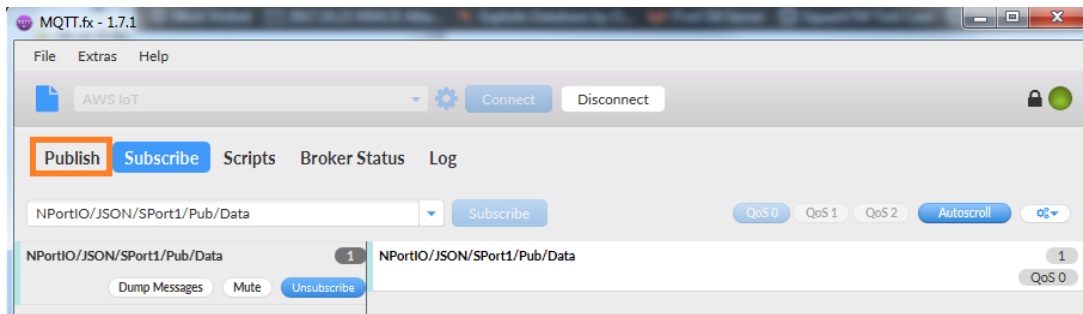
2. The copied message has a lot of space and line feed. We can use a tool to compact it. Below is a free online tool:

<https://jsonformatter.org/json-minify>

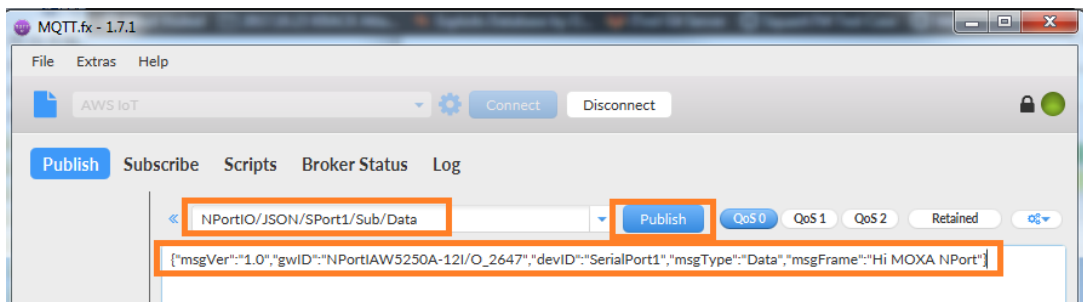
Paste the message in the column on the left and change the msgFrame stating: "Raw data to serial port" to read: "Hi MOXA NPort"; then, click **Minify JSON**. It will show a compact JSON format message in the column on the right. **Click Copy to Clipboard**.



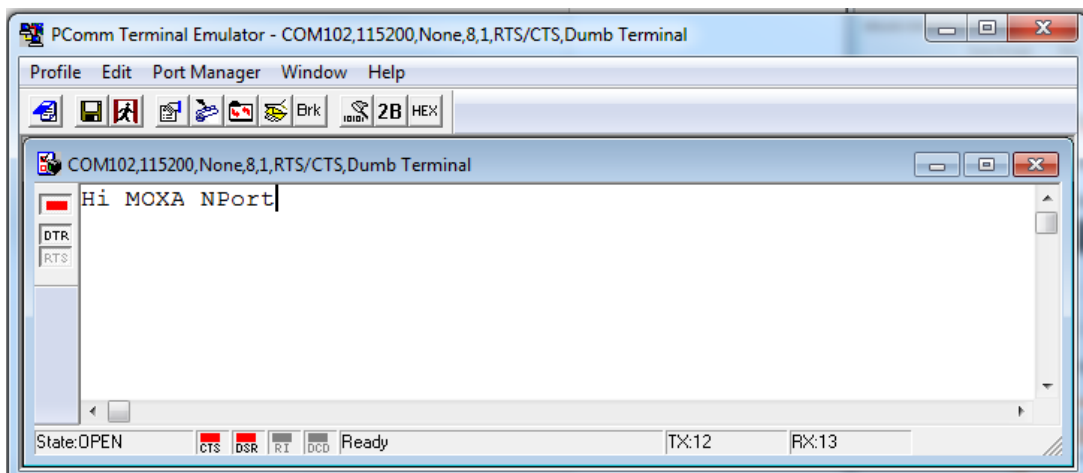
3. On the **MQTT.fx** page, click the **Publish** tab.



4. To publish a topic to the NPort's Serial Port 1, under the **Publish** tab, paste the clipboard message in the big textbox, fill in the topic string as "NPortIO/JSON/SPort1/Sub/Data" in the drop-down field, and click the **Publish** button.



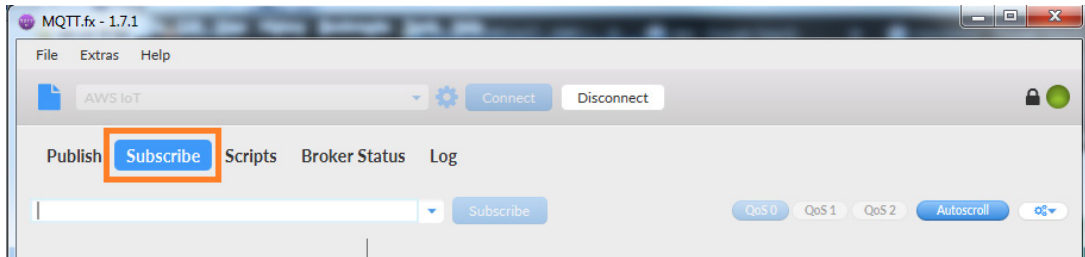
In the **PComm Terminal Emulator** window, you will receive a message from the cloud that was sent from MQTT.fx.



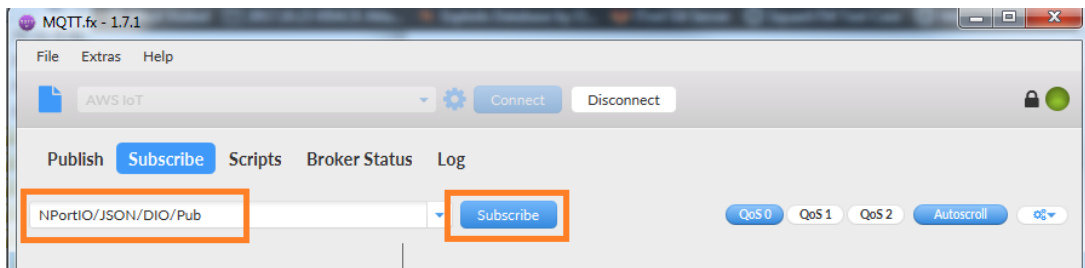
3.3 Sending the NPorts' DI and DO Status to the Cloud

In this section, we will instruct you on how to trigger the DI status to the cloud. First, we use MQTT.fx to subscribe the NPort's I/O topic; then, trigger the DI status to change; lastly, you will receive a message from the cloud regarding the NPort's DI and DO status.

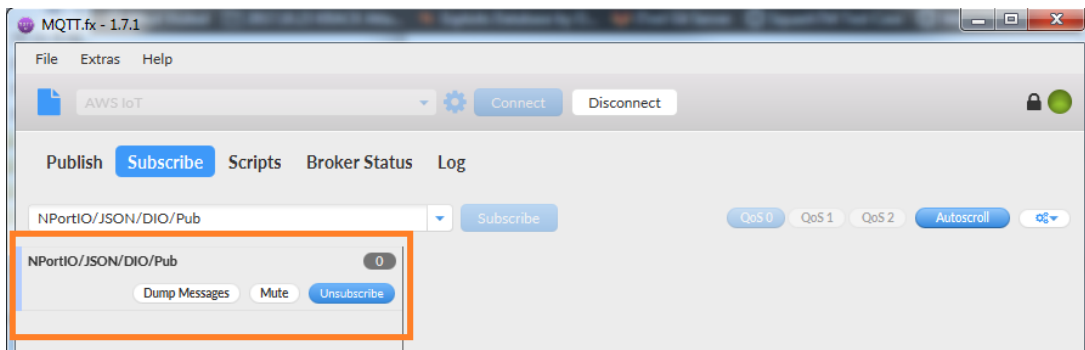
1. In the **MQTT.fx** window, click the **Subscribe** tab.



2. To subscribe the NPort's I/O topic, fill in the topic string as **"NPortIO/JSON/DIO/Pub"** in the drop-down field, and click the **Subscribe** button.



Registered topics are listed to the left of the **Subscribe** tab.



3. Log in to NPort’s web console and change **DI assess interface** to “IoT+Web+Modbus address mapping” on the **Remote I/O Access Interface** page.

Remote I/O Access Interface

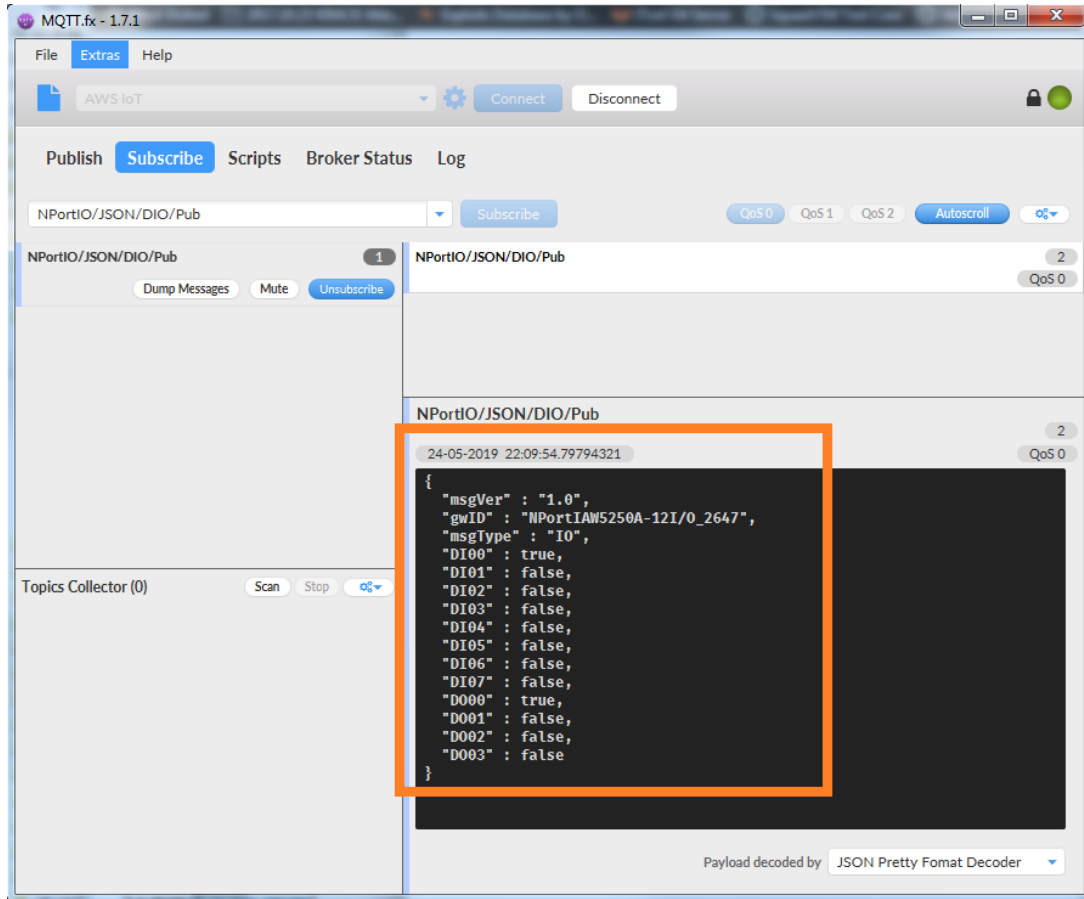
DI Channels	
DI access interface	IoT + Web + Modbus address mapping ▼
DO Channels	
DO access interface	IoT + Web + Modbus address mapping

4. Change **DO Status** to “ON” on the **DO Channel 0 Settings** page.

DO Channel 0 Settings

Mode	DO Status	ON Width*	OFF Width*	Pulse Count	Pulse Start
1. Current Setting					
DO ▼	ON ▼				
2. Power On Setting					
	OFF ▼				
3. Safe Status Setting					
	OFF ▼				
Apply to all					
<input type="checkbox"/> Apply to all DO channels					
4. Alias Name					
Alias name of channel					
DO-00					
Alias name of "OFF" status					
OFF					
Alias name of "ON" status					
ON					

5. On the **MQTT.fx** page, you will receive a message from the cloud that was sent from the NPort.



The DI and DO status will appear in JSON message format. The DI00 and DO00 status will read as "true".

```
NPortIO/JSON/DIO/Pub
24-05-2019 22:09:54.79794321
{
  "msgVer" : "1.0",
  "gwID" : "NPortIAW5250A-12I/O_2647",
  "msgType" : "IO",
  "DI00" : true,
  "DI01" : false,
  "DI02" : false,
  "DI03" : false,
  "DI04" : false,
  "DI05" : false,
  "DI06" : false,
  "DI07" : false,
  "DO00" : true,
  "DO01" : false,
  "DO02" : false,
  "DO03" : false
}
```

3.4 Control the NPort's DO Status Through the Cloud

In this section, we will instruct you on how to change an NPort's DO status via cloud. First, we use MQTT.fx to publish the NPort's I/O topic; then, the NPort will receive an IoT message from the cloud to change the DO status; lastly, we will set the NPorts' DO status on the web console.

1. Click **I/O JSON**.

IoT Mode

Basic Settings

IoT platform MQTT Broker

MQTT Connection Settings

Host address ats.iot.us-west-2.amaz

Host port 8883

Username

Password

Client ID 1136d71d-ead0-4dde-ae64-bb8ef9986e Generate

Keep alive 60 (1 - 65535 sec.)

Clean session enable

TLS (Transport Layer Security)

TLS mode TLS v1.2

CA file AmazonRootCA1.pem Browse... No file selected.

Client certificate file certificate.pem.crt1 Browse... No file selected.

Client key file private.pem.key Browse... No file selected.

Client key password

MQTT Will Message

Enable Will message enable

Serial and I/O Message Format Settings

Message format JSON Raw

Serial and I/O JSON message definition Serial JSON **I/O JSON**

I/O publish trigger mode Specific I/O change DI-00

Copy **Subscribe JSON Message**:

```
Subscribe JSON Message

The following DI and DO key-values are all optional

{
  "msgVer"      :      "1.0",
  "gwID"        :      "NPortIAW5250A-12I/O_2647",
  "msgType"     :      "IO",
  "DO00"        :      true/false,
  "DO01"        :      true/false,
  "DO02"        :      true/false,
  "DO03"        :      true/false
}
```

2. The copied message has a lot of space and line feed. We can use a tool to compact it.

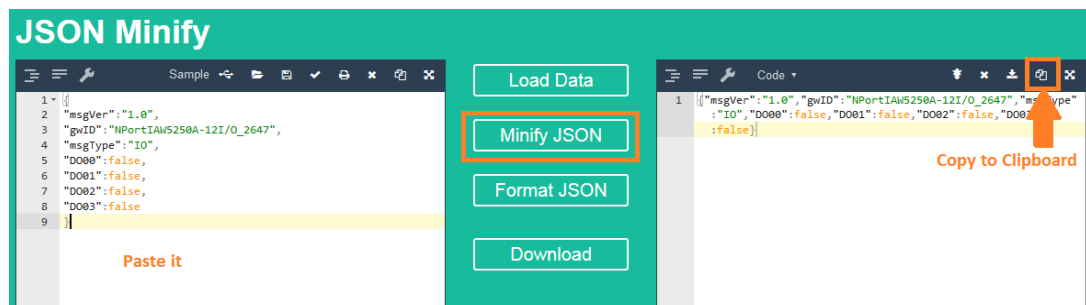
Below is a free online tool:

<https://jsonformatter.org/json-minify>

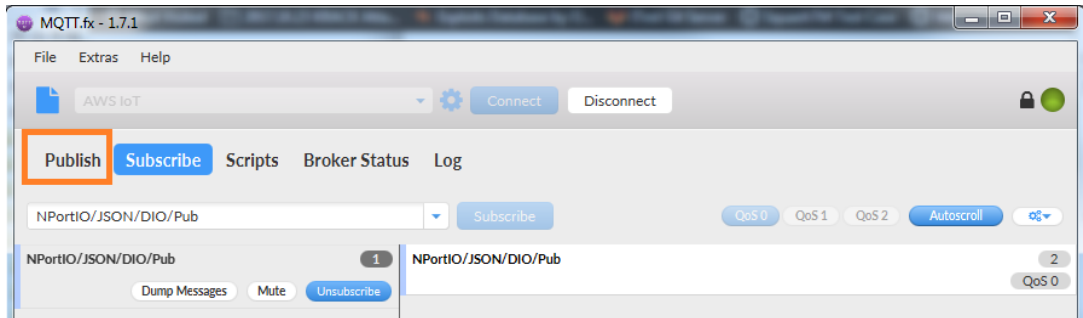
Paste the message in the column on the left side and change all the DO statuses to false.



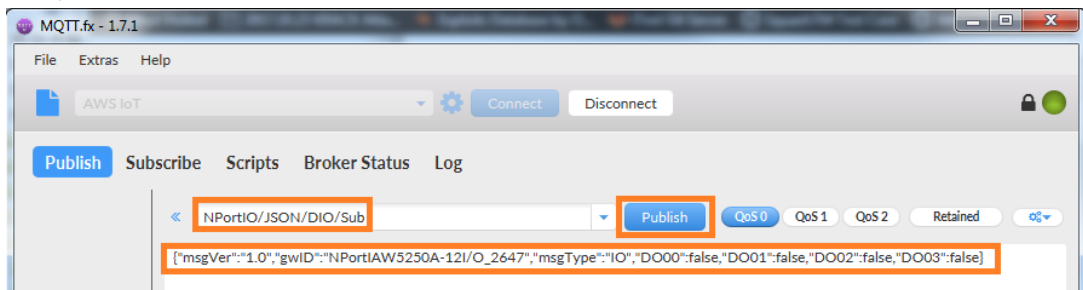
Click **Minify JSON**. It will show a compact JSON format message in the column on the right. Click **Copy to Clipboard**.



3. On the **MQTT.fx** page, click the **Publish** tab.



4. To publish a topic to the NPort. Under the **Publish** tab, paste the clipboard message in the big textbox, fill in the topic string as "NPortIO/JSON/DIO/Sub" in the drop-down field, and click the **Publish** button.



On the NPort's web console, check DO-00 status as OFF.

DO Channel Settings

DO Channel	Mode	Status	ON Width	OFF Width
DO-00	DO	OFF	--	--
DO-01	DO	OFF	--	--
DO-02	DO	OFF	--	--
DO-03	DO	OFF	--	--

Also you can find the new message under the **Subscribe** tab of MQTT.fx, because we connect DO-00 to DI-00. The new message shows both DI-00 and DO-00 statuses as false.

