

# NR500 Series Industrial Cellular VPN Router

## Application Note 067

### Modbus Sparkplug B

**Version:** V1.0.0  
**Date:** Apr 2023  
**Status:** Confidential



## Directory

1. Introduction .....	3
1.1 Overview .....	3
1.2 Compatibility .....	3
1.3 Version .....	3
1.4 Corrections .....	3
2. Topology .....	4
3. Configuration .....	5
3.1 Configuration on Modbus Slave .....	5
3.2 Configuration on Modbus Poll .....	6
3.3. Configuration on Modbus to Sparkplug B .....	8
4. Testing .....	10

# 1. Introduction

## 1.1 Overview

This document contains information regarding the configuration and use of Modbus Sparkplug B.

This guide has been written for use by technically competent personnel with a good understanding of the communications technologies used in the product, and of the requirements for their specific application.

## 1.2 Compatibility

This application note applies to:

**Models Shown:** NR500 series.

**Firmware Version:** V3.0.0 (1f18909-1) or newer

**Other Compatible Models:** None

## 1.3 Version

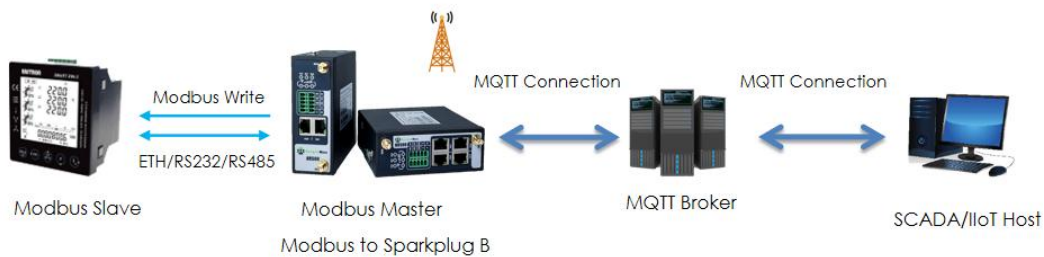
Updates between document versions are cumulative. Therefore, the latest document will include all the content of previous versions.

Release Date	Doc. Version	Firmware Version	Change Description
2023/04/12	V1.0.0	V3.0.0 (1f18909-1)	First released

## 1.4 Corrections

Appreciate for corrections or rectifications to this application note, and if any request for new application notes please email to: [support@navigateworx.com](mailto:support@navigateworx.com)

## 2. Topology



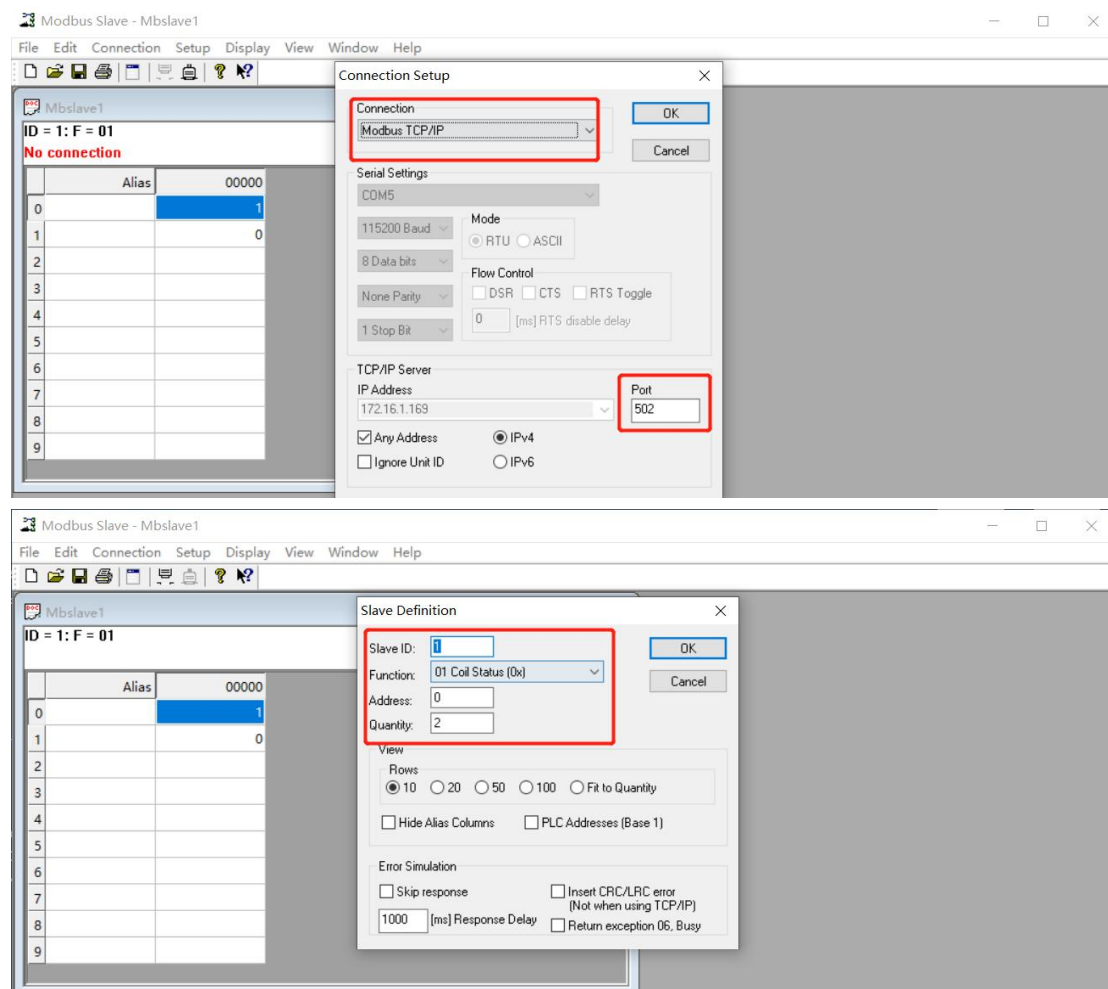
1. NR500 Router runs as Modbus Master and send Modbus Sparkplug B data using MQTT, it acts as Modbus Master and MQTT EoN Node.
2. A serial device support Modbus protocol and acts as Modbus Slave, NR500 router poll the Modbus data from end device (Modbus Slave).
3. NR500 router run as MQTT EoN Node, it can convert Modbus data or the status of Local-IO to sparkplug B data to MQTT broker via MQTT protocol.
4. The SCADA/IloT Host can subscribe the specified topic from the MQTT broker to read the register value/Local-IO of NR500 router or send NCMD/DCMD to write the register value/Local-IO of NR500 router.

*Note: For this Application Note, we will set Modbus Master's Connection Type as "TCP" as an example, which means that NR500(Modbus Master) will connect to the Modbus Slave and read the value via Ethernet port. Of course, it also works with Serial Port (RS232/RS485).*

## 3. Configuration

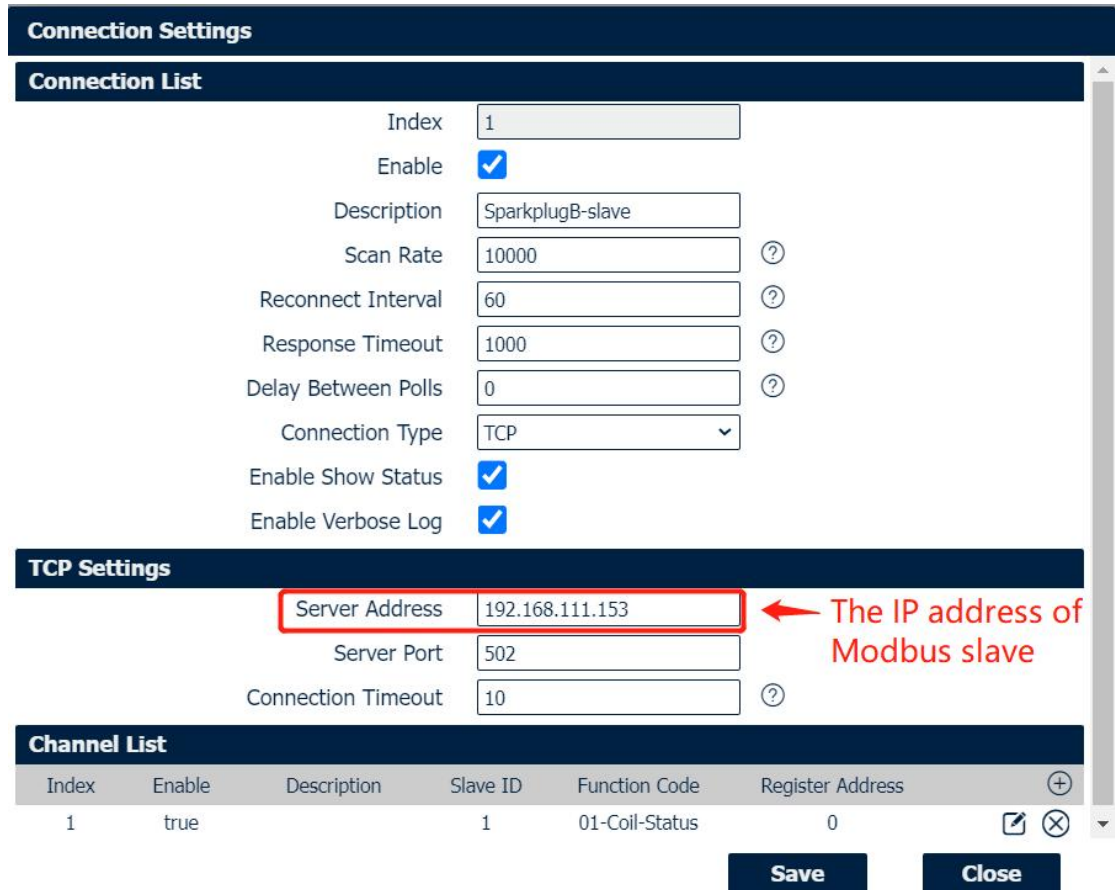
### 3.1 Configuration on Modbus Slave

- Here we use “Modbus Slave” software to simulate the end device (Modbus Slave device), and the **TCP Port: 502, Slave ID: 1, Function Code: 01-Coil Status**, like below settings:



### 3.2 Configuration on Modbus Poll

1. Go to **Application>Modbus Master>Modbus Poll**, add a "Connection List" and specify the "Connection Type" as "TCP", specify the "TCP Settings" to connect to Modbus Slave, like below:



The screenshot shows the configuration interface for Modbus Poll. It is divided into three main sections: Connection List, TCP Settings, and Channel List.

**Connection List**

- Index: 1
- Enable:
- Description: SparkplugB-slave
- Scan Rate: 10000
- Reconnect Interval: 60
- Response Timeout: 1000
- Delay Between Polls: 0
- Connection Type: TCP
- Enable Show Status:
- Enable Verbose Log:

**TCP Settings**

- Server Address: 192.168.111.153 (highlighted with a red box and an arrow pointing to it with the text "The IP address of Modbus slave")
- Server Port: 502
- Connection Timeout: 10

**Channel List**

Index	Enable	Description	Slave ID	Function Code	Register Address
1	true		1	01-Coil-Status	0

At the bottom of the interface, there are two buttons: **Save** and **Close**.

2. Click Save.
3. Enable "Channel List", and specify the Slave ID as " 1 ", Function Code as " **01-Coil-status**", Register Address to " 0 ", then it will poll the value from register address 0 of Modbus Slave:

**Channel Settings**

**Channel List**

Index	<input type="text" value="1"/>
Enable	<input checked="" type="checkbox"/>
Description	<input type="text"/>
Slave ID	<input type="text" value="1"/>
Function Code	<input type="text" value="01-Coil-Status"/>
Register Address	<input type="text" value="0"/>
Data type	<input type="text" value="Bool"/>
Multiple Register	<input type="checkbox"/>

Save
Close

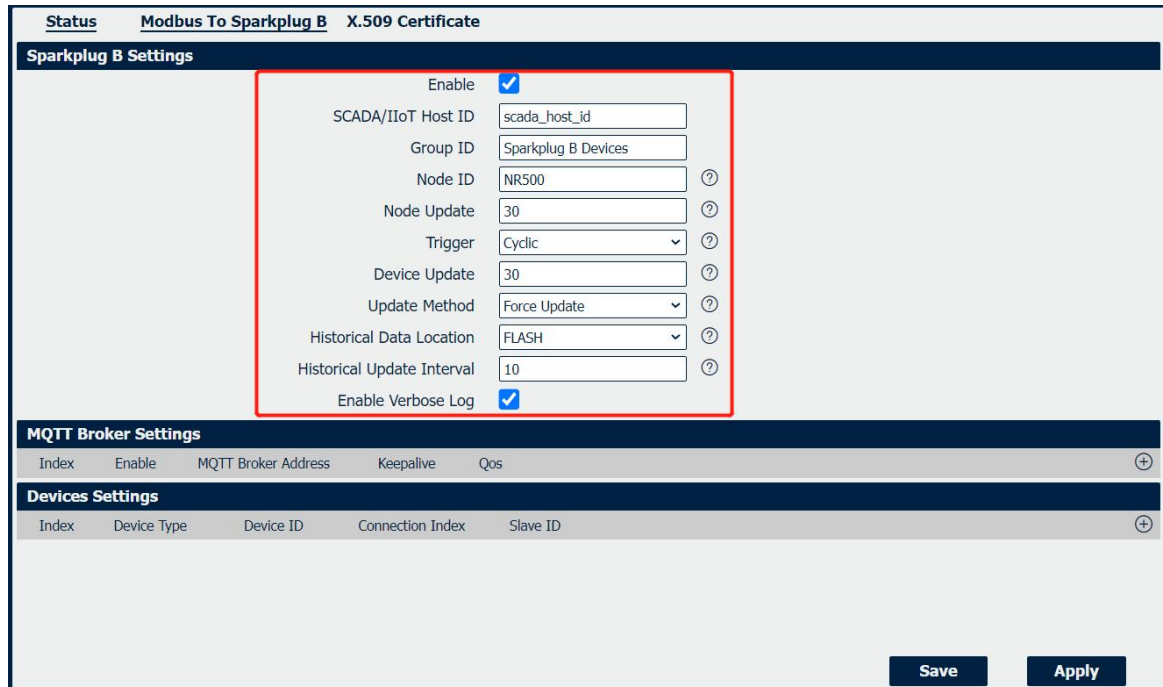
4. Click Save>Save>Apply. *(Note: This is a secondary list, it needs to double click save)*

5. Go to **Application>Modbus Master>Status**, then we can check the router had read the value from Modbus Slave successfully.

Overview	Status	Modbus Poll	Modbus Alarm	Modbus Write					
Link Management Industrial Interface Network Applications DDNS SMS Schedule Reboot GPS Call	<b>Channel Status</b>								
	Index	Description	Connection Index	Type	Slave ID	Register Address	Function Code	Status	Value
	1		1	TCP	1	0	1	Read successfully	1

### 3.3. Configuration on Modbus to Sparkplug B

1. Go to **Application>Modbus Sparkplug>Modbus To Sparkplug B**, click enable button and specify the SCADA/IIoT Host ID, Group ID and Node ID.



The screenshot shows the configuration page for 'Modbus To Sparkplug B'. The 'Sparkplug B Settings' section is highlighted with a red box. It contains the following fields and values:

- Enable:
- SCADA/IIoT Host ID:
- Group ID:
- Node ID:  ?
- Node Update:  ?
- Trigger:  ?
- Device Update:  ?
- Update Method:  ?
- Historical Data Location:  ?
- Historical Update Interval:  ?
- Enable Verbose Log:

Below this section are 'MQTT Broker Settings' and 'Devices Settings' sections, each with a table header and a '+' icon for expansion. At the bottom right, there are 'Save' and 'Apply' buttons.

Note:

1. SCADA/IIoT Host ID: NR500 subscribes the specified topic to get the status of SCADA/IIoT Host.

The format of the specified topic: STATE/[SCADA/IIoT Host ID],

Qos: the Qos of the MQTT broker,

Content: ONLINE/OFFLINE.

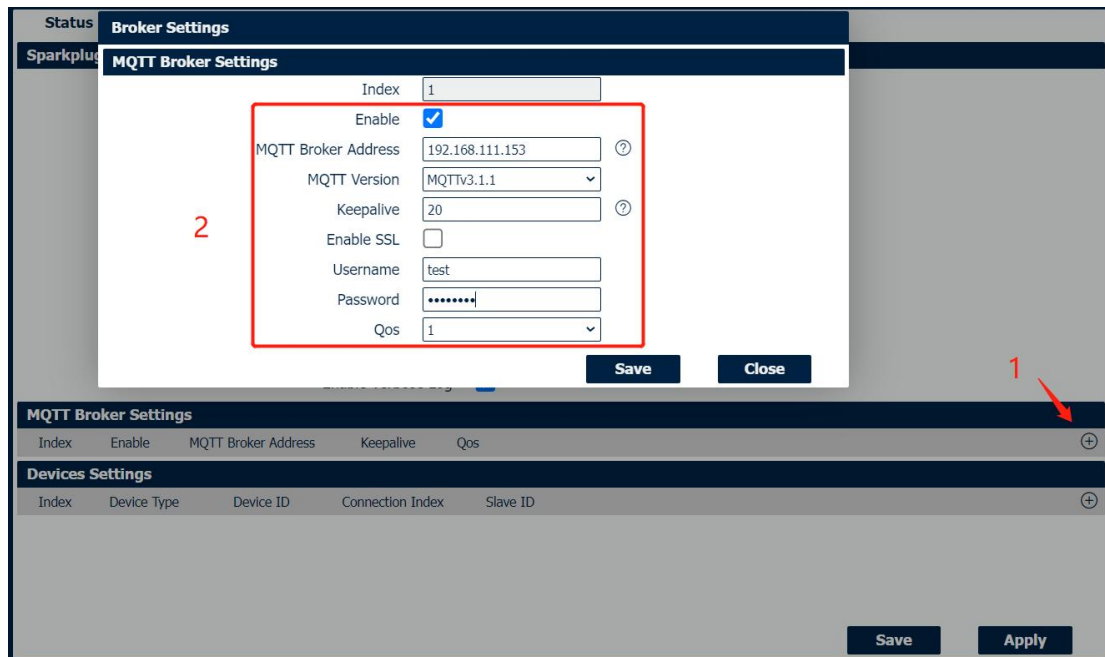
2. Node updates operates on NDATA message type.
3. Device updates operates on DDATA message type.
4. Trigger only operates on Device update.

There are two update methods, [SCADA/IIoT Host Online] and [Force update]. Under SCADA/IIoT Host online Method, Modbus to Sparkplug B only will publish the data to MQTT broker after received the SCADA/IIoT Host ONLINE message. Under Force update method, Modbus to Sparkplug B will force publish the data to MQTT broker even SCADA/IIoT Host OFFLINE.

2. Click Save > Apply.



3. Enable "MQTT Broker settings", click enable button and specify the MQTT Broker Address, select MQTT Version and config Username and Password.



MQTT Broker Settings

Index: 1

Enable:

MQTT Broker Address: 192.168.111.153

MQTT Version: MQTTv3.1.1

Keepalive: 20

Enable SSL:

Username: test

Password: .....

Qos: 1

MQTT Broker Settings Table:

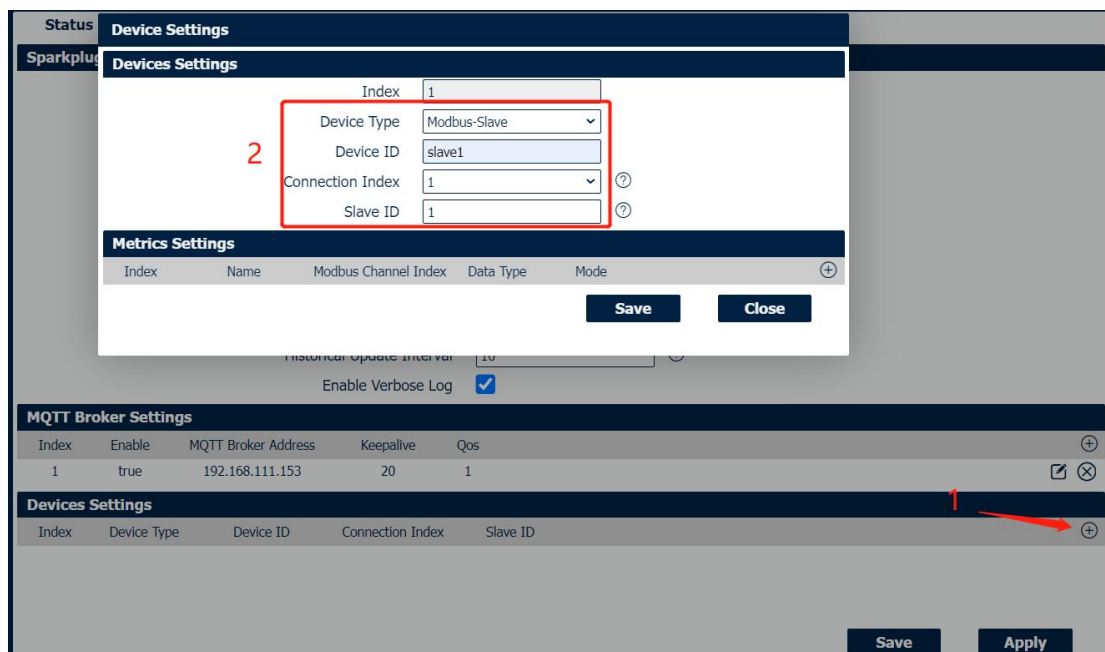
Index	Enable	MQTT Broker Address	Keepalive	Qos
1	true	192.168.111.153	20	1

Devices Settings Table:

Index	Device Type	Device ID	Connection Index	Slave ID

4. Click Save>Save>Apply.

5. Enable "Devices settings", click enable button and select the Device Type, enter Device ID and select Connection Index and Slave ID.



Device Settings

Index: 1

Device Type: Modbus-Slave

Device ID: slave1

Connection Index: 1

Slave ID: 1

Metrics Settings Table:

Index	Name	Modbus Channel Index	Data Type	Mode

MQTT Broker Settings Table:

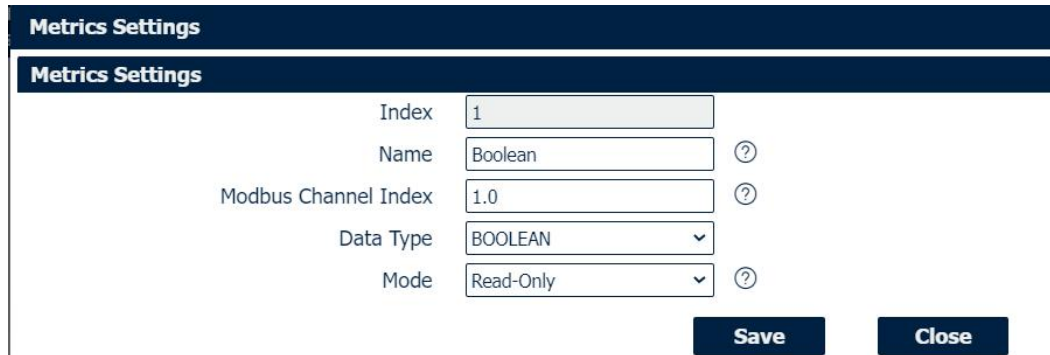
Index	Enable	MQTT Broker Address	Keepalive	Qos
1	true	192.168.111.153	20	1

Devices Settings Table:

Index	Device Type	Device ID	Connection Index	Slave ID

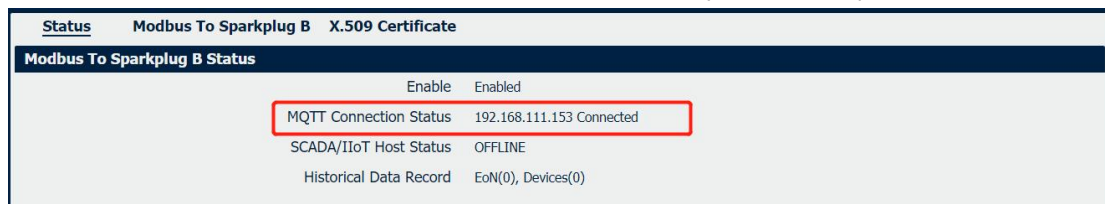
Note: When Device Type is Modbus-Slave, NR500 will convert modbus data to sparkplug B data and publish it to remote MQTT broker. If Device Type is Local-IO, NR500 will convert the status of local IO to the sparkplug B data and publish it to remote MQTT broker.

6. Enable “Metrics Settings”, enter Name and Modbus Channel Index, select Data Type and Mode.



Note: The Format of Modbus Channel Index: X.Y;  
[X]:Modbus poll channel index, [Y]:Modbus register address.

7. Click Save>Save>Apply.
8. Go to **Application>Modbus Sparkplug B>Status**, NR500(Modbus to Sparkplug B) had connected to the remote server successfully via MQTT protocol.



## 4. Testing

1. Run the MQTT Client (MQTT.fx), to subscribe the topic, then we can get the sparkplug B data successfully.

