NPort IA5000A-I/O or NPort IAW5000A-I/O to Mosquitto MQTT

Moxa Technical Support Team
support@moxa.com

Contents

Released on May 31, 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

Reliable Networks ▲ Sincere Service

1. Introduction

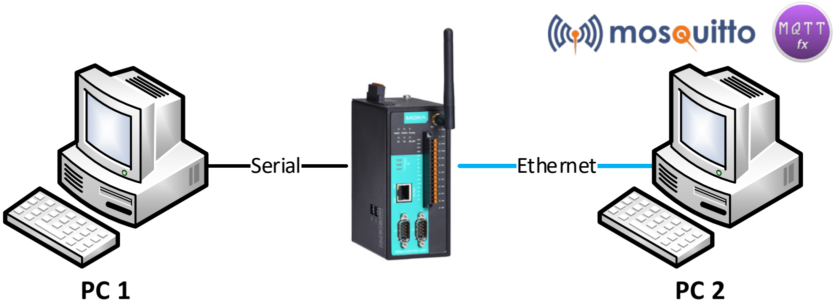
The NPort IA5000A-I/O and NPort IAW5000A-I/O serial device servers, which have built-in digital I/Os, provide maximum flexibility when you need to integrate serial equipment in the field with an Ethernet network or cloud platform. From Firmware Version 2.0 onwards, they support communications with IIoT 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 devices connecting to Eclipse Mosquitto MQTT Broker. We also demonstrate how to publish serial or I/O data messages to Mosquitto MQTT Broker and subscribe messages from Mosquitto MQTT Broker.

2. System Topology

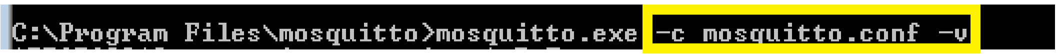
Figure 1 illustrates the system topology. PC1 runs PComm Lite to act as a series device. It connects to Port 1 of the NPort IA5000A-I/O or NPort IAW5000A-I/O serial device. The NPort IA5000A-I/O or NPort IAW5000A-I/O serial device acts as a MQTT Client device and connect to Mosquitto MQTT Broker. PC2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

< Figure 1. System Topology >



2.1 Set Up MQTT Broker(Server)—Mosquitto

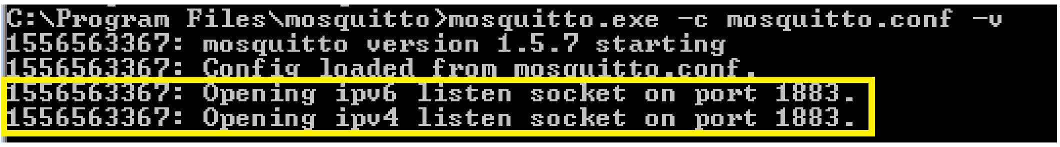
1. Download Mosquitto and install it on PC 2. It can be download from https://mosquitto.org/
2. To launch mosquitto (PC 2's OS is Windows 7 x64 and broker's OS is Mosquitto x64 version; the default folder: C:\Program Files\mosquitto) via Command Processor with default configuration and verbose mode, the command is **mosquitto.exe -c mosquitto.conf -v**.



```
C:\Program Files\mosquitto>mosquitto.exe -c mosquitto.conf -v
```

Parameter **-c** is using a specific config file: -v, which is verbose mode.

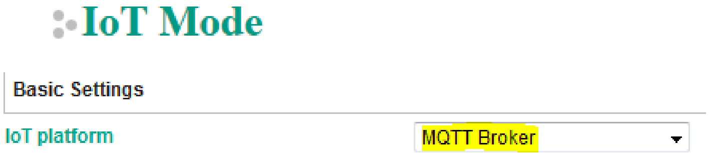
3. After successfully launching Mosquitto, Mosquitto will listen on default port 1883.



```
C:\Program Files\mosquitto>mosquitto.exe -c mosquitto.conf -v
1556563367: mosquitto version 1.5.7 starting
1556563367: Config loaded from mosquitto.conf.
1556563367: Opening ipv6 listen socket on port 1883.
1556563367: Opening ipv4 listen socket on port 1883.
```

2.2 Set Up NPort IA5000A-I/O and NPort IAW5000A-I/O to Connect to MQTT Broker

1. Log in to the NPort IA5000A-I/O or NPort IAW5000A-I/O's web console, on **Main Menu → IoT Management → IoT Mode**, to set IoT platform as "MQTT Broker".

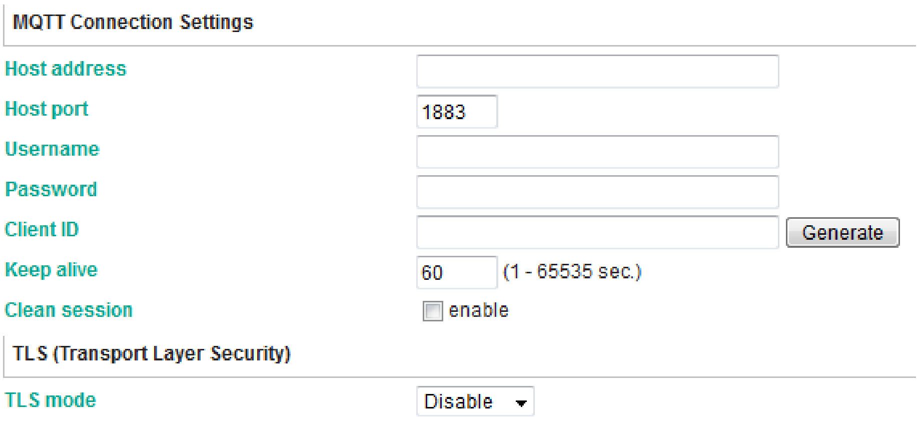


IoT Mode

Basic Settings

IoT platform	MQTT Broker ▾

The IoT Mode is running with MQTT Broker; it will show more settings about MQTT as below:



MQTT Connection Settings

Host address	
Host port	1883
Username	
Password	
Client ID	Generate
Keep alive	60 (1 - 65535 sec.)
Clean session	□ enable

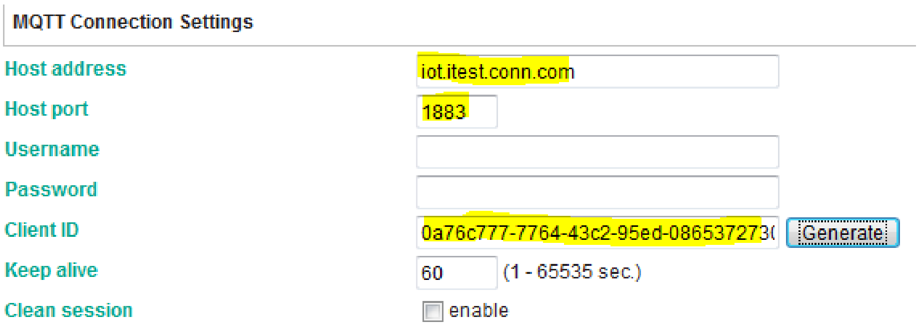
TLS (Transport Layer Security)

TLS mode	Disable ▾

2. In **MQTT Connection Settings → Host address** string, fill in your MQTT Broker IP address or domain name, and **Host port** as "1883" (The Mosquitto default port is 1883).

   Client ID setting sets the identity of the MQTT session. It must be unique. Broker doesn't accept the same Client ID connection. You can fill in an identifiable ID or click **Generate** button to generate a random ID.

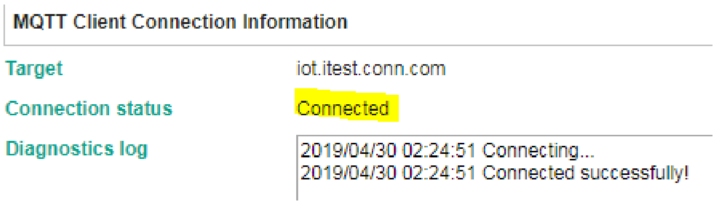
   Broker may need the client to provide the username and password to authenticate the client connection. If needed, fill in the correct username and password. For more about customize MQTT Broker secure settings, please reference the chapter "Customize MQTT Broker settings".



MQTT Connection Settings

Host address	iot.itest.conn.com
Host port	1883
Username	
Password	
Client ID	0a76c777-7764-43c2-95ed-086537273 Generate
Keep alive	60 (1 - 65535 sec.)
Clean session	□ enable

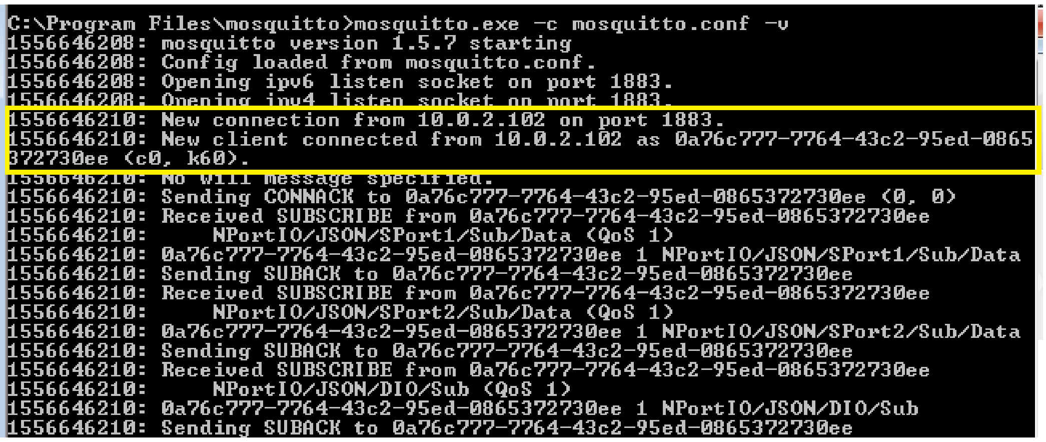
   After clicking **Submit**, the NPort IA5000A-I/O or NPort IAW5000A-I/O will connect to MQTT Broker, and you can check that **Connection status** shows "Connected" on "IoT Connection Monitoring", as below:



MQTT Client Connection Information

Target	iot.itest.conn.com
Connection status	Connected
Diagnostics log	2019/04/30 02:24:51 Connecting... 2019/04/30 02:24:51 Connected successfully!

Also, you will see a new client connected, of which the Client ID is NPort on the MQTT Broker side. The information will appear as below:

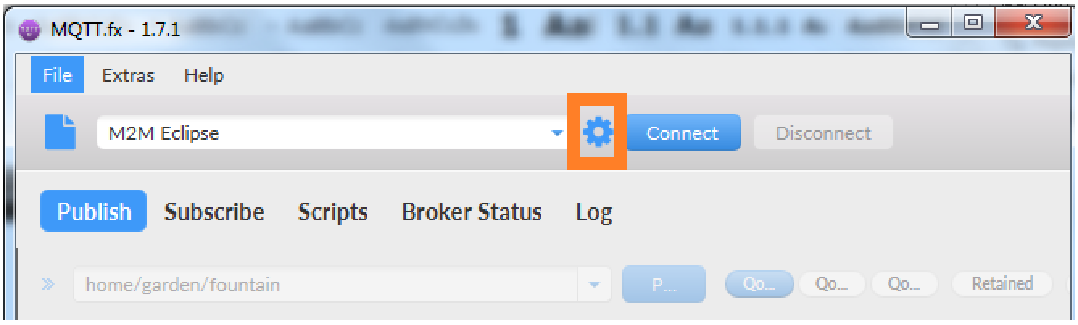


```
C:\Program Files\mosquitto>mosquitto.exe -c mosquitto.conf -v
1556646208: mosquitto version 1.5.7 starting
1556646208: Config loaded from mosquitto.conf.
1556646208: Opening ipv6 listen socket on port 1883.
1556646208: Opening ipv4 listen socket on port 1883.
1556646210: New connection from 10.0.2.102 on port 1883.
1556646210: New client connected from 10.0.2.102 as 0a76c777-7764-43c2-95ed-0865
372730ee (c0, k60).
1556646210: No will message specified.
1556646210: Sending CONNACK to 0a76c777-7764-43c2-95ed-0865372730ee (0, 0)
1556646210: Received SUBSCRIBE from 0a76c777-7764-43c2-95ed-0865372730ee
1556646210:     NPortIO/JSON/SPort1/Sub/Data (QoS 1)
1556646210: 0a76c777-7764-43c2-95ed-0865372730ee 1 NPortIO/JSON/SPort1/Sub/Data
1556646210: Sending SUBACK to 0a76c777-7764-43c2-95ed-0865372730ee
1556646210: Received SUBSCRIBE from 0a76c777-7764-43c2-95ed-0865372730ee
1556646210:     NPortIO/JSON/SPort2/Sub/Data (QoS 1)
1556646210: 0a76c777-7764-43c2-95ed-0865372730ee 1 NPortIO/JSON/SPort2/Sub/Data
1556646210: Sending SUBACK to 0a76c777-7764-43c2-95ed-0865372730ee
1556646210: Received SUBSCRIBE from 0a76c777-7764-43c2-95ed-0865372730ee
1556646210:     NPortIO/JSON/DIO/Sub (QoS 1)
1556646210: 0a76c777-7764-43c2-95ed-0865372730ee 1 NPortIO/JSON/DIO/Sub
1556646210: Sending SUBACK to 0a76c777-7764-43c2-95ed-0865372730ee
```

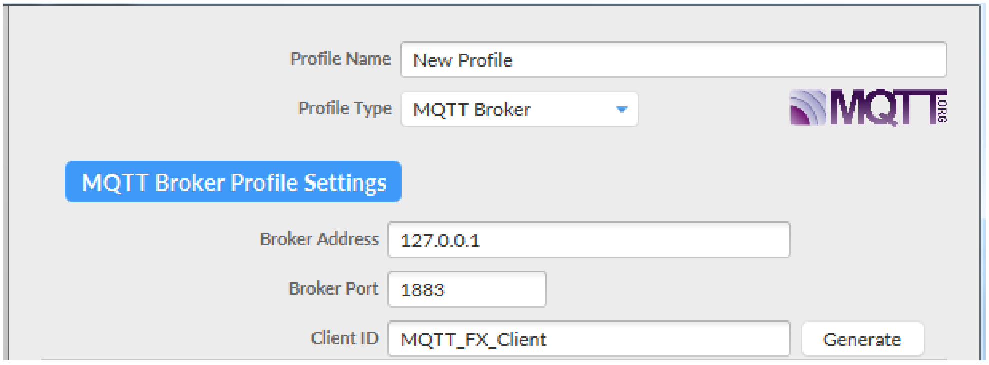
2.3 Set Up MQTT.fx(Client) Connect to MQTT Broker

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**. It 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 profile settings.



An **Edit Connetion Profile** window will pop up.



Profile Name: New Profile

Profile Type: MQTT Broker

MQTT Broker Profile Settings

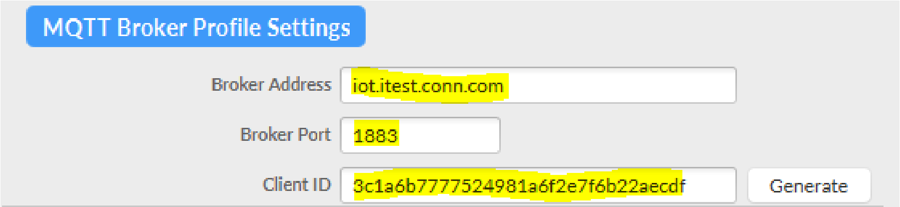
Broker Address: 127.0.0.1

Broker Port: 1883

Client ID: MQTT_FX_Client

3. To configure the **MQTT Broker Profile Settings** session, set **Profile Name** as “Mosquitto”, and“Profile Type as “MQTT Broker”. For **Broker Address**, fill in your MQTT Broker IP address or domain name, and for **Broker Port** fill in “1883”. **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.

   Broker may need the client to provide the username and password to authenticate client connection. If you need it, fill in the correct username and password.



MQTT Broker Profile Settings

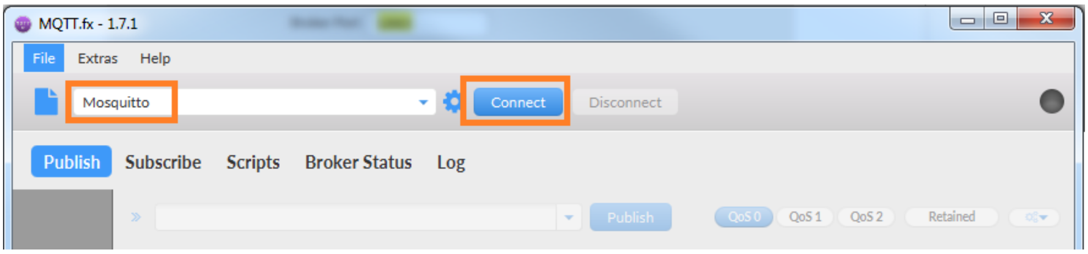
Broker Address: iot.itest.conn.com

Broker Port: 1883

Client ID: 3c1a6b7777524981a6f2e7f6b22aecdf

   Click OK to confirm and close the window.

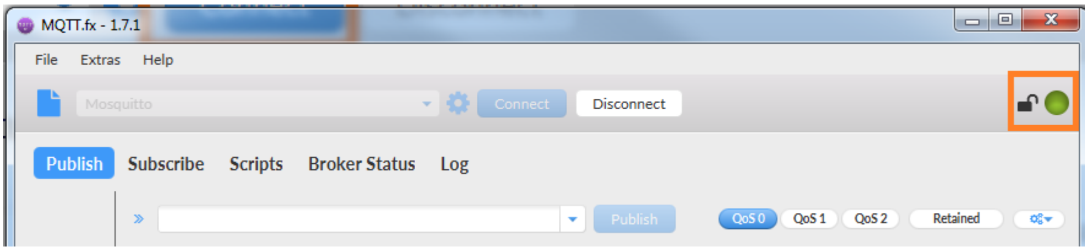
4. For a profile name, select “Mosquitto” and click **Connect**.



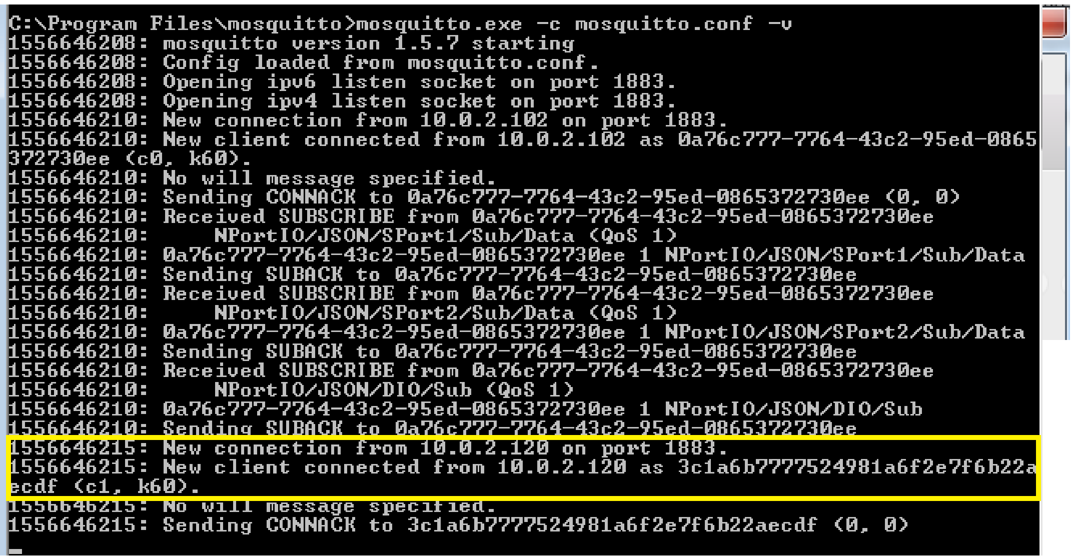
MQTT.fx - 1.7.1

Mosquitto

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



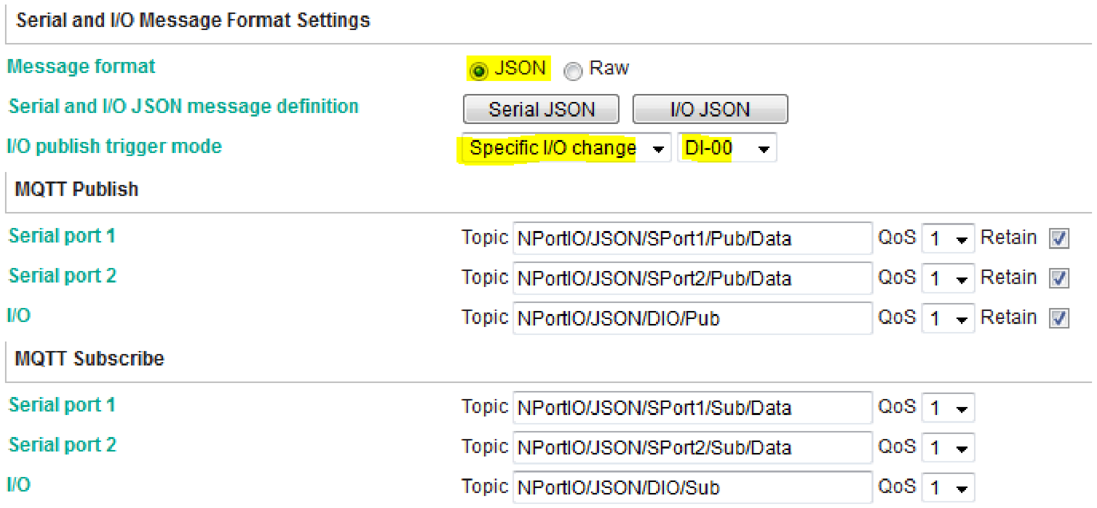
Also, you will see a new client connected, of which the Client ID is MQTT.fx on the MQTT Broker side. The information will be shown as below:



```
C:\Program Files\mosquitto>mosquitto.exe -c mosquitto.conf -v
1556646208: mosquitto version 1.5.7 starting
1556646208: Config loaded from mosquitto.conf.
1556646208: Opening ipv6 listen socket on port 1883.
1556646208: Opening ipv4 listen socket on port 1883.
1556646210: New connection from 10.0.2.102 on port 1883.
1556646210: New client connected from 10.0.2.102 as 0a76c777-7764-43c2-95ed-0865
372730ee (c0, k60).
1556646210: No will message specified.
1556646210: Sending CONNACK to 0a76c777-7764-43c2-95ed-0865372730ee (0, 0)
1556646210: Received SUBSCRIBE from 0a76c777-7764-43c2-95ed-0865372730ee
1556646210:     NPortIO/JSON/SPort1/Sub/Data (QoS 1)
1556646210: 0a76c777-7764-43c2-95ed-0865372730ee 1 NPortIO/JSON/SPort1/Sub/Data
1556646210: Sending SUBACK to 0a76c777-7764-43c2-95ed-0865372730ee
1556646210: Received SUBSCRIBE from 0a76c777-7764-43c2-95ed-0865372730ee
1556646210:     NPortIO/JSON/SPort2/Sub/Data (QoS 1)
1556646210: 0a76c777-7764-43c2-95ed-0865372730ee 1 NPortIO/JSON/SPort2/Sub/Data
1556646210: Sending SUBACK to 0a76c777-7764-43c2-95ed-0865372730ee
1556646210: Received SUBSCRIBE from 0a76c777-7764-43c2-95ed-0865372730ee
1556646210:     NPortIO/JSON/DIO/Sub (QoS 1)
1556646210: 0a76c777-7764-43c2-95ed-0865372730ee 1 NPortIO/JSON/DIO/Sub
1556646210: Sending SUBACK to 0a76c777-7764-43c2-95ed-0865372730ee
1556646215: New connection from 10.0.2.120 on port 1883.
1556646215: New client connected from 10.0.2.120 as 3c1a6b7777524981a6f2e7f6b22a
ecdf (c1, k60).
1556646215: No will message specified.
1556646215: Sending CONNACK to 3c1a6b7777524981a6f2e7f6b22aecdf (0, 0)
```

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

In this section, we will instruct you on how the NPort IA5000A-I/O and NPort IAW5000A-I/O (in the chapters below referenced to 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 I/O status to the cloud, and receive a message from the cloud to NPorts to change I/O status. The NPorts support two kinds of MQTT data message formats: JSON and RAW. In this demonstration, we use JSON format. We select “JSON” for **Message format** and “Specific I/O change”, along with “DI-00” for **I/O publish trigger mode**.



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 ▾

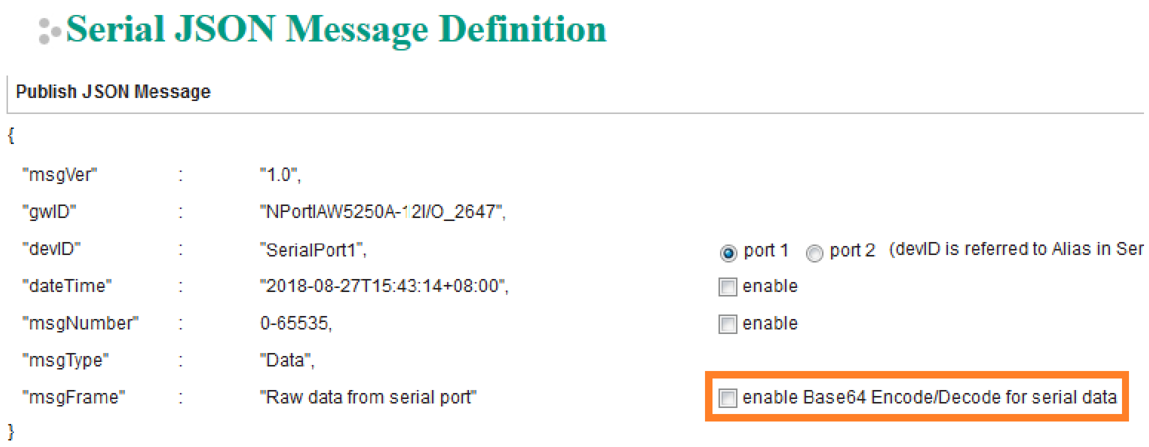
MQTT Publish

Serial port 1	Topic NPortIO/JSON/SPort1/Pub/Data	QoS 1	Retain ☑
Serial port 2	Topic NPortIO/JSON/SPort2/Pub/Data	QoS 1	Retain ☑
I/O	Topic NPortIO/JSON/DIO/Pub	QoS 1	Retain ☑

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 special characters. If needed, set correct Encode/Decode for serial data. For more about JSON format rules, please reference http://json.org/

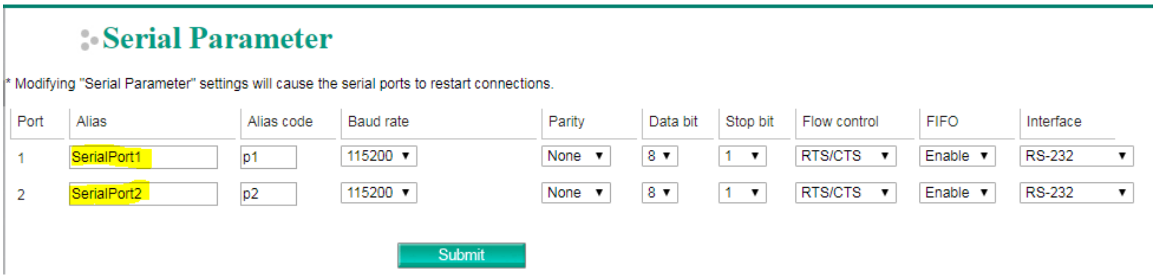


Serial JSON Message Definition

Publish JSON Message

```
{
  "msgVer"      :   "1.0",
  "gwID"        :   "NPortIAW5250A-12I/O_2647",
  "devID"       :   "SerialPort1",                    ● port 1  ○ port 2  (devID is referred to Alias in Ser
  "dateTime"    :   "2018-08-27T15:43:14+08:00",      □ enable
  "msgNumber"   :   0-65535,                          □ enable
  "msgType"     :   "Data",
  "msgFrame"    :   "Raw data from serial port"       □ 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.

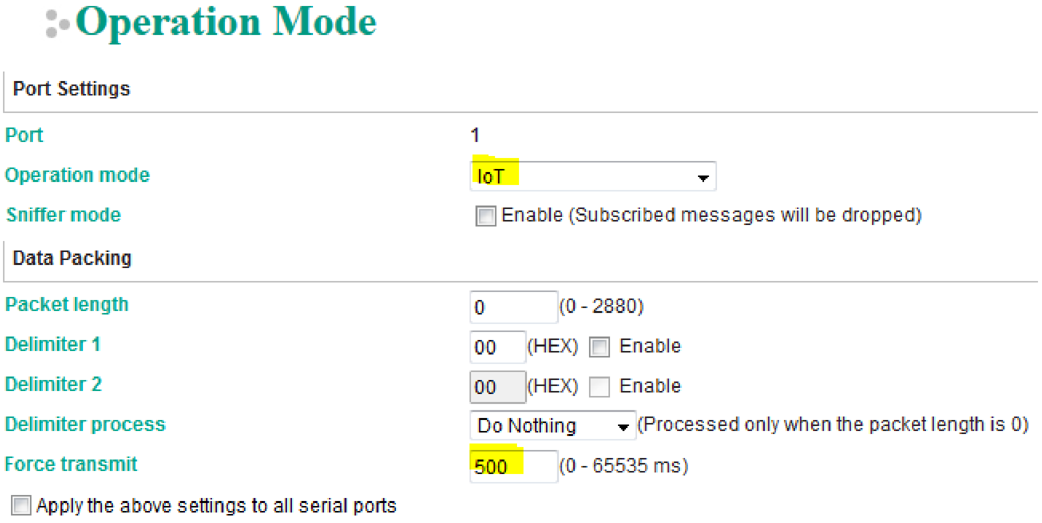
Port	Alias	Alias code	Baud rate	Parity	Data bit	Stop bit	Flow control	FIFO	Interface
1	SerialPort1	p1	115200	None	8	1	RTS/CTS	Enable	RS-232
2	SerialPort2	p2	115200	None	8	1	RTS/CTS	Enable	RS-232

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

3.1 Send Serial Pattern From the Device to the Cloud

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

1. Log in to NPort's web console and change Serial Port 1's **Operation mode** to IoT and **Force transmit** to 500. The NPorts support several types of data packing combinations. For the purposes of this demonstration, we use **Force transmit**. If you need to, set the correct Data Packing method.



Operation Mode

Port Settings

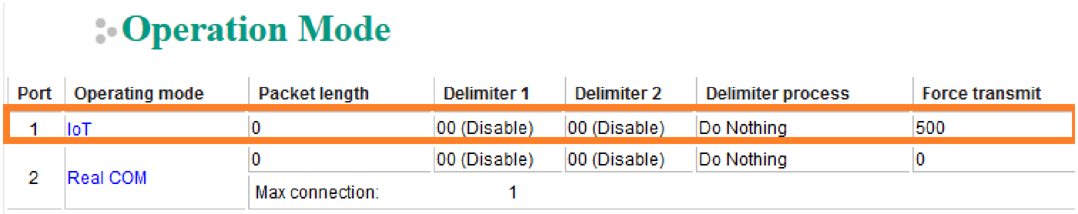
Port	1
Operation mode	IoT
Sniffer mode	□ Enable (Subscribed messages will be dropped)

Data Packing

Packet length	0 (0 - 2880)
Delimiter 1	00 (HEX) □ Enable
Delimiter 2	00 (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

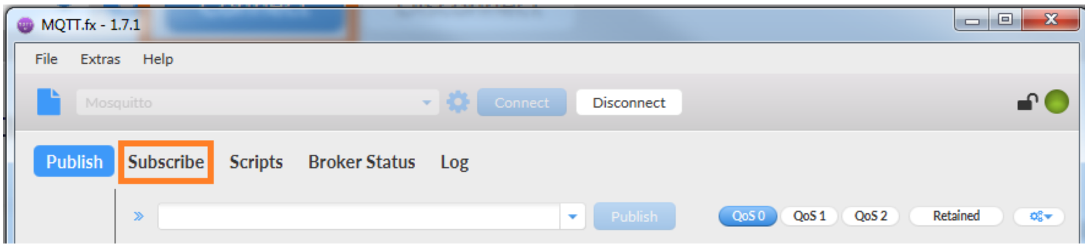
Click **Submit** to activate the configuration process.



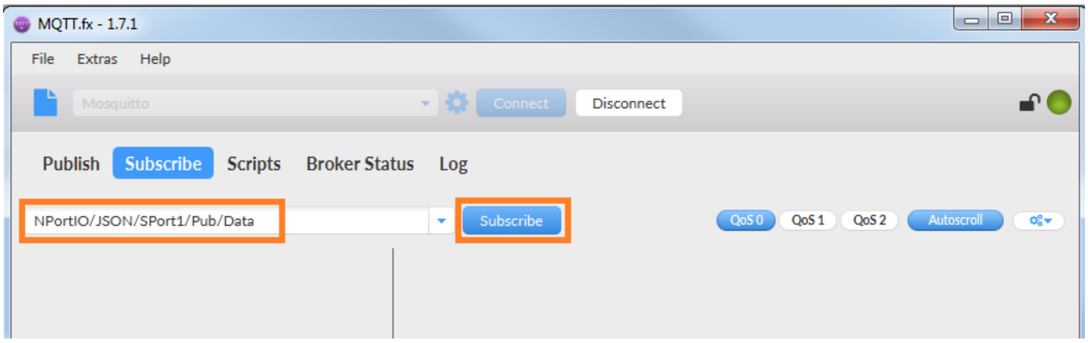
Operation Mode

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				

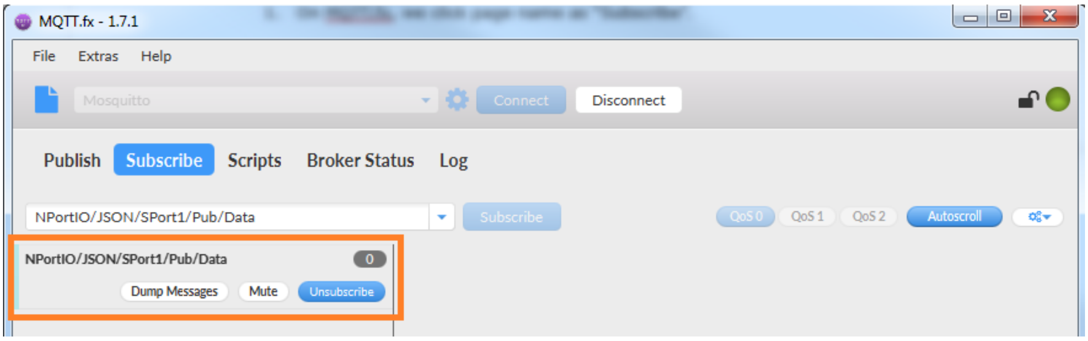
2. On the **MQTT.fx** page, click the **Subscribe** tab.



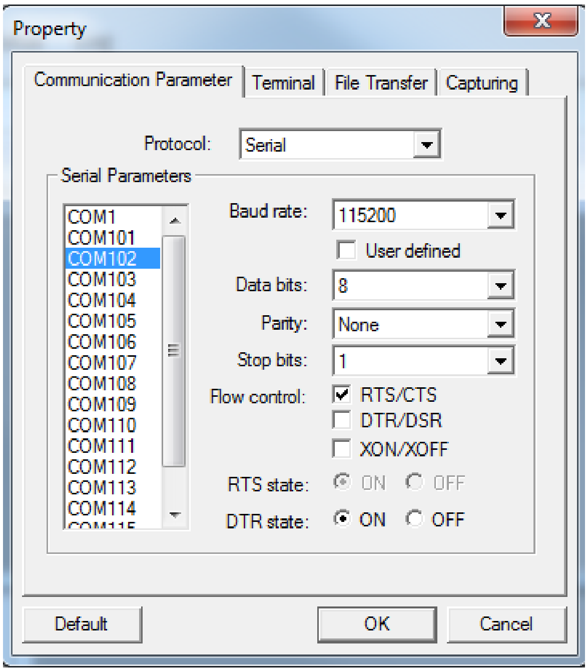
3. To subscribe a topic from NPort's Serial Port 1: under the **Subscribe** tab, fill in the topic string as "NPortIO/JSON/SPort1/Pub/Data" in the drop-down menu, and click **Subscribe**.



Registered topics are listed on 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 that are 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



Property

Communication Parameter | Terminal | File Transfer | Capturing

Protocol: Serial

Serial Parameters

COM1, COM101, COM102, COM103, COM104, COM105, COM106, COM107, COM108, COM109, COM110, COM111, COM112, COM113, COM114

Baud rate: 115200

□ User defined

Data bits: 8

Parity: None

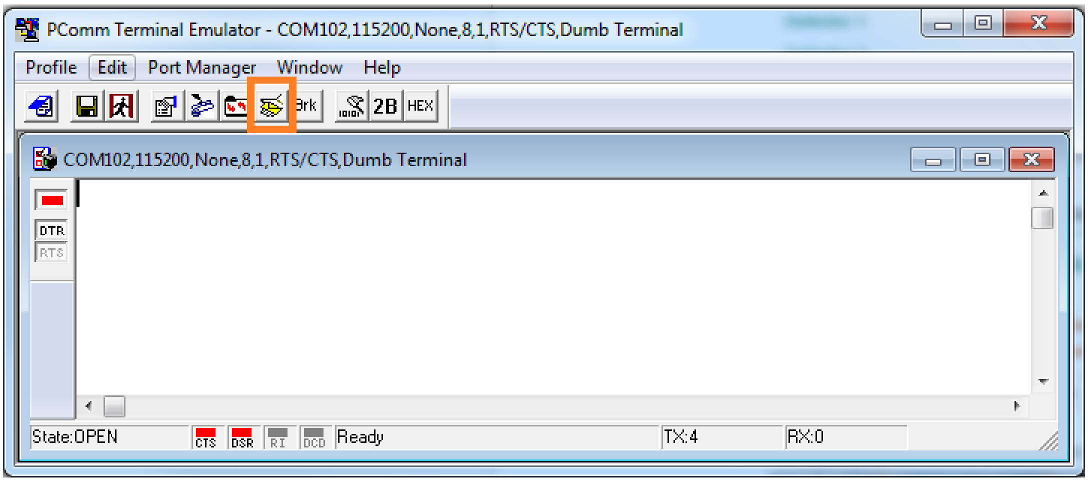
Stop bits: 1

Flow control: ☑ RTS/CTS □ DTR/DSR □ XON/XOFF

RTS state: ● ON ○ OFF

DTR state: ● ON ○ OFF

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

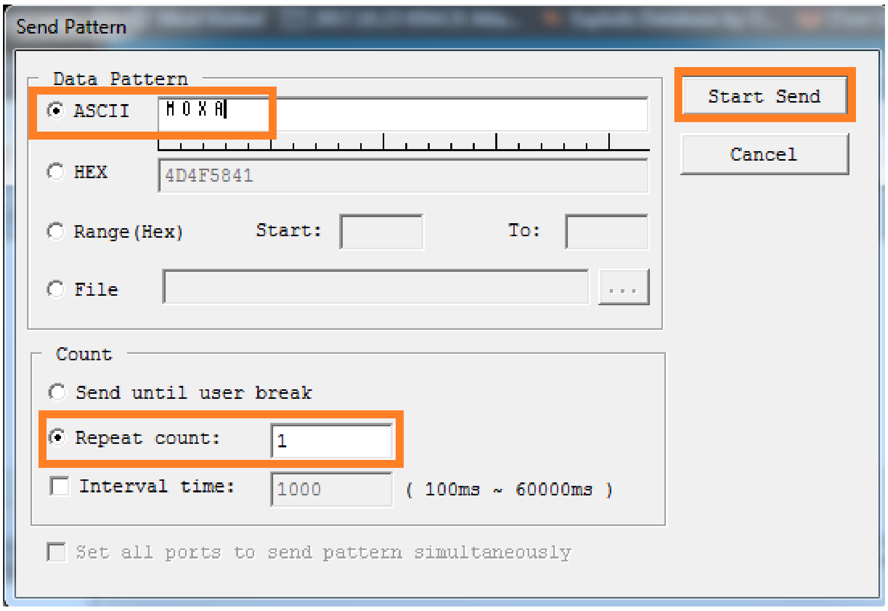


PComm Terminal Emulator - COM102,115200,None,8,1,RTS/CTS,Dumb Terminal

COM102,115200,None,8,1,RTS/CTS,Dumb Terminal

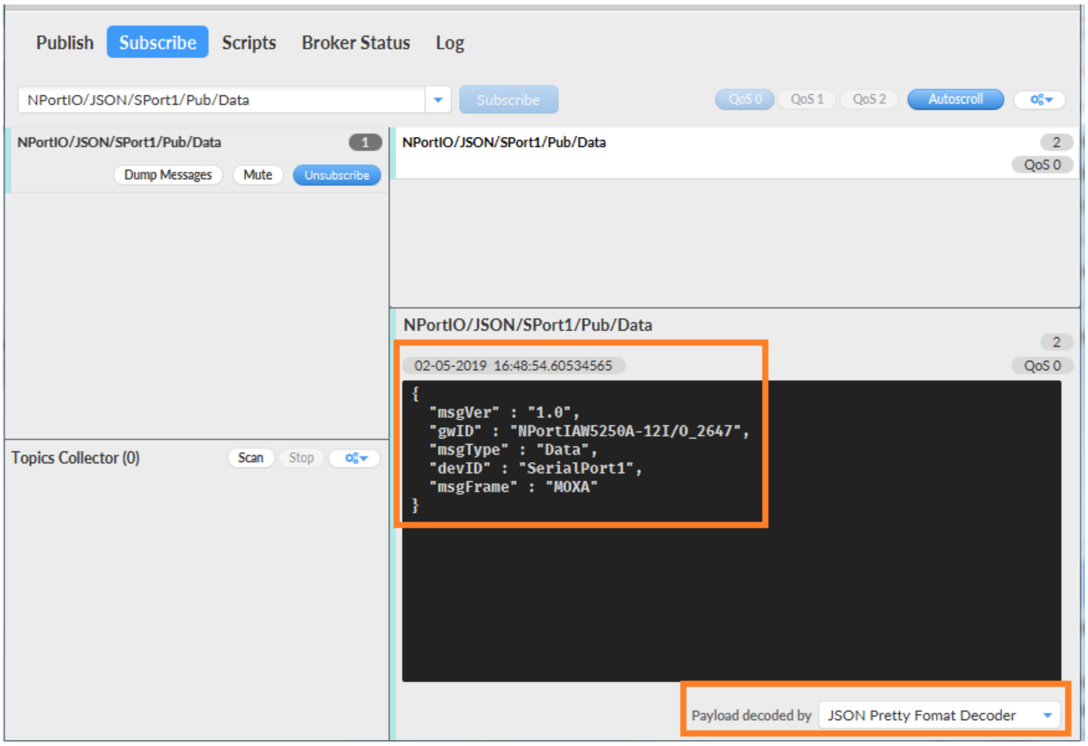
State:OPEN CTS DSR RI DCD Ready TX:4 RX:0

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.



● ASCII: MOXA
○ HEX: 4D4F5841
○ Range (Hex) Start: To:
○ File
○ Send until user break
● Repeat count: 1
□ Interval time: 1000 (100ms ~ 60000ms)
□ Set all ports to send pattern simultaneously

6. On the **MQTT.fx** page, you will successfully receive a message from the cloud that was sent from NPorts. Select "JSON Pretty Format Decoder" for **Payload decoded by,** which enhances the readability of the message.



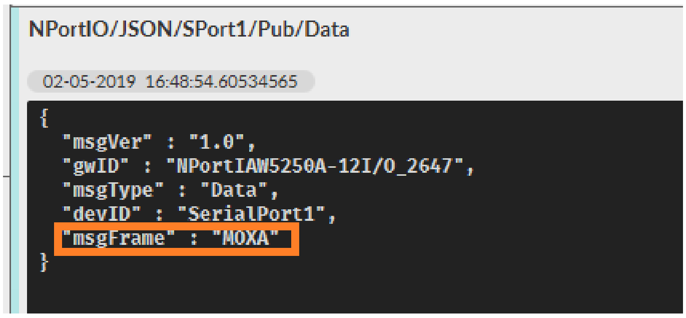
NPortIO/JSON/SPort1/Pub/Data

02-05-2019 16:48:54.60534565

```
{
  "msgVer" : "1.0",
  "gwID" : "NPortIAW5250A-12I/O_2647",
  "msgType" : "Data",
  "devID" : "SerialPort1",
  "msgFrame" : "MOXA"
}
```

Payload decoded by: JSON Pretty Fomat Decoder

The serial data pattern will be filled in msgFrame.



NPortIO/JSON/SPort1/Pub/Data

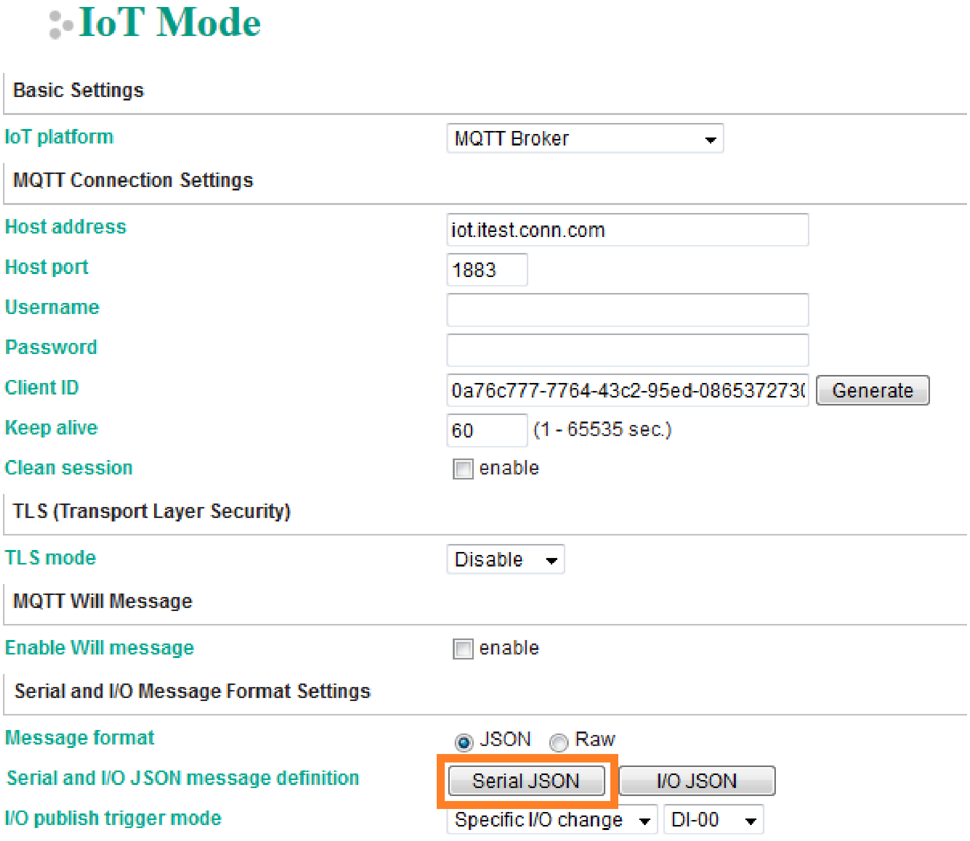
02-05-2019 16:48:54.60534565

```
{
  "msgVer" : "1.0",
  "gwID" : "NPortIAW5250A-12I/O_2647",
  "msgType" : "Data",
  "devID" : "SerialPort1",
  "msgFrame" : "MOXA"
}
```

3.2 Send Serial Data From the Cloud to the Device

In this section, we will instruct you on how to send serial data to PC 1. First, we use MQTT.fx to publish the topic of Serial Port 1 of the NPort to the cloud; second, you 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: iot.itest.conn.com
Host port: 1883
Username:
Password:
Client ID: 0a76c777-7764-43c2-95ed-086537273(Generate
Keep alive: 60 (1 - 65535 sec.)
Clean session: □ enable

TLS (Transport Layer Security)

TLS mode: Disable

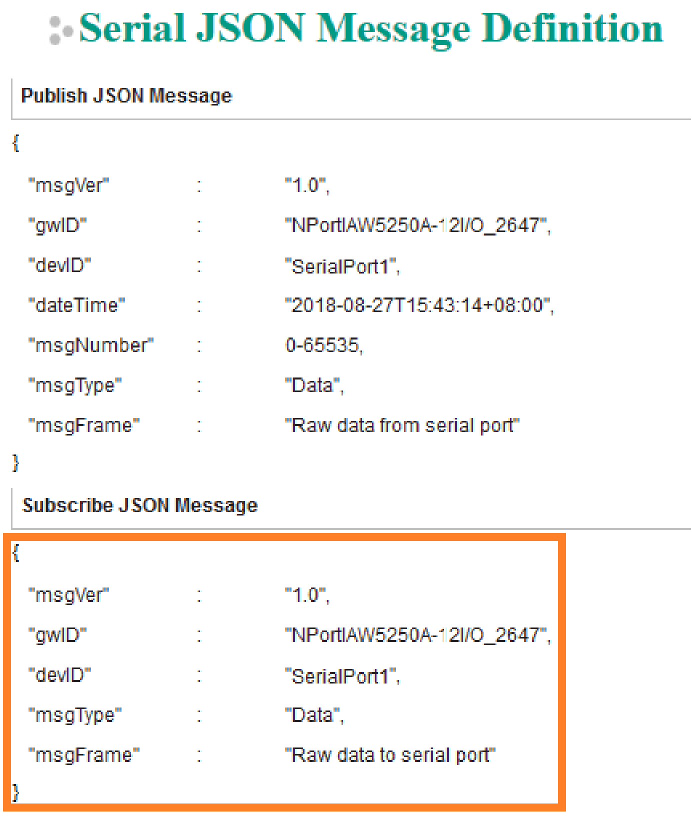
MQTT Will Message

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**:



Serial JSON Message Definition

Publish JSON Message

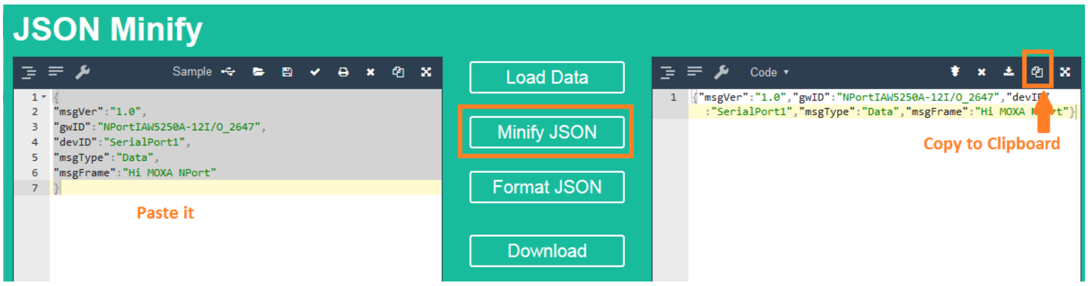
```
{
  "msgVer"       :     "1.0",
  "gwID"         :     "NPortIAW5250A-12I/O_2647",
  "devID"        :     "SerialPort1",
  "dateTime"     :     "2018-08-27T15:43:14+08:00",
  "msgNumber"    :     0-65535,
  "msgType"      :     "Data",
  "msgFrame"     :     "Raw data from serial port"
}
```

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 r column on the right. Click **Copy to Clipboard**.

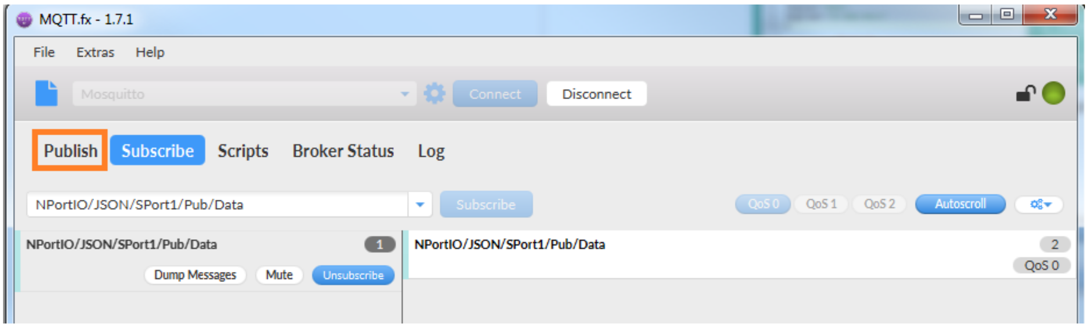


JSON Minify

```
{
"msgVer":"1.0",
"gwID":"NPortIAW5250A-12I/O_2647",
"devID":"SerialPort1",
"msgType":"Data",
"msgFrame":"Hi MOXA NPort"
}
```

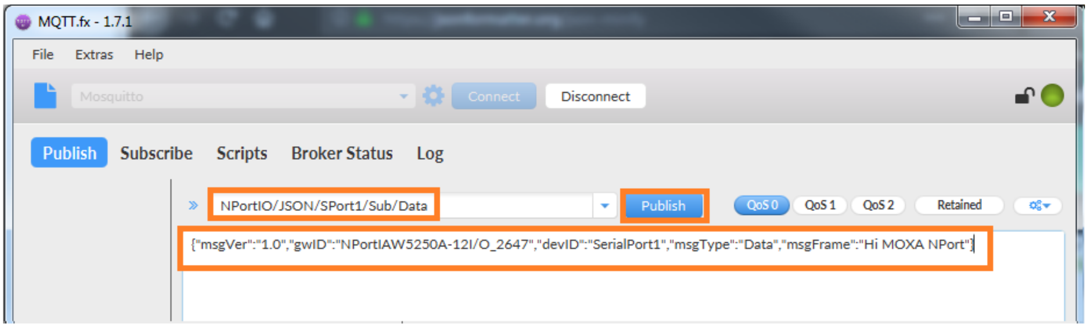
```
{"msgVer":"1.0","gwID":"NPortIAW5250A-12I/O_2647","devI
:"SerialPort1","msgType":"Data","msgFrame":"Hi MOXA N t"}
```

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



NPortIO/JSON/SPort1/Pub/Data

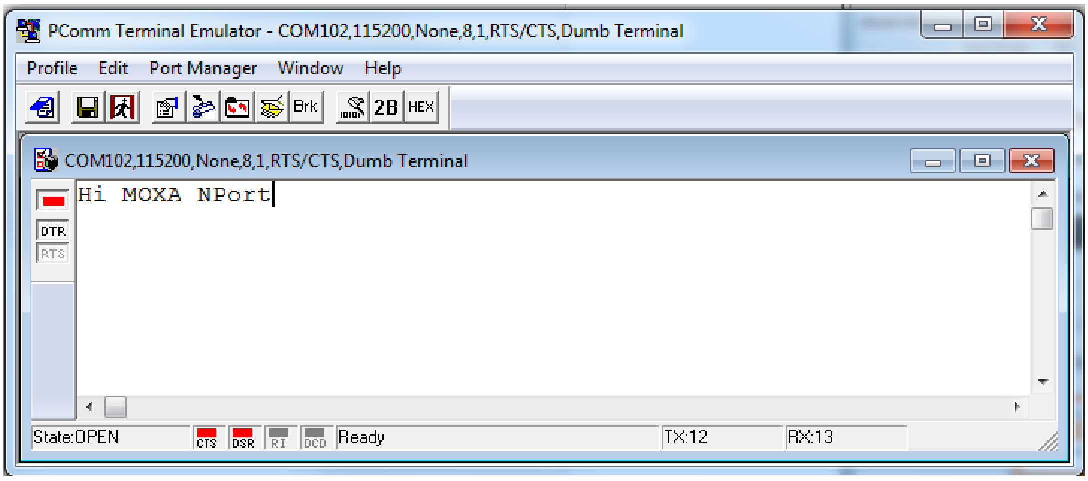
4. To publish a topic to the NPort's Serial Port 1, paste the clipboard message in the big text box. Fill in "NPortIO/JSON/SPort1/Sub/Data" as the topic string in the drop-down menu, and click **Publish**.



NPortIO/JSON/SPort1/Sub/Data

```
{"msgVer":"1.0","gwID":"NPortIAW5250A-12I/O_2647","devID":"SerialPort1","msgType":"Data","msgFrame":"Hi MOXA NPort"}
```

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


PComm Terminal Emulator - COM102,115200,None,8,1,RTS/CTS,Dumb Terminal
Profile
Edit
Port Manager
Window
Help
Brk
2B
HEX
COM102,115200,None,8,1,RTS/CTS,Dumb Terminal
DTR
RTS
CTS
DSR
RI
DCD

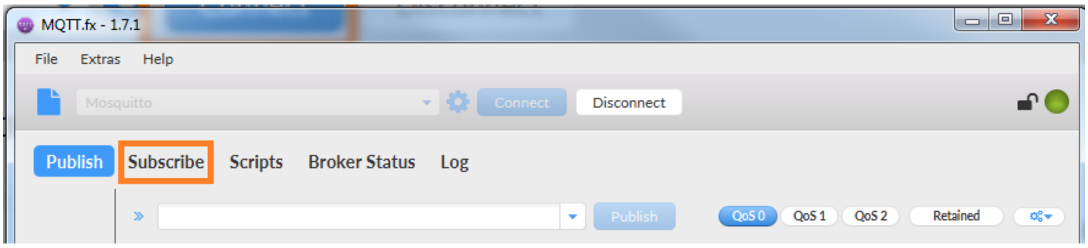
```
Hi MOXA NPort
```

State:OPEN Ready TX:12 RX:13

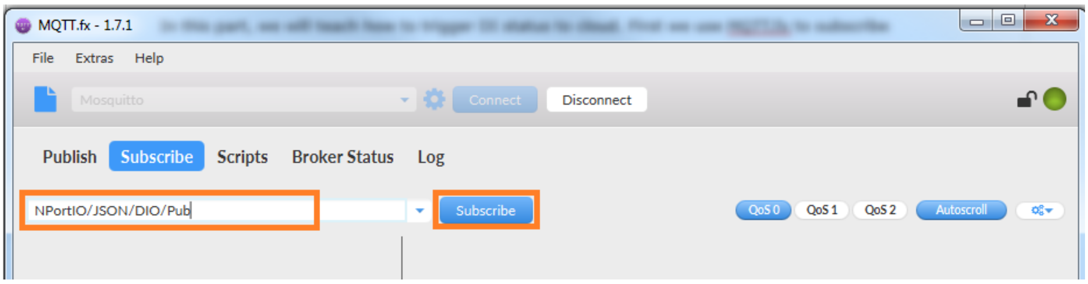
3.3 Send NPorts’ DI and DO Status to the Cloud

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

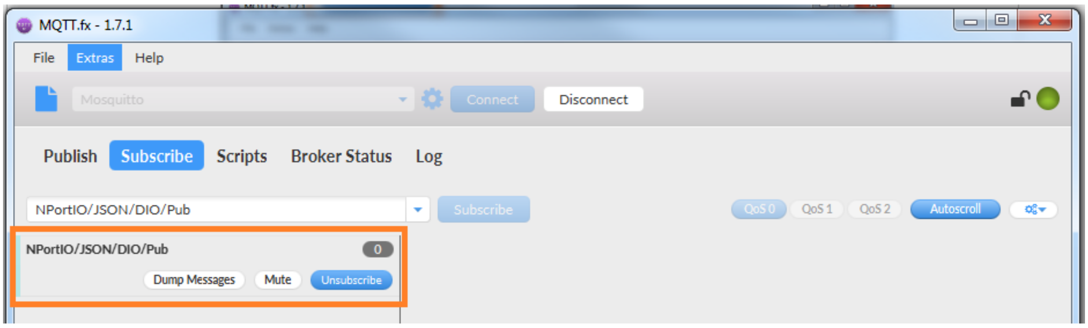
1. On the **MQTT.fx** page, click the **Subscribe** tab.



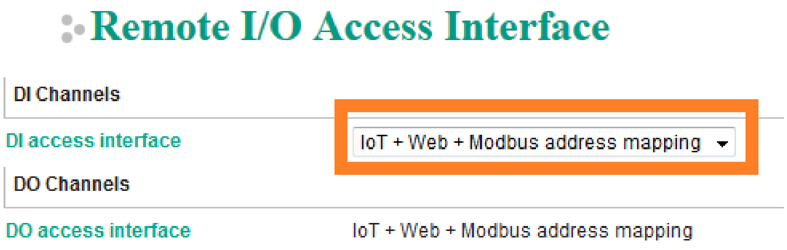
2. To subscribe the NPort’s I/O topic: under the **Subscribe** tap, fill in “NPortIO/JSON/DIO/Pub” as the topic string in the drop-down menu and click **Subscribe**.



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

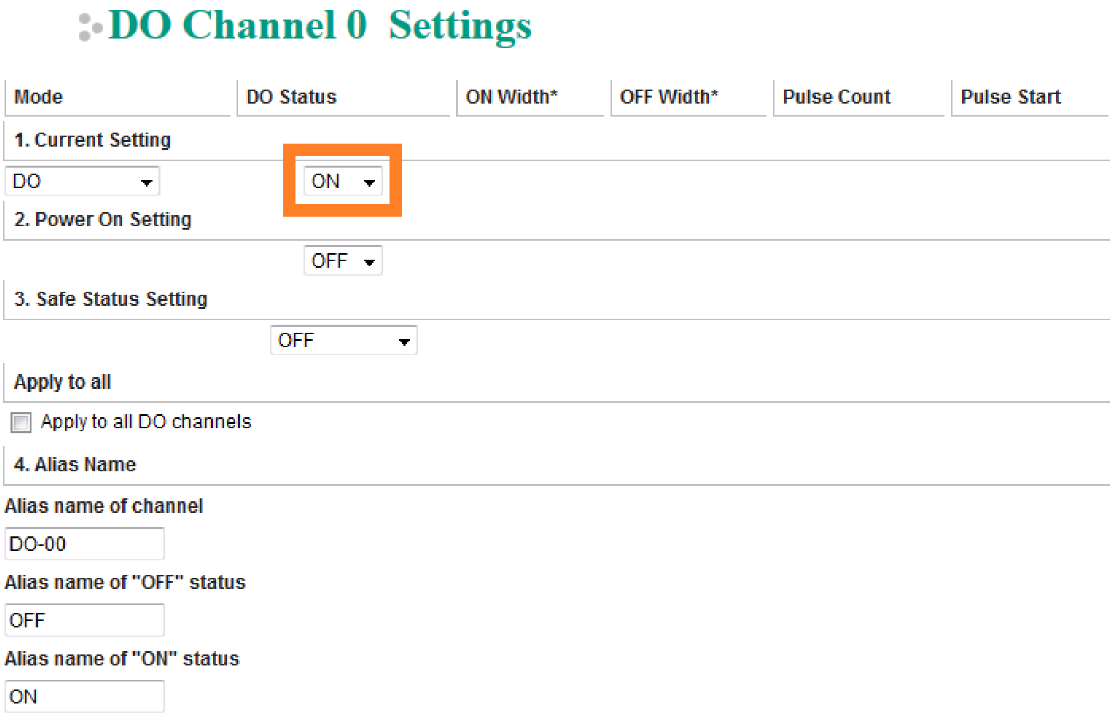
DI Channels

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

□ Apply to all DO channels

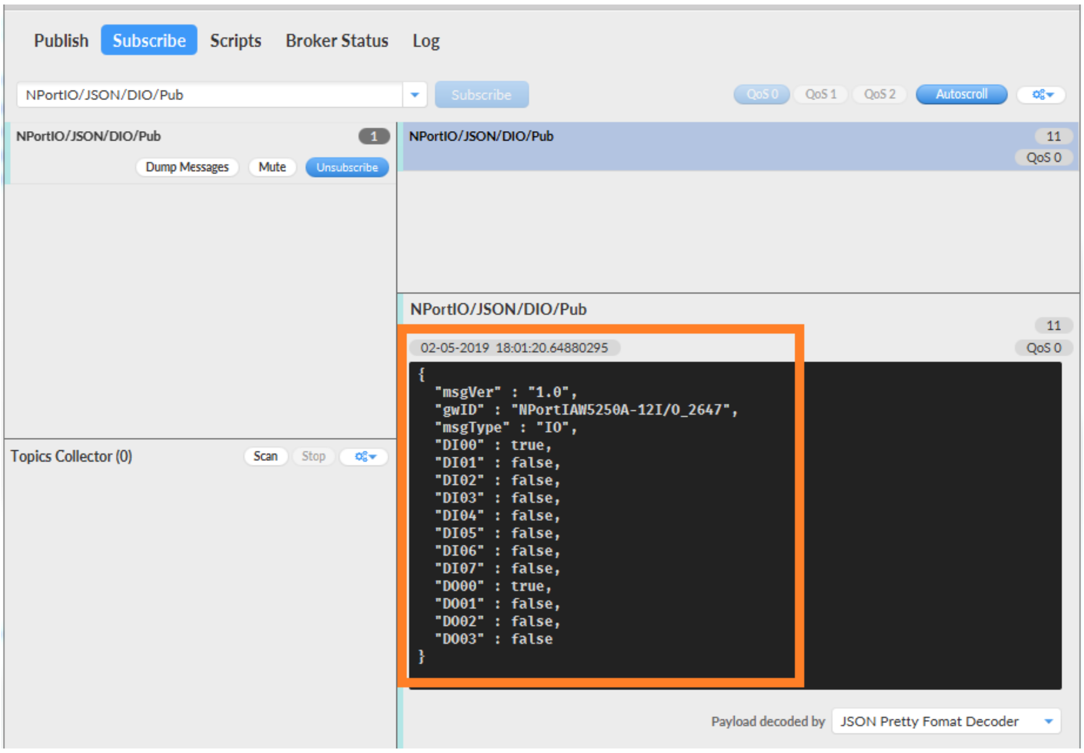
4. Alias Name

Alias name of channel: DO-00

Alias name of "OFF" status: OFF

Alias name of "ON" status: ON

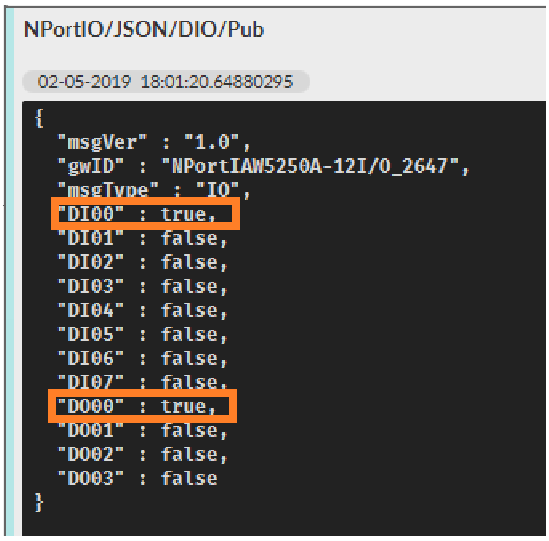
5. On the **MQTT.fx** page, you will receive a message from the cloud, which was sent from the NPort.



02-05-2019 18:01:20.64880295

```
{
  "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
}
```

The DI and DO status will be filled in JSON message. The DI00 and DO00 status will show as true.



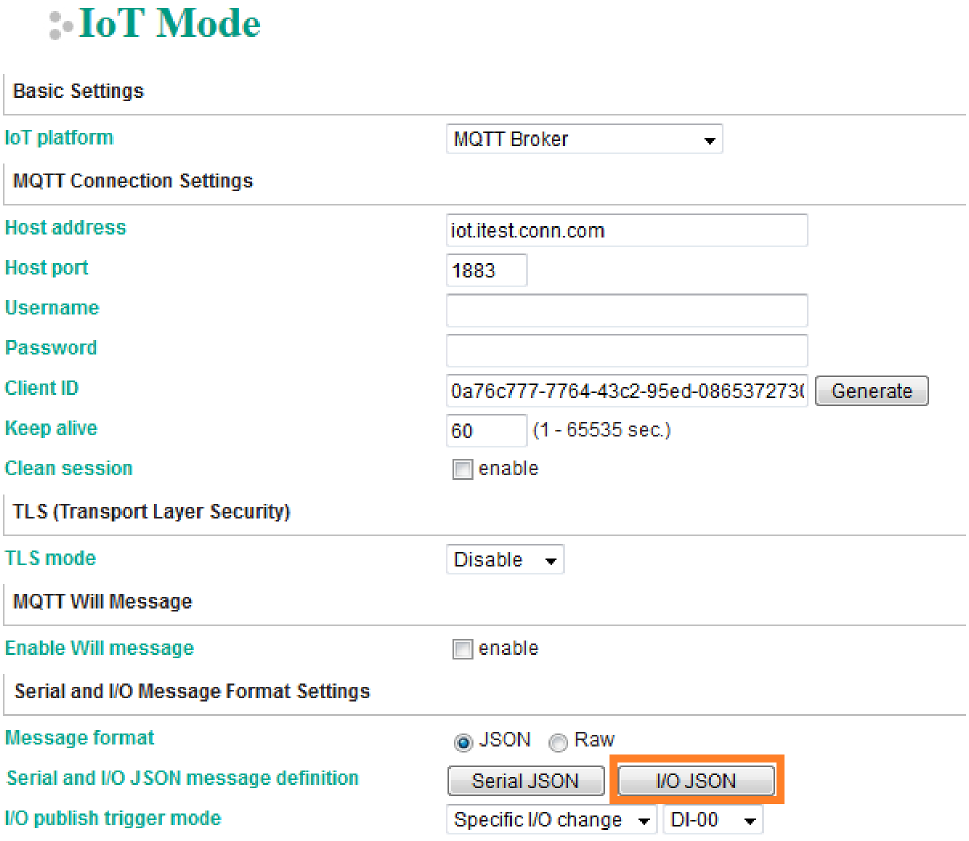
02-05-2019 18:01:20.64880295

```
{
  "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 Via the Cloud

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

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



IoT Mode

Basic Settings

IoT platform	MQTT Broker

MQTT Connection Settings

Host address	iot.itest.conn.com
Host port	1883
Username	
Password	
Client ID	0a76c777-7764-43c2-95ed-08653727З(Generate
Keep alive	60 (1 - 65535 sec.)
Clean session	□ enable

TLS (Transport Layer Security)

TLS mode	Disable

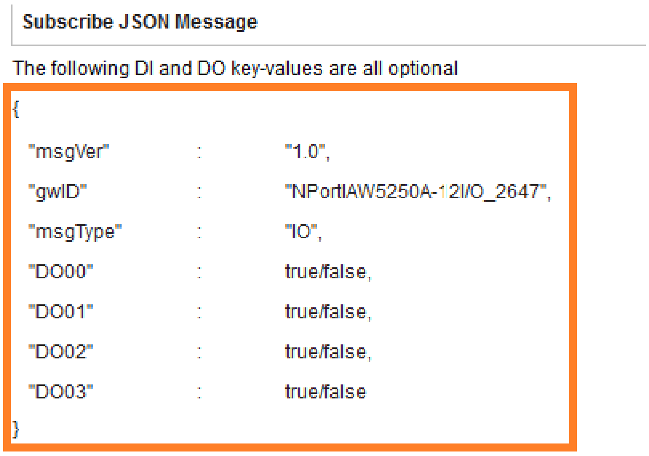
MQTT Will Message

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**:

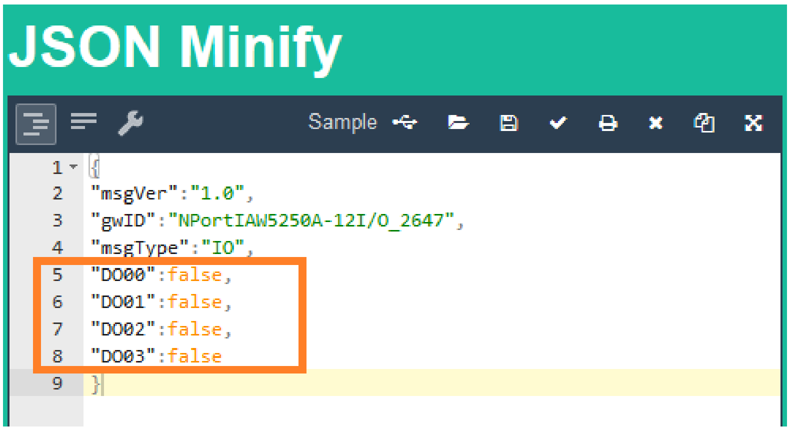


Subscribe JSON Message

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
}
```

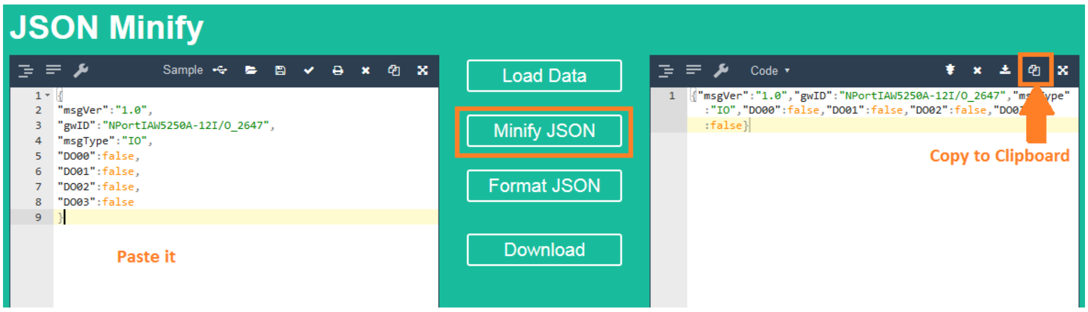
2. The copied message has a lot of space and line feed. We can use tool to compact it. Below is a free online tool: https://jsonformatter.org/json-minify Paste the message in the column on the left side and change all of the DO statuses to false.



JSON Minify

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

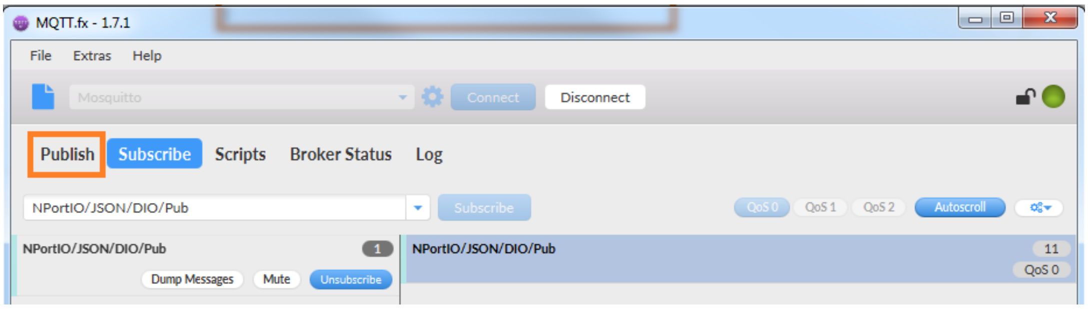
Click Minify JSON. It will show a compact JSON format message in the column on the right. Click **Copy to Clipboard**.



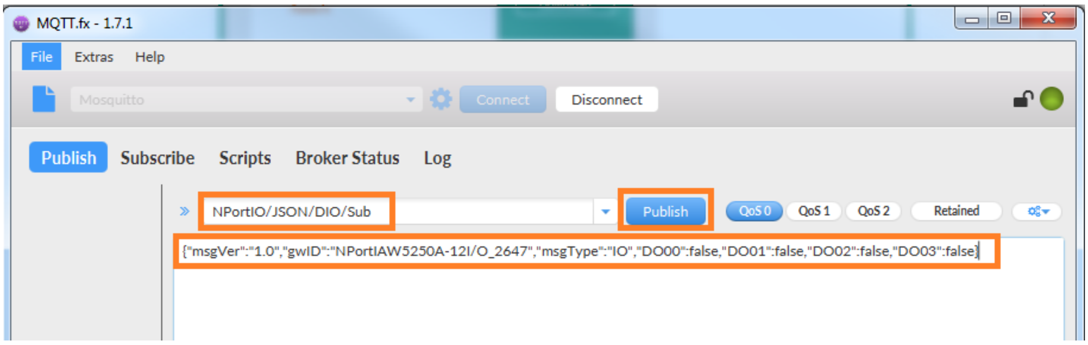
```
{
"msgVer":"1.0",
"gwID":"NPortIAW5250A-12I/O_2647",
"msgType":"IO",
"DO00":false,
"DO01":false,
"DO02":false,
"DO03":false
}
```

```
{"msgVer":"1.0","gwID":"NPortIAW5250A-12I/O_2647","ms   ype"
:"IO","DO00":false,"DO01":false,"DO02":false,"DO0
:false}
```

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



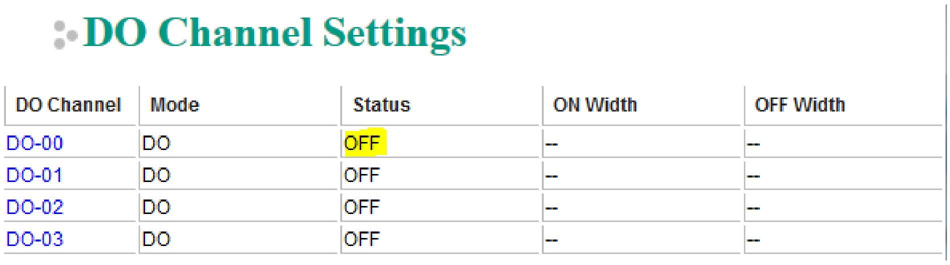
4. To publish a topic to the NPort: Under the **Publish** tap, paste the clipboard board message in big text box and fill in "NPortIO/JSON/DIO/Sub" as the topic string in the drop-down menu, and click **Publish**.



NPortIO/JSON/DIO/Sub

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

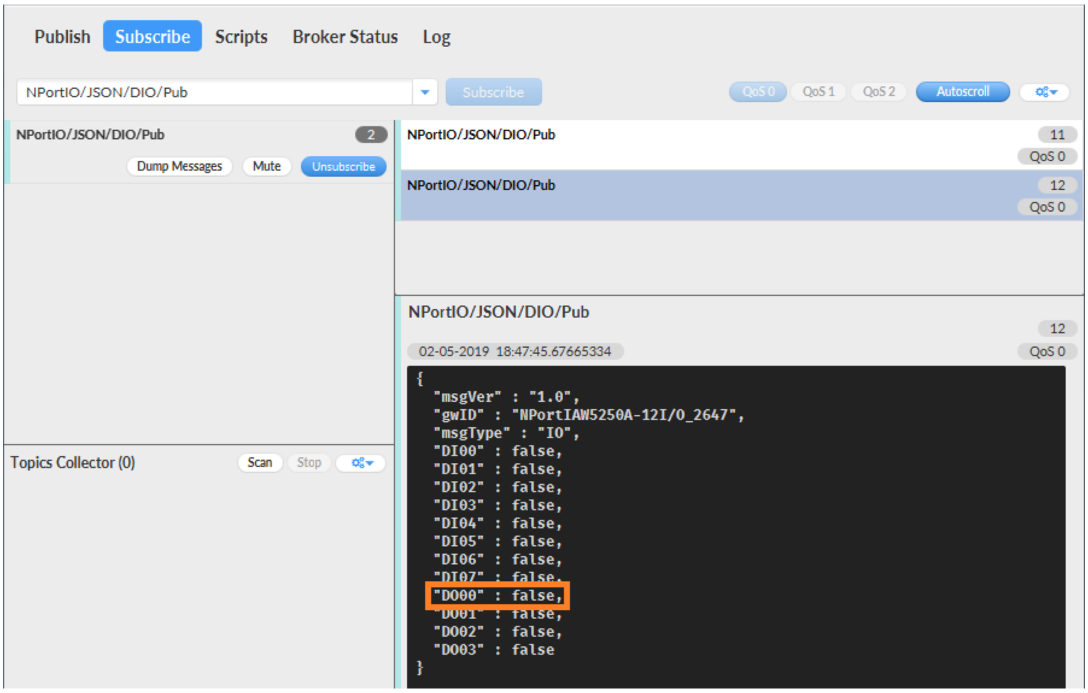
On the NPort's web console, check the DO-00 status as OFF.



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 will find the new message on the **Subscribe** page on MQTT.fx as we connect DO-00 to DI-00. This new message will show both the DI-00 DO-00 status as false.



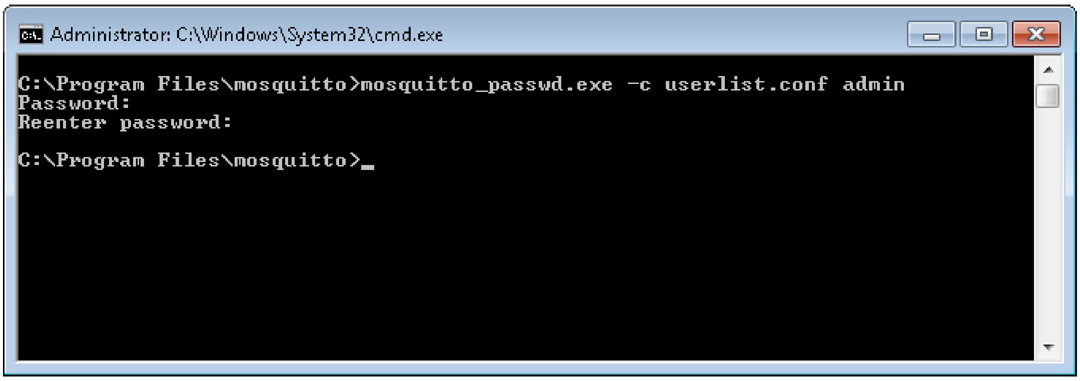
```
{
  "msgVer" : "1.0",
  "gwID" : "NPortIAW5250A-12I/O_2647",
  "msgType" : "IO",
  "DI00" : false,
  "DI01" : false,
  "DI02" : false,
  "DI03" : false,
  "DI04" : false,
  "DI05" : false,
  "DI06" : false,
  "DI07" : false,
  "DO00" : false,
  "DO01" : false,
  "DO02" : false,
  "DO03" : false
}
```

4. Customizing MQTT Broker Settings

If you need to customize the MQTT Broker settings, you can modify the mosquitto.conf file. The file is also under path C:\Program Files\mosquitto. The customize settings are below:

4.1 To access broker authenticate with username and password

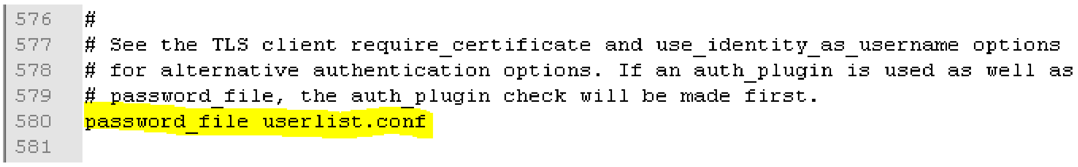
1. Create a new broker username and password, **admin** and moxa, respectively, in userlist.conf.
2. Use the command “**mosquitto_passwd.exe -c userlist.conf admin**”
3. Enter username’s password: **moxa**
4. Re-enter password: **moxa**



```
C:\Program Files\mosquitto>mosquitto_passwd.exe -c userlist.conf admin
Password:
Reenter password:

C:\Program Files\mosquitto>_
```

5. To set control access to the broker, use a password file to change “**#password_file**” to “**password_file userlist.conf”** on “Default authentication and topic access control” section in mosquitto.conf.

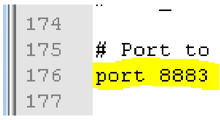


```
#
# See the TLS client require_certificate and use_identity_as_username options
# for alternative authentication options. If an auth_plugin is used as well as
# password_file, the auth_plugin check will be made first.
password_file userlist.conf

```

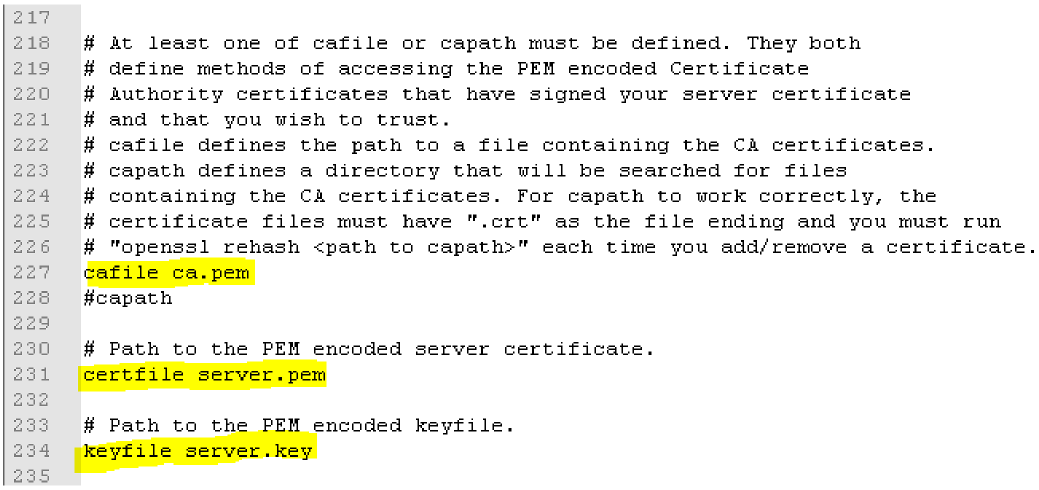
4.2 To Enable SSL/TLS Support (MQTT Over TLS)

1. To use openssl to generate CA/Server/Client certificates, please reference https://github.com/FreeRADIUS/freeradius-server/tree/master/raddb/certs
2. To set broker listen on TCP 8883 port, we change "#port" to "port 8883**"** on the "Default listener" section in mosquitto.conf



```
# Port to use for the default listener.
port 8883
```

3. Copy ca.pem, server.pem, and server.key under MQTT Broker folder. Change "**#cafile**" to "**cafile ca.pem**", "**#certfile**" to "**certfile server.pem**", "**#keyfile**" to "**keyfile server.key**" on the "Certificate based SSL/TLS support" section in mosquitto.conf.

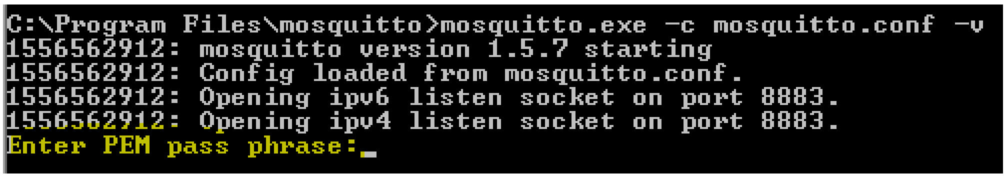


```
# At least one of cafile or capath must be defined. They both
# define methods of accessing the PEM encoded Certificate
# Authority certificates that have signed your server certificate
# and that you wish to trust.
# cafile defines the path to a file containing the CA certificates.
# capath defines a directory that will be searched for files
# containing the CA certificates. For capath to work correctly, the
# certificate files must have ".crt" as the file ending and you must run
# "openssl rehash <path to capath>" each time you add/remove a certificate.
cafile ca.pem
#capath

# Path to the PEM encoded server certificate.
certfile server.pem

# Path to the PEM encoded keyfile.
keyfile server.key
```

4. The MQTT Broker is running with TLS enabled. If your server keyfile has a passphrase, a prompt will pop up to ask you to input the passphrase as below:



```
C:\Program Files\mosquitto>mosquitto.exe -c mosquitto.conf -v
1556562912: mosquitto version 1.5.7 starting
1556562912: Config loaded from mosquitto.conf.
1556562912: Opening ipv6 listen socket on port 8883.
1556562912: Opening ipv4 listen socket on port 8883.
Enter PEM pass phrase:_
```

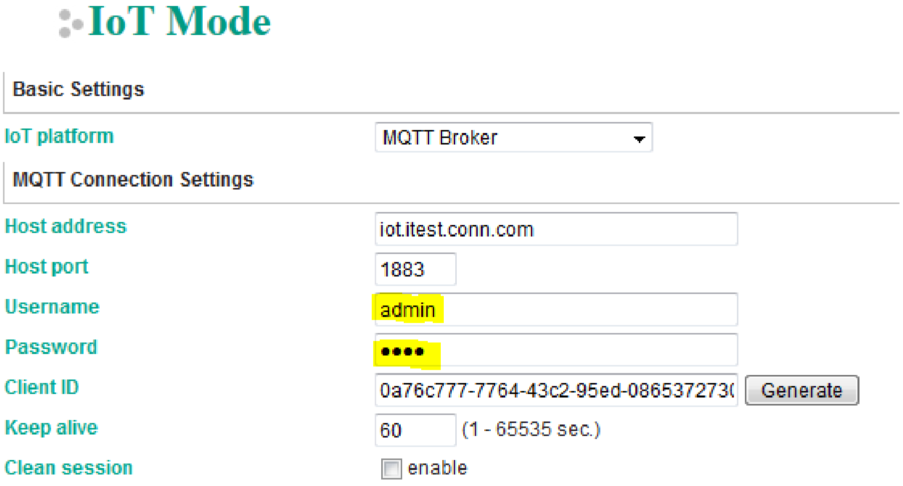
When MQTT Broker has enabled MQTT over TLS, please reference the "Optional NPort IA5000A-I/O and NPort IAW5000A-I/O settings" chapter for more about NPort IA5000A-I/O and NPort IAW5000A-I/O secure settings.

5. Optional NPort IA5000A-I/O or NPort IAW5000A-I/O Settings

If you have customize MQTT Broker settings, you need to modify the Nport’s configuration.

5.1 To Connect to Secure MQTT Broker With Authenticate

When you assess MQTT Broker to authenticate, you need to fill in the correct username and password under **MQTT Connection Settings**.



IoT Mode

Basic Settings

IoT platform	MQTT Broker

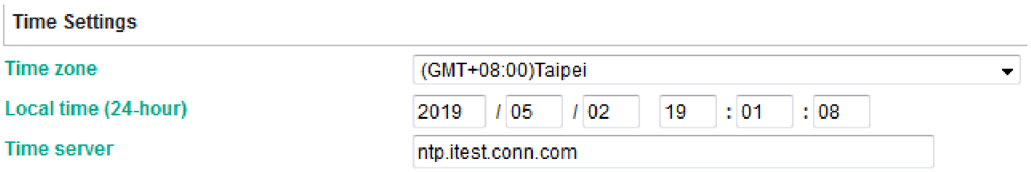
MQTT Connection Settings

Host address	iot.itest.conn.com
Host port	1883
Username	admin
Password	●●●●
Client ID	0a76c777-7764-43c2-95ed-086537273 Generate
Keep alive	60 (1 - 65535 sec.)
Clean session	□ enable

5.2 To Connect to Secure MQTT Broker With MQTT Over TLS

The NPorts support TLS to secure communications between MQTT Broker and Client. Here, we use version 1.2.

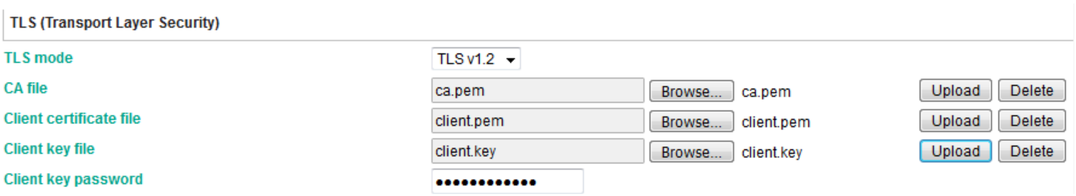
1. Set the correct time setting on the **Basic Settings** page of the NPorts, and fill in the time server to correct the NPorts’ time-on period.



Time Settings

Time zone	(GMT+08:00)Taipei
Local time (24-hour)	2019 / 05 / 02 19 : 01 : 08
Time server	ntp.itest.conn.com

2. To enable TLS transmission, set **TLS mode** to “TLS v1.2”. Upload the CA certificate, client certificate, and client key file. The certificates and key file must be PEM encoded. If your key file has a passphrase, fill in the correct passphrase when uploading the key file as below:



TLS (Transport Layer Security)

TLS mode: TLS v1.2

CA file: ca.pem — ca.pem

Client certificate file: client.pem — client.pem

Client key file: client.key — client.key

Client key password: ••••••••••••