How to Build a Real-Time IIoT Application with Moxa's ThingsPro and Cirrus Link's Sparkplug

Moxa Technical Support Team
support@moxa.com

Contents

1 Introduction 2
2 Application Scenario 2
3 Prerequisites 3
4 Solution 3
4.1 Set up ThingsPro Gateway software to read Modbus data from your device 3
4.2 Enable Sparkplug MQTT communication for your device 8
4.3 Use Ignition Designer to monitor & control device data 11
5 Additional reading 12

Released on November 30, 2017

About Moxa

Moxa is a leading provider of edge connectivity, industrial computing, 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

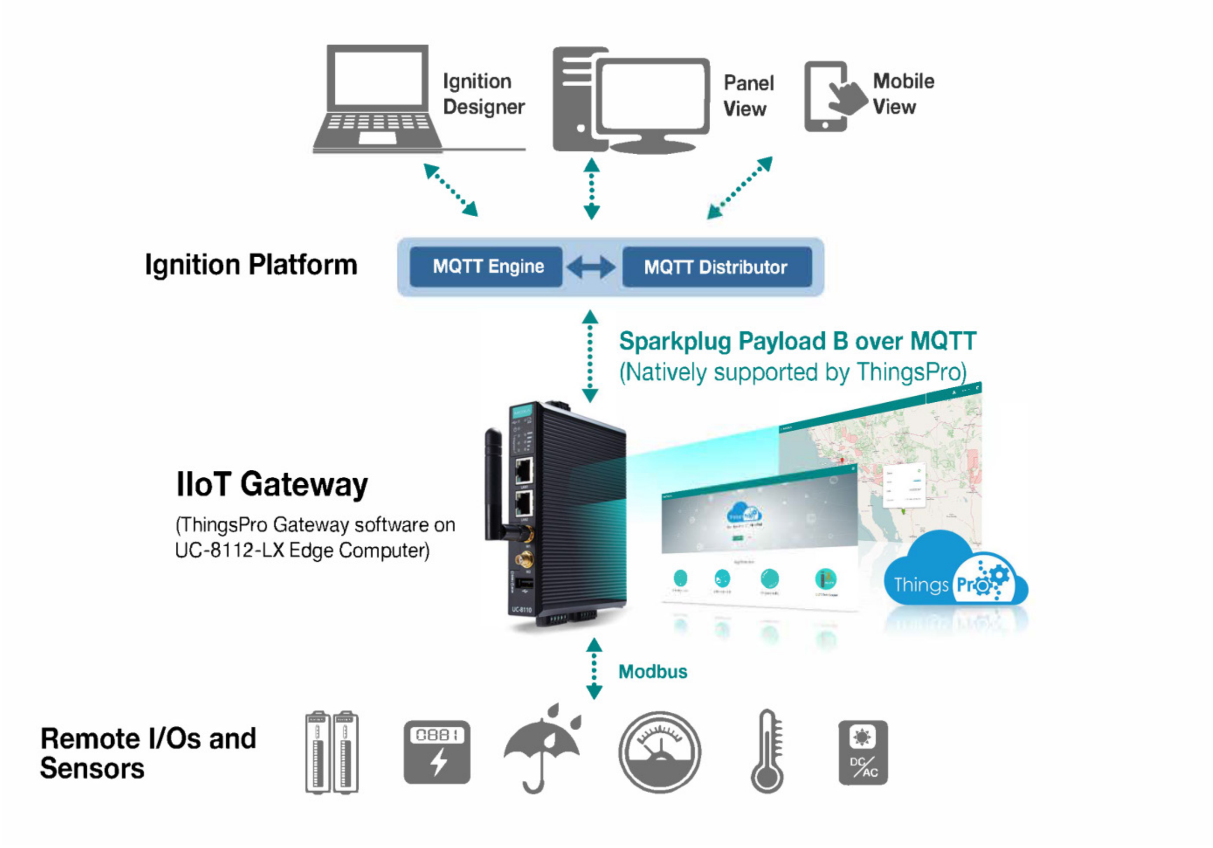
MOXA®
Reliable Networks ▲ Sincere Service

1 Introduction

Moxa's ThingsPro™ 2 Suite simplifies the development of IIoT applications, facilitates data acquisition, and enables better remote device management. Using ThingsPro, users can configure commonly used functions such as Modbus communication, data acquisition, wireless networking, and device management in just a few steps. This allows users to focus on developing applications instead of on complex system integration.

2 Application Scenario

For users who want to use Inductive Automation's Ignition platform, ThingsPro provides the ability to transfer field data to the Ignition Server without requiring any additional programming on the gateway. Users can use this to develop software programs for data acquisition and asset management on their IIoT gateway. ThingsPro 2.0 includes a Sparkplug client that supports *Sparkplug Payload B*, which enables users to easily configure Modbus data polling tables and configure Ignition communication settings. Users can then upload the data collected from their devices to an Ignition server at the back end.



3 Prerequisites

1. ThingsPro Gateway software, installed on an eligible Moxa device, such as a UC-8112-LX (For information about setting up a UC-8112 computer with ThingsPro, refer to the *ThingsPro Gateway Quick Installation Guide.)*
2. A PC/notebook, with Chrome browser installed, for accessing the web interface of the ThingsPro 2 Gateway software on the UC-8112 computer
3. A Modbus device (e.g., Moxa's ioLogik-E2242)
4. A working Ignition Platform (see note below) with Cirrus Link's MQTT Distributor and MQTT Engine installed

Note: The instructions in this document assume that you are familiar with Inductive Automation's Ignition platform. For additional details visit: https://inductiveautomation.com/solutions/iiot and http://pages.inductiveautomation.com/rs/966-TUX-261/images/ProductSheet_MQTT-Modules.pdf

4 Solution

Take the following steps to configure the Modbus device template in ThingsPro, to retrieve device data from ThingsPro, and to upload the data to the Ignition Server:

- Set up ThingsPro Gateway software to read Modbus data from your device
- Enable Sparkplug MQTT communication for your device
- Use Ignition Designer to monitor & control device data

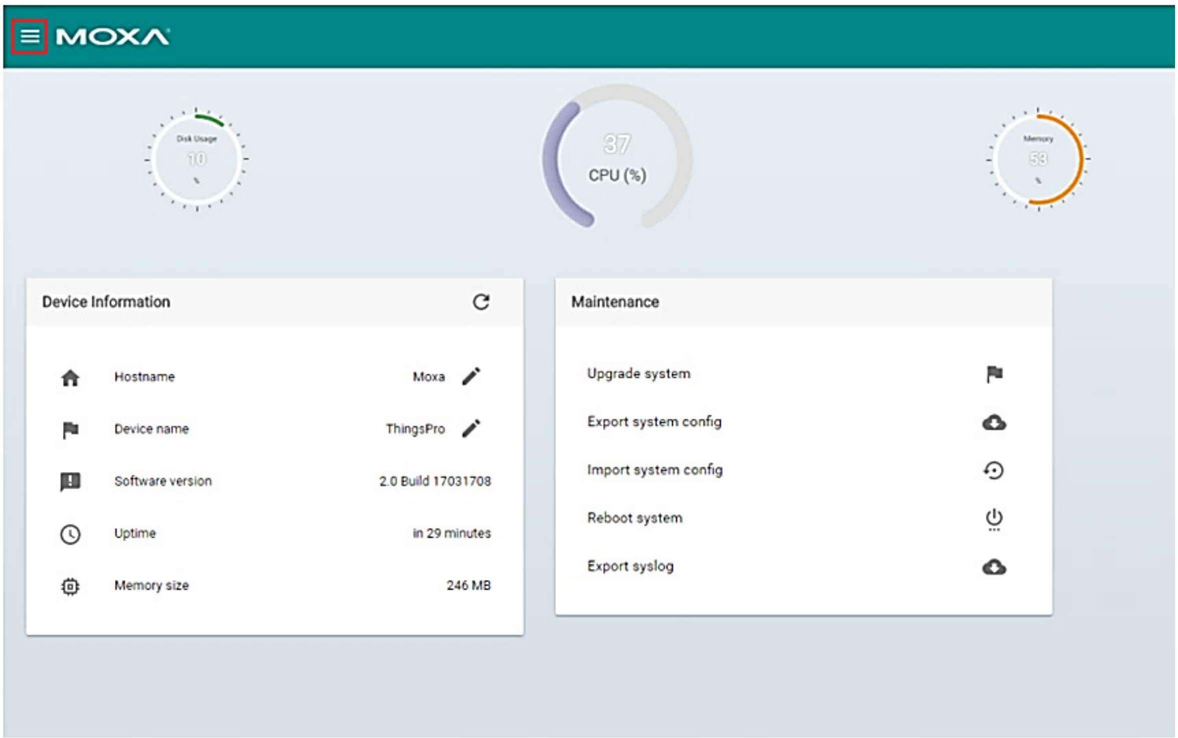
4.1 Set up ThingsPro Gateway software to read Modbus data from your device

You must first connect your device to the UC-8112 Series gateway computer. In our example, the device we use is a Moxa ioLogik-E1242.

To establish a connection between the ThingsPro Gateway software and the ioLogik-E1242 device, do the following:

1. **Choose an equipment template**

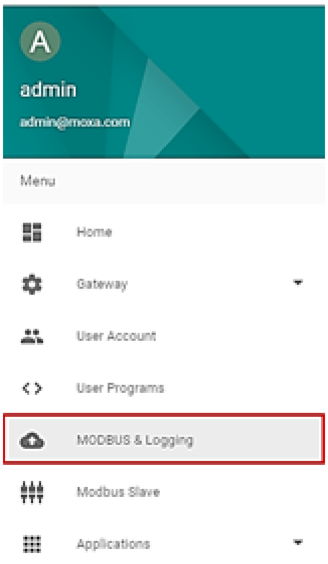
   a. In the ThingsPro Gateway web UI, click on the ☰ (Menu) icon.



Device Information

Hostname	Moxa
Device name	ThingsPro
Software version	2.0 Build 17031708
Uptime	in 29 minutes
Memory size	246 MB

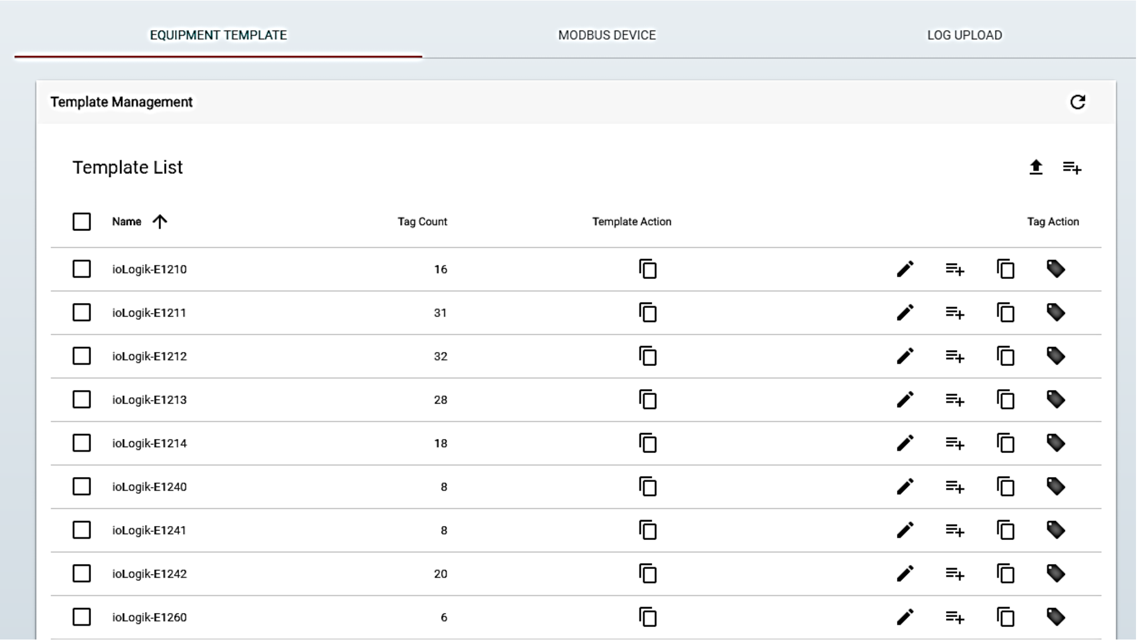
   b. In the ThingsPro Gateway menu, click on the **MODBUS & Logging** link.



c. Open the **EQUIPMENT TEMPLATE** tab.

d. Browse through the **Template List** to locate the template that corresponds with your Modbus device.

If the template for your device is not in the list, then create a new template. For details on creating a new template, refer to the *ThingsPro Software Suite User's Manual*.



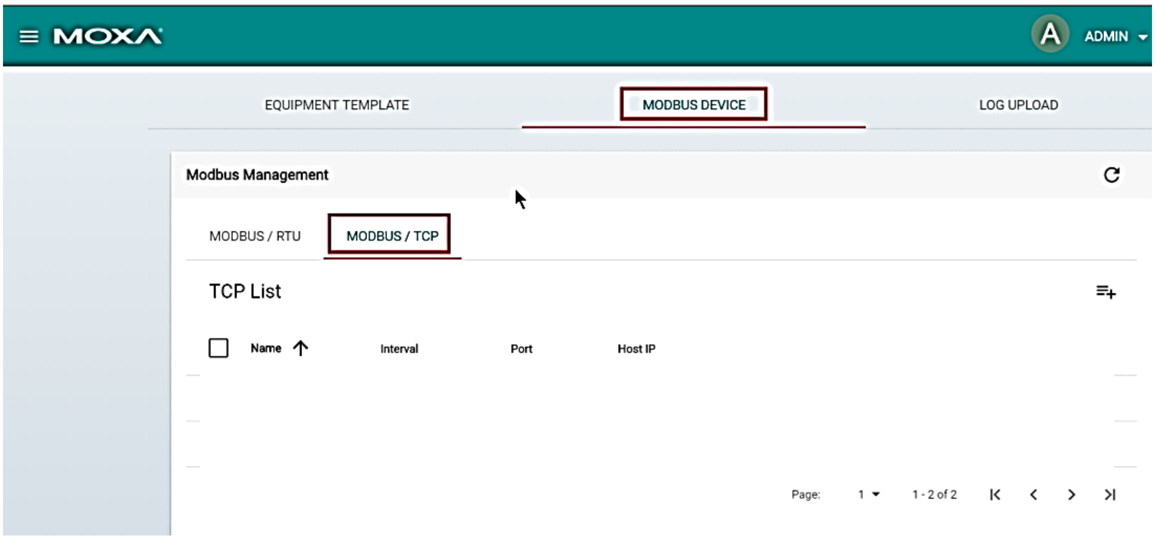
□	Name ↑	Tag Count	Template Action	Tag Action
□	ioLogik-E1210	16	⧉	✎ ≡+ ⧉ 🏷
□	ioLogik-E1211	31	⧉	✎ ≡+ ⧉ 🏷
□	ioLogik-E1212	32	⧉	✎ ≡+ ⧉ 🏷
□	ioLogik-E1213	28	⧉	✎ ≡+ ⧉ 🏷
□	ioLogik-E1214	18	⧉	✎ ≡+ ⧉ 🏷
□	ioLogik-E1240	8	⧉	✎ ≡+ ⧉ 🏷
□	ioLogik-E1241	8	⧉	✎ ≡+ ⧉ 🏷
□	ioLogik-E1242	20	⧉	✎ ≡+ ⧉ 🏷
□	ioLogik-E1260	6	⧉	✎ ≡+ ⧉ 🏷

In this example, we use the Modbus device template **E2242**, which is already available in the **Template List**.

2. Create an instance of the Modbus device

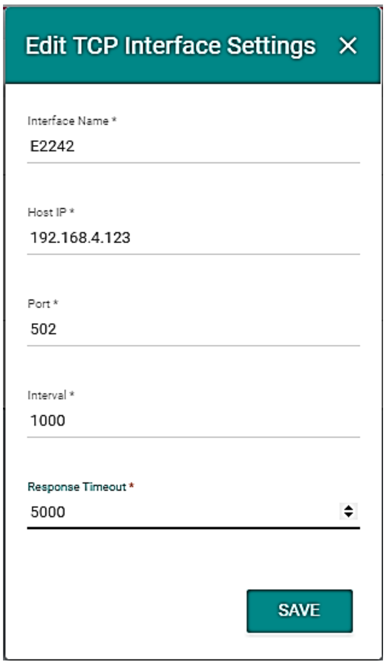
a. Open the **MODBUS DEVICE** tab.

b. Configure the Modbus device with a communication interface (serial or Ethernet). Because the connected device (E2242) uses Modbus TCP protocol, click on the **MODBUS/TCP** tab.



i. Click on the ≡+ (Add TCP interface) icon.

ii. Fill in the TCP interface settings and click **SAVE**.



Interface Name * E2242

Host IP * 192.168.4.123

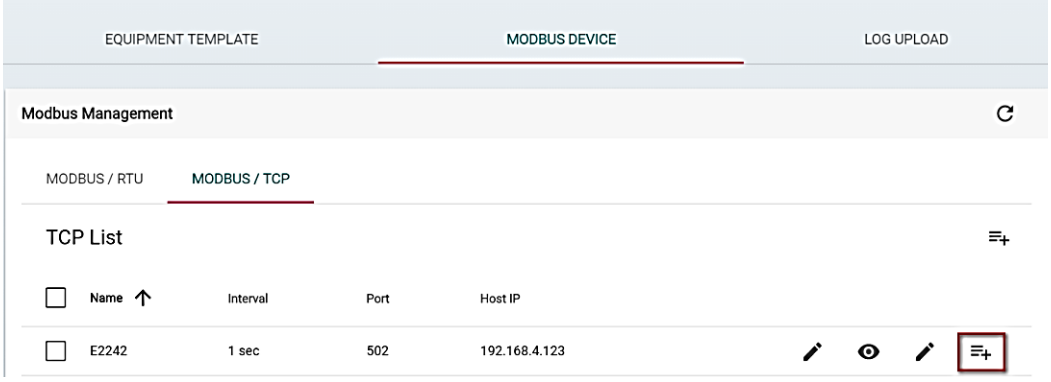
Port * 502

Interval * 1000

Response Timeout * 5000

The Modbus/TCP interface for E2242 will be added to the **TCP List**.

iii. Click on the ≡+ (Add connected device) icon for the E2242 to add a connected device.



Modbus Management

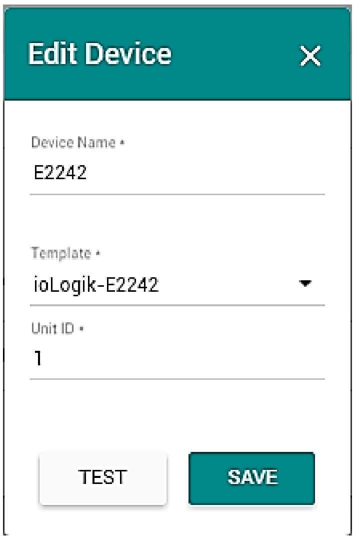
TCP List

□	Name ↑	Interval	Port	Host IP
□	E2242	1 sec	502	192.168.4.123

iv. Fill in the **Device Name**.

v. Select a **Template** to combine the Modbus/TCP interface with the Modbus device.

vi. Specify the **Unit ID** for the device and click **SAVE**.



Edit Device

Device Name *
E2242

Template *
ioLogik-E2242

Unit ID *
1

ThingsPro Gateway will now start continuously polling data from the ioLogik-E2242 device.

To check the status, click on the 👁 (Show connected devices) icon.

If the **Request Status** is green, it indicates that ThingsPro is able to send requests to the Modbus device and receive data correctly.

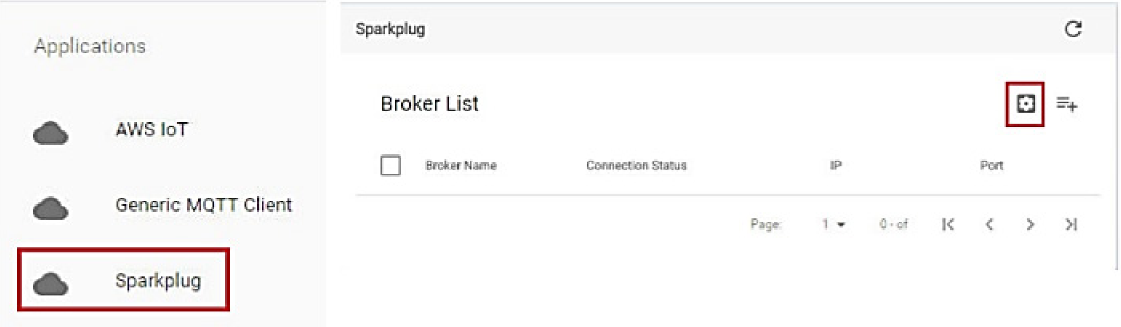
4.2 Enable Sparkplug MQTT communication for your device

Sparkplug is a specification for MQTT-enabled devices and applications that allows you to send and receive messages in a stateful way. Sparkplug also provides a mechanism to ensure that the messages from remote devices or applications are current and valid. ThingsPro provides an interface to enable Sparkplug-based communication between servers, gateways, and edge devices.

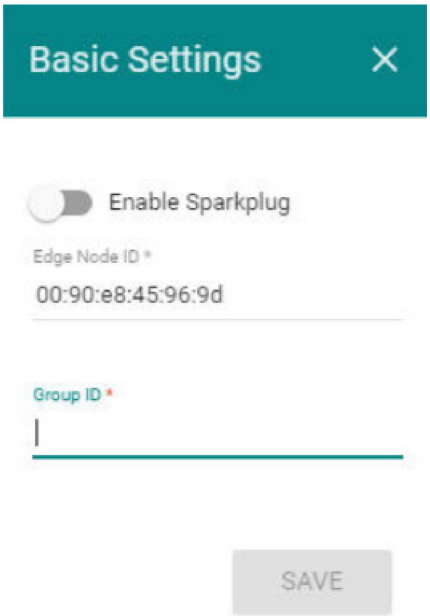
a. Enable the SparkPlug client in ThingsPro Gateway

To enable the Sparkplug interface in ThingsPro Gateway, do the following:

1. Click on the **Sparkplug** link in the **Applications** section of the main menu.
2. Click on the ⚙ (Basic Settings) icon.



3. Click on **Enable Sparkplug** to activate the function.
   The **Edge Node ID** of the device will be automatically retrieved and displayed.
4. Enter the **Group ID.**



Edge Node ID *
00:90:e8:45:96:9d

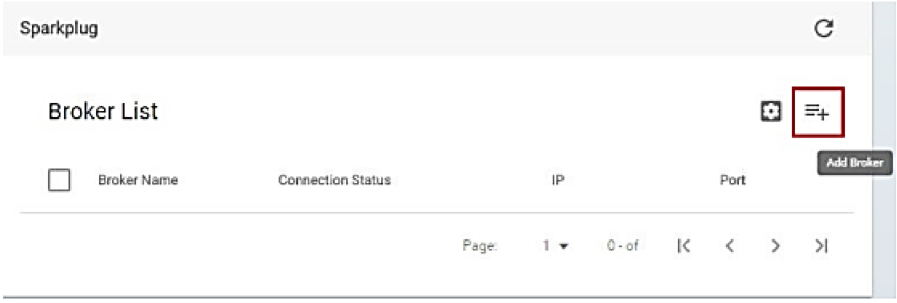
Group ID *

5. Click **SAVE**.

b. Add a Message Broker for Sparkplug

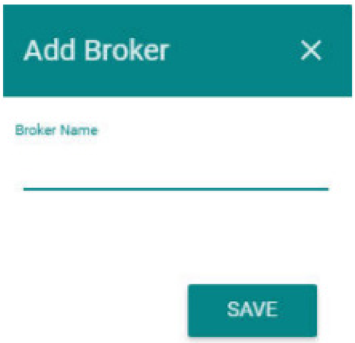
You must create a message broker in the ThingsPro Sparkplug interface. To create a message broker for Sparkplug, do the following:

1. In the Sparkplug interface, click on the ≡+ (Add Broker) icon.



□	Broker Name	Connection Status	IP	Port

2. Type in the **Broker Name**.

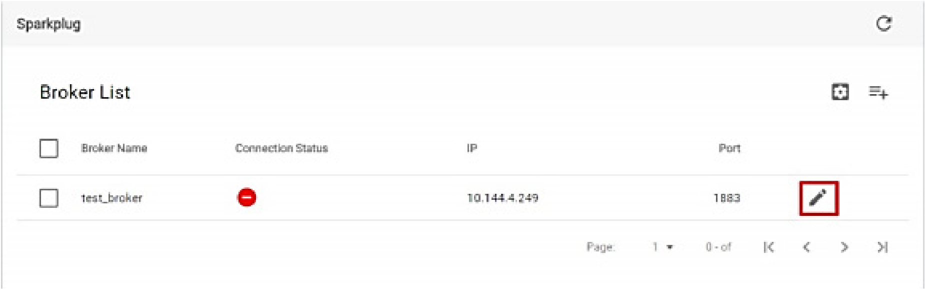


3. Click **SAVE.**

c. Configure Broker settings for Sparkplug

To configure/modify the Sparkplug broker settings, do the following:

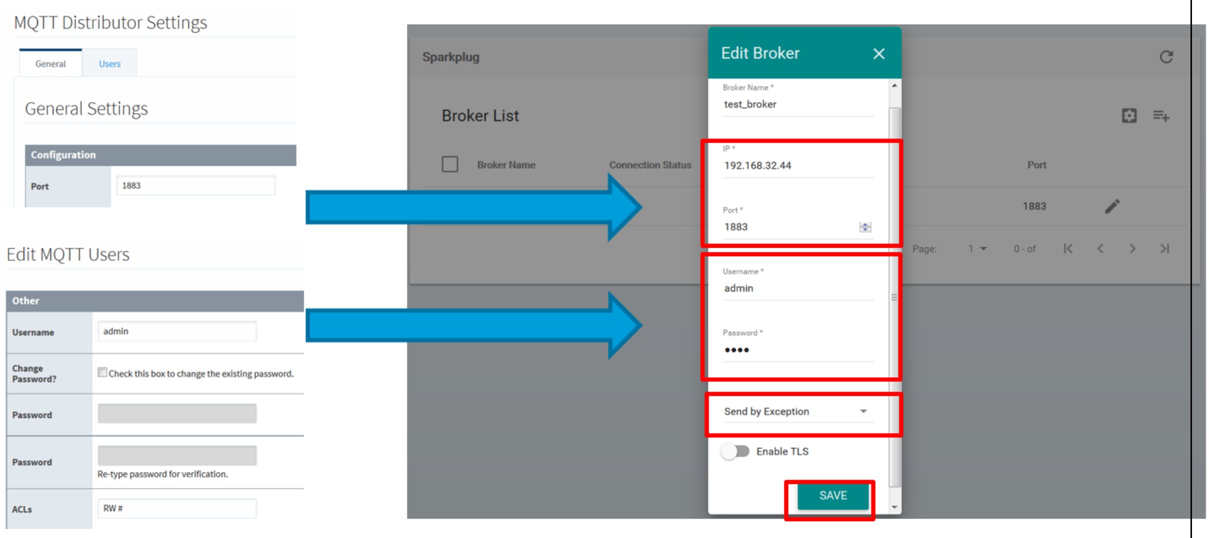
1. In the Sparkplug configuration page, click on the ✎ (Edit Broker) icon.



□	Broker Name	Connection Status	IP	Port	
□	test_broker	⛔	10.144.4.249	1883	✎

2. Enter the broker configuration details.

   You can get the broker configuration details from the Ignition platform. Some broker configuration details are available from the MQTT Distributor while the other details are available from the MQTT Engine, as shown below:



MQTT Distributor Settings

General | Users

General Settings

Configuration	
Port	1883

Edit MQTT Users

Other	
Username	admin
Change Password?	□ Check this box to change the existing password.
Password	
Password	Re-type password for verification.
ACLs	RW #

Sparkplug

Broker List

□	Broker Name	Connection Status	Port	
			1883	

Page: 1 0 - of

Edit Broker

Broker Name *: test_broker

IP *: 192.168.32.44

Port *: 1883

Username *: admin

Password *: ●●●●

Send by Exception

○ Enable TLS

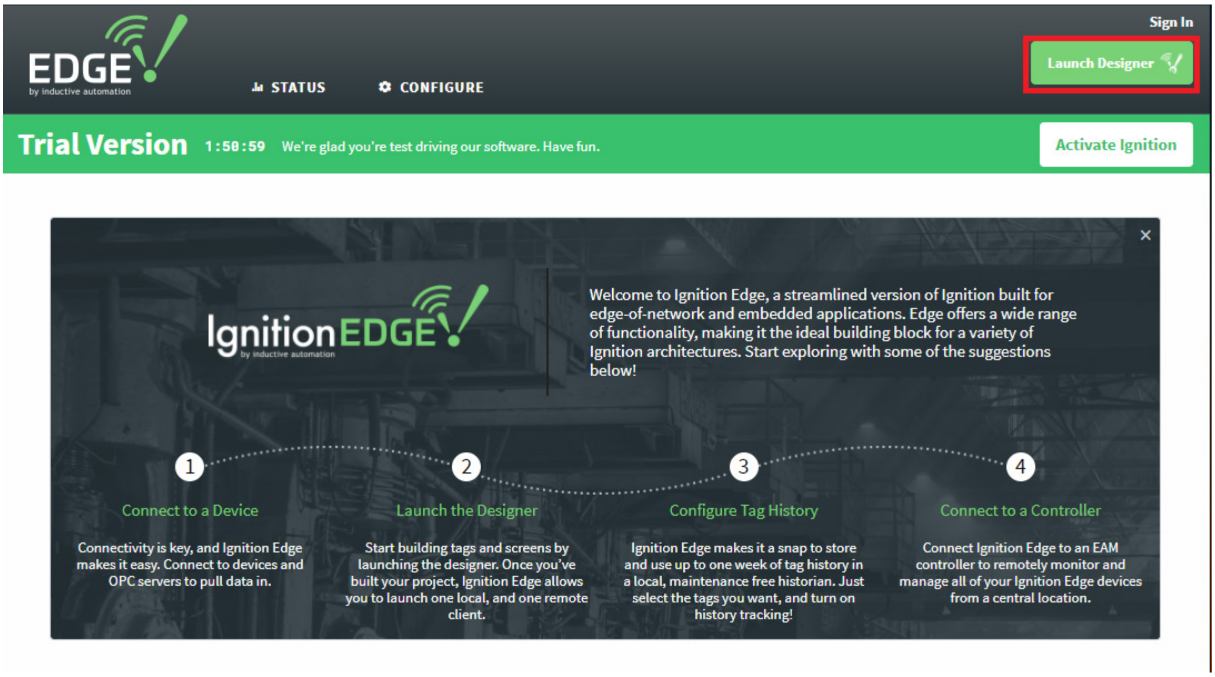
SAVE

If necessary, enable TLS and fill in the certificate information for the TLS connection.

3. Click **SAVE**.

4.3 Use Ignition Designer to monitor & control device data

You can use the Ignition Designer software to monitor and control your device data as shown below:



Trial Version 1:50:59 We're glad you're test driving our software. Have fun.

Welcome to Ignition Edge, a streamlined version of Ignition built for edge-of-network and embedded applications. Edge offers a wide range of functionality, making it the ideal building block for a variety of Ignition architectures. Start exploring with some of the suggestions below!

1 Connect to a Device

Connectivity is key, and Ignition Edge makes it easy. Connect to devices and OPC servers to pull data in.

2 Launch the Designer

Start building tags and screens by launching the designer. Once you've built your project, Ignition Edge allows you to launch one local, and one remote client.

3 Configure Tag History

Ignition Edge makes it a snap to store and use up to one week of tag history in a local, maintenance free historian. Just select the tags you want, and turn on history tracking!

4 Connect to a Controller

Connect Ignition Edge to an EAM controller to remotely monitor and manage all of your Ignition Edge devices from a central location.

5 Additional reading

- Cirrus Link IIoT modules:
  https://inductiveautomation.com/ignition/modules#cirrus-link-solutions
- Sparkplug—MQTT Topic & Payload:
  http://www.cirrus-link.com/oem-device-data-integration/
- Cirrus Link documentation:
  https://docs.chariot.io/display/CLD/Architecture+Examples
- ThingsPro resources:
  https://www.moxa.com/Event/industrial-computers/thingspro-data-acquisition-device-management-platform/resources.htm