Machfu Industrial IoT Gateway with SignalFire Reactor User Guide

Table of Contents

Table of Figures

1 Introduction

This document provides instructions for configuring the Machfu Industrial IoT (IIoT) Gateway to communicate with a SignalFire Gateway Stick, support polling of remotes and publish their data to any MQTT broker with SparkplugB or Plaintext (JSON) encoding.

1.1 System Overview

The Machfu IIoT Gateway locally polls SignalFire Gateway Stick and supported Remotes using Modbus protocol and publishes the polled data to topics on an MQTT broker using the SparkPlugB/JSON format.

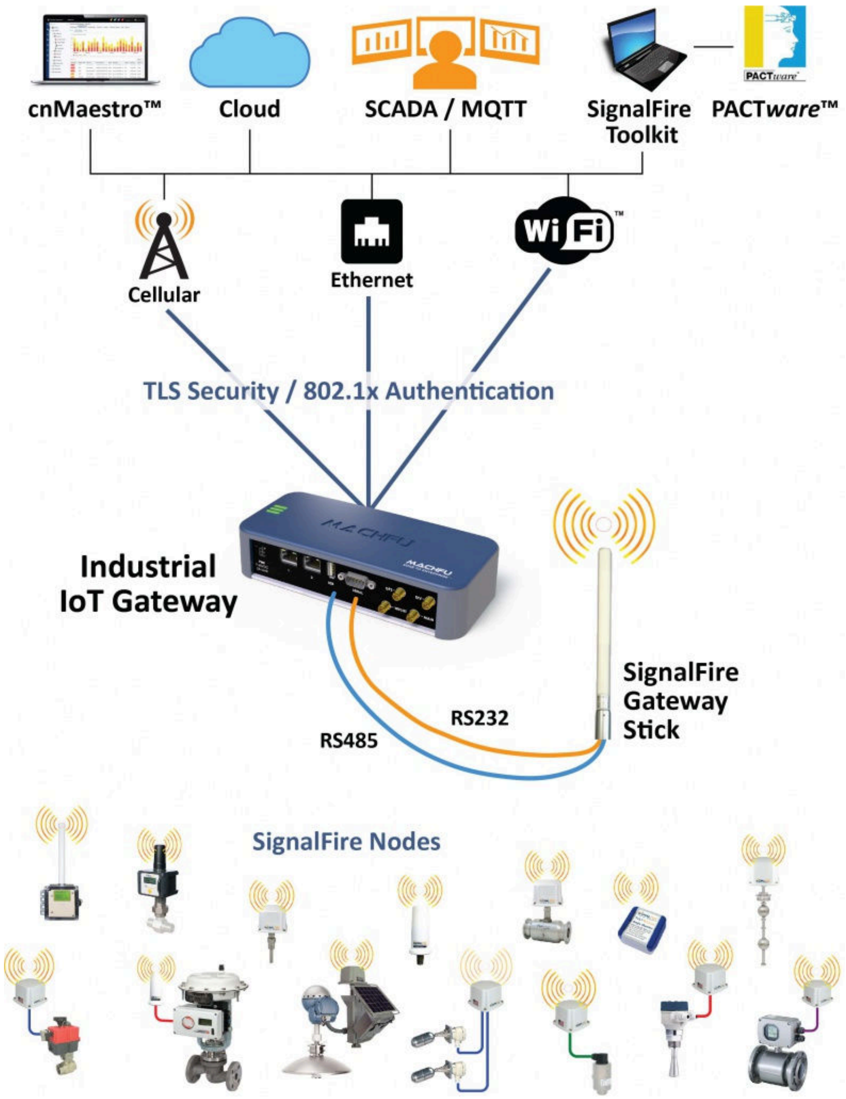


Figure 1: Machfu IIoT Gateway Reactor Connectivity with SignalFire

2 Mechanical installation of the Machfu IIoT Gateway

2.1 Mounting Plate Assembly

The Machfu Gateway makes optional mounting options available – mounting plate (Machfu Part No: AM002) to mount the Gateway directly to a wall, and DIN rail mount (Machfu Part No: AM001) to mount the gateway onto a DIN rail.

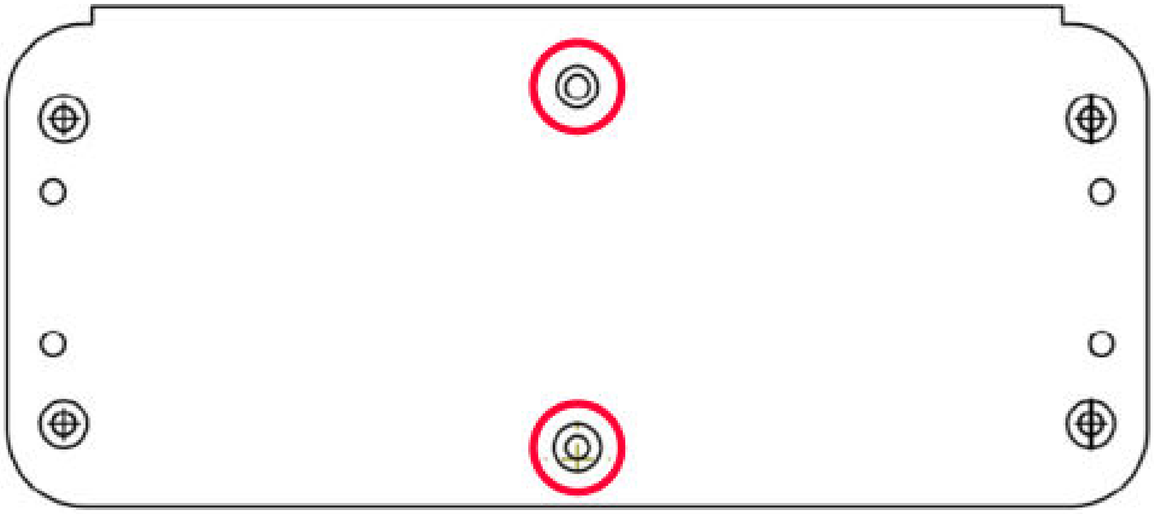


Figure 2: Machfu IIoT Gateway's base plate

1. Remove the two center screws from the base plate of the Machfu Gateway (marked with a red circle in Figure 2)
2. Align the mounting plate's counter screw holes with the center screw holes on the base plate of the Gateway and screw them together as indicated in Figure 3
3. Mount the Gateway into your unit using the exposed screw holes available on the mounting plate.

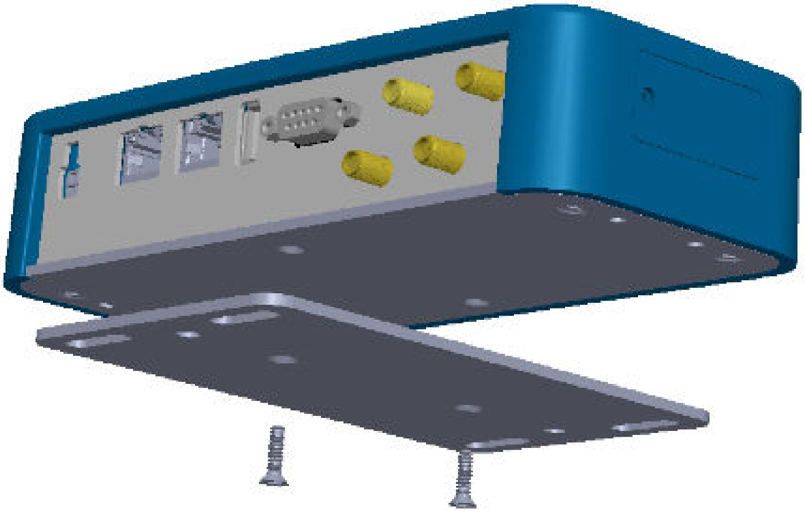


Figure 3: Machfu IIoT Gateway's mounting plate screwed into the bottom base plate

2.2 Power Connection

⚠ Only use the power cable supplied with the gateway! (Machfu Part No: AP001)

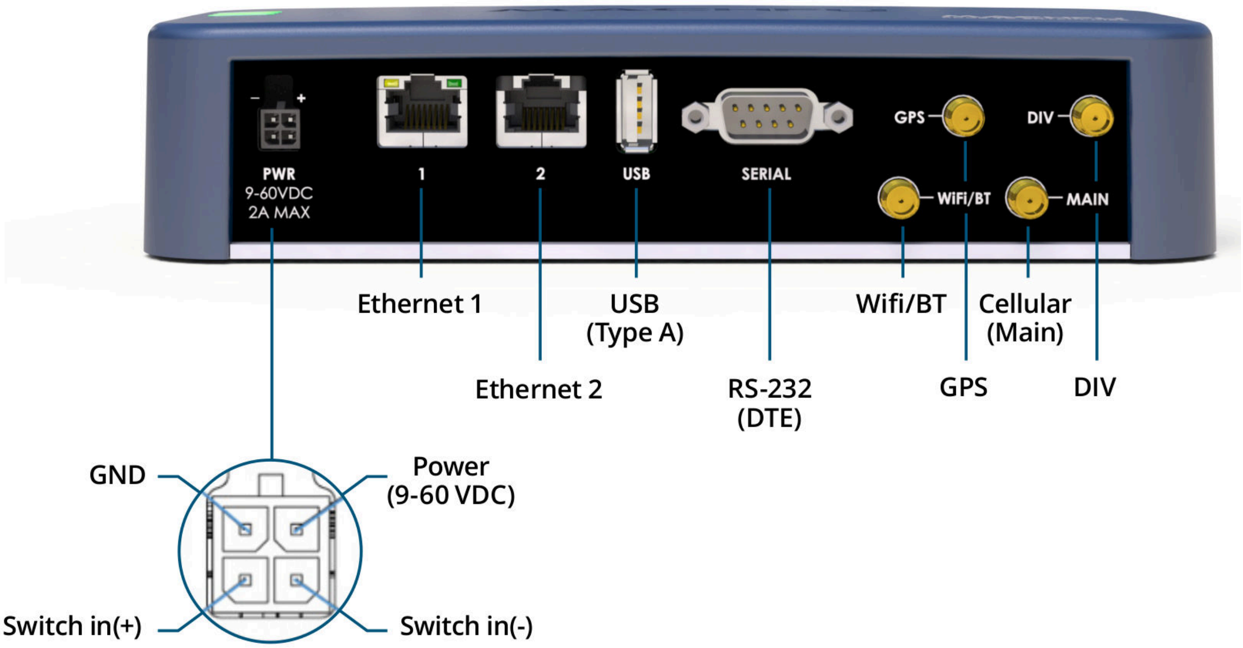


Figure 4: Machfu IIoT Gateway Interfaces

⚠ DO NOT USE ANY UNSUPPORTED POWER CABLE THAT APPEARS TO HAVE THE SAME POWER CONNECTOR AS THE MACHFU GATEWAY WITHOUT REWIRING AS PER Figure 4. Please follow the details below to connect the power cable correctly.

- **Molded Cable with Machfu Part No. AP001**
  - Red Wire – Positive Terminal
  - Green Wire – Negative Terminal
  - Black Wire – Clip off or secure with insulation tape
  - White Wire – Clip off or secure with insulation tape

2.3 SignalFire Gateway Stick

Figure 5: SignalFire Gateway Stick

2.4 Connecting the Machfu IIoT Gateway to SignalFire Gateway Stick

Connect the SignalFire Breakout to an AC Power supply

1. Connect one end of the SignalFire Breakout to the Gateway Stick and connect the other end that says A and B to the RS 485 to RS 232 converter into T/R+ and T/R- terminal ports respectively as shown in Figure 8 below.
2. Connect the converter to the Pluggable Dongle provided with the Machfu kit. Refer Figure 6 below for proper connection set up.

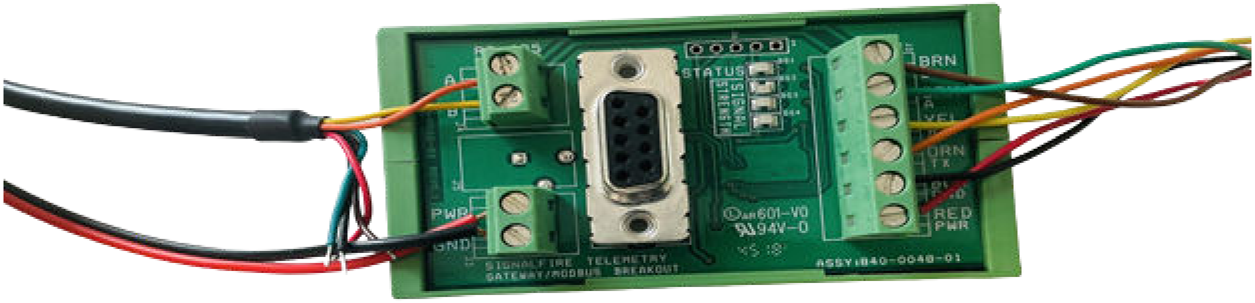


Figure 6: SignalFire Breakout Dongle interface

3 Configuring the Machfu IIoT Gateway

3.1 Overview

Configuring the Machfu Gateway has the following steps:

1. Log-in to the Machfu Gateway.
2. Browsing to SignalFire MachReactor Discovery UI.
3. Configure MQTT Client to communicate with the Broker and click the “Submit” button.

3.2 Logging into the Gateway's web UI

1. Turn on the Gateway and connect the **Ethernet 1** port on the Gateway to your computer using a LAN cable.
2. DHCP is turned on by default on this port, hence it will issue an IP address to your computer.
3. Go to your browser and type https://192.168.1.1:8443 in the address field.
4. The browser will load the MACHFU Web UI login page.
5. The default username is **admin** and the password is **ChangeIt**
6. The Gateway will now prompt you to change the default password. You may click the **DASHBOARD** tab on the left side menu to exit this screen. **We recommend you change the password to prevent unauthorized access to the Gateway UI.**

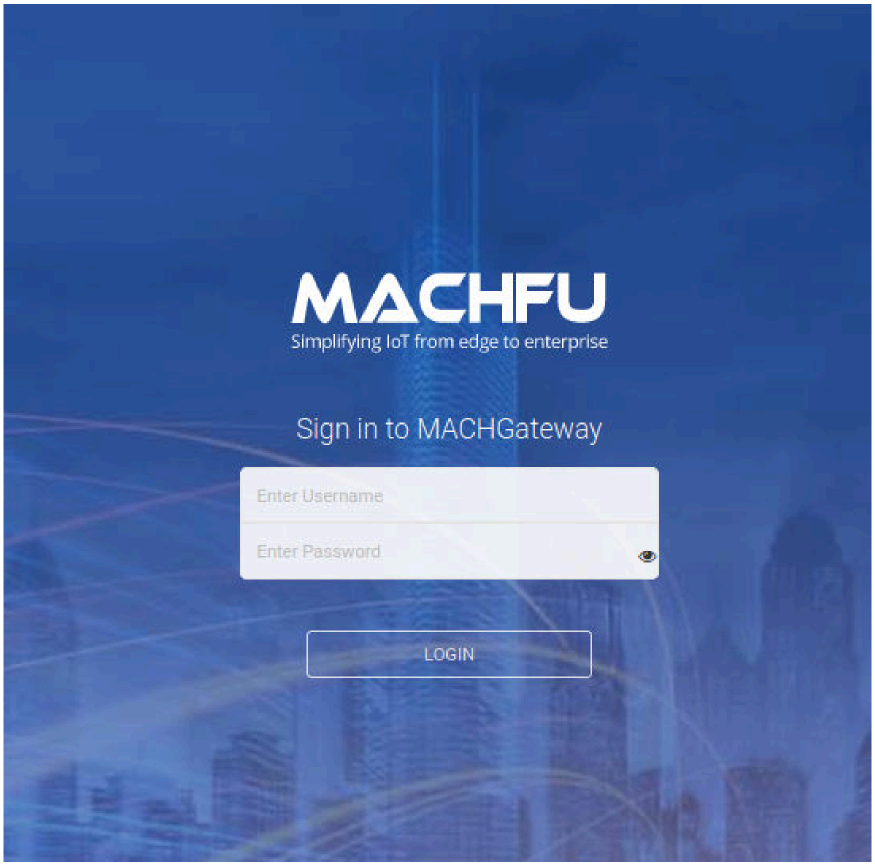


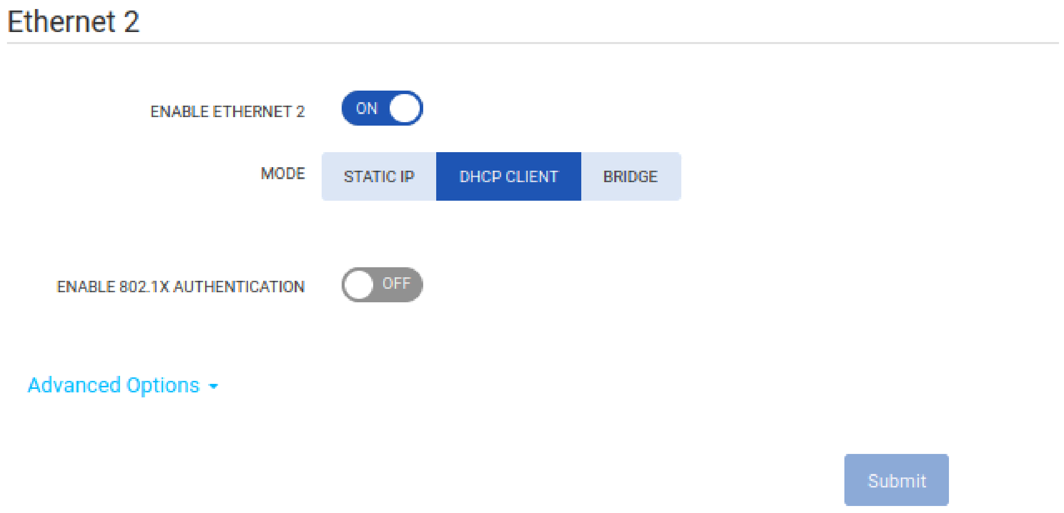
Figure 7: Machfu IIoT Gateway Web UI Login

3.3 Connectivity to the Cloud

The Machfu Gateway has several network interfaces that helps you connect to the outside internet and the cloud. Let us look at some of the interfaces:

3.3.1 Ethernet 2

1. On the Dashboard page, click on the Ethernet 2 link under the Wired section on the left navigation panel. You have the option of enabling or disabling an Ethernet interface even if the interface is physically connected.
2. To enable the Ethernet 2 interface, set Enable Ethernet switch to 'ON' and to disable, set it to 'OFF'.
3. Click the Submit button once enabling the Ethernet interface.
4. By default, Ethernet 2 interface is configured as DHCP client.



ENABLE ETHERNET 2: ON

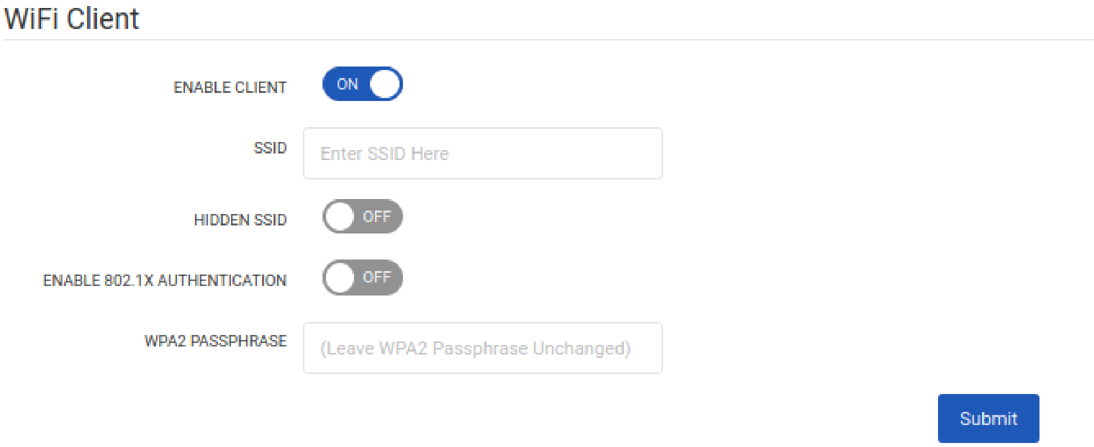
MODE: DHCP CLIENT

ENABLE 802.1X AUTHENTICATION: OFF

Figure 8: Machfu IIoT Gateway Ethernet 2 Settings

3.3.2 WiFi Client

1. On the Dashboard page, click on the WiFi Client link under the Wireless section on the left navigation panel.
2. Set the 'Enable' button to use the Wi-Fi in the Client mode.
3. SSID – Specify the wireless network name or SSID (Service Set Identifier) used to identify the WLAN.
4. Enter the WPA2 Passphrase.
5. Click the Submit button once enabling the WiFi Client interface.



WiFi Client

ENABLE CLIENT ON

SSID Enter SSID Here

HIDDEN SSID OFF

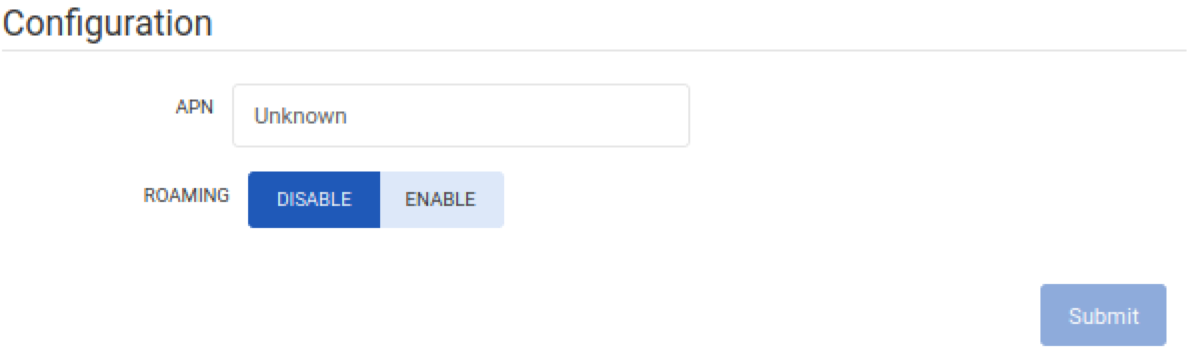
ENABLE 802.1X AUTHENTICATION OFF

WPA2 PASSPHRASE (Leave WPA2 Passphrase Unchanged)

Figure 9: Machfu IIoT Gateway WiFi Client Settings

3.3.3 Cellular

1. On the Dashboard page, click on the Cellular link under the Wireless section on the left navigation panel.
2. Set the 'APN' of the cellular SIM.
3. Set cellular 'Operator' (for select models).
4. Click the Submit button once enabling the Cellular interface.



Configuration

APN Unknown

ROAMING DISABLE ENABLE

Figure 10: Machfu IIoT Gateway Cellular Settings

3.4 Browsing to SignalFire Reactor Discovery UI

At the top on the Dashboard page in the gateway UI, click the SignalFire Reactor icon to browse to the configuration UI as shown in the Figure below.

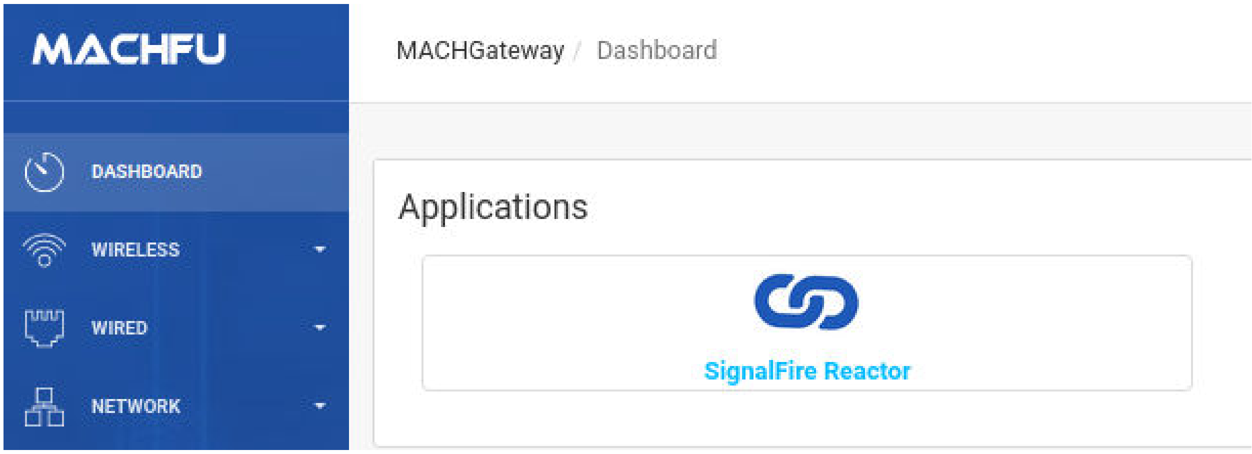


Figure 11: SignalFire Reactor Icon on the Dashboard Page

The SignalFire Reactor Discovery UI can also be accessed by typing the following in the address bar of the browser:

https://192.168.1.1:8443/MachReactor/#discovery

3.5 MQTT Broker settings

3.5.1 Mandatory fields

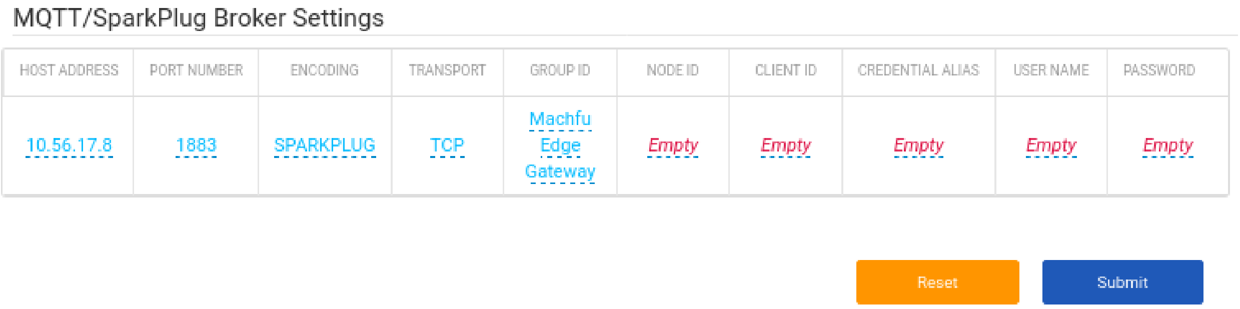
The mandatory fields for setting up the MQTT broker are:

1. Host Address: Any valid URL or IPv4 address are accepted inputs.
2. Port Number: Only numerical values are allow. The default value for secured TLS connections is 8883, while the default value for unsecured TCP connections is 1883.
3. Encoding: The provided options are:
   a. "SPARKPLUG" for SparkPlugB-encoded messages and this is the default option.
   b. "JSON" for plaintext JSON-formatted messages
4. Transport: The provided options are:
   a. "TLS" for secured TLS connections using an imported credential alias
   b. "TCP" for unsecured TCP connections using optional username and password
5. Group ID: Any text is accepted to represent the parent node of the MQTT messages.

3.5.2 Configuration

To configure the MQTT broker, please follow the steps below.

1. Enter the Host Address / URL to identify the address to which the Gateway will publish and subscribe MQTT data from. This field by default is "Empty".
2. Select the desired Transport Protocol:
   - Selecting TCP sets an insecure TCP connection between the Machfu Gateway and the MQTT broker. Username and Password may be required to connect to the broker.

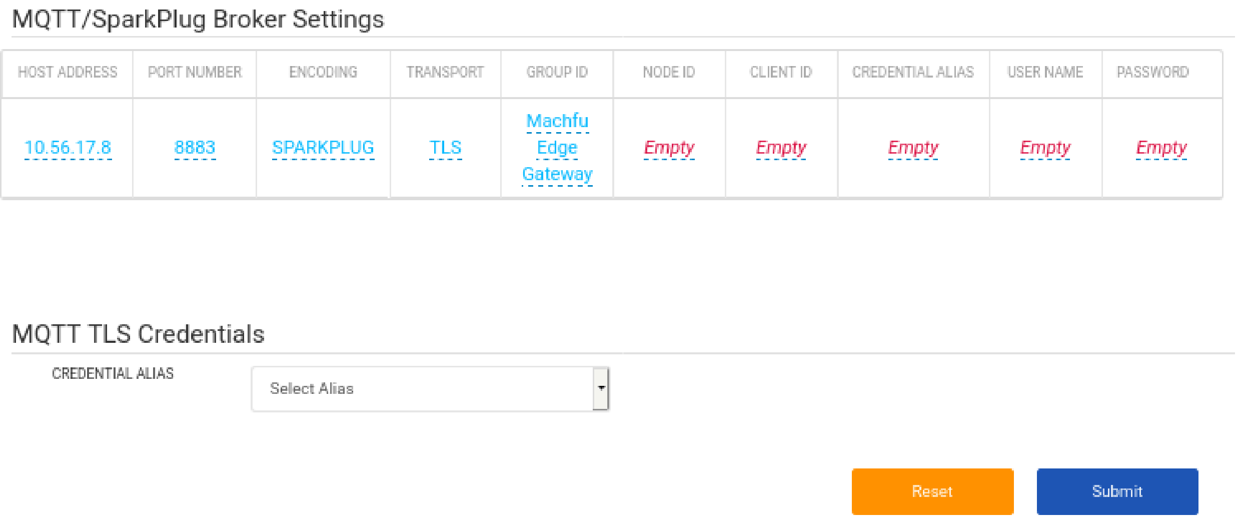


MQTT/SparkPlug Broker Settings

HOST ADDRESS	PORT NUMBER	ENCODING	TRANSPORT	GROUP ID	NODE ID	CLIENT ID	CREDENTIAL ALIAS	USER NAME	PASSWORD
10.56.17.8	1883	SPARKPLUG	TCP	Machfu Edge Gateway	Empty	Empty	Empty	Empty	Empty

Figure 12: MQTT Broker Settings TCP Protocol

   - Selecting TLS ensures a secure TLS connection between the Machfu Gateway and the MQTT broker. Once selected, a configuration box will appear enabling the selection of a set of TLS credentials. Note that the TLS credentials need to be loaded on the gateway as described in 4 Downloading TLS Credentials.



MQTT/SparkPlug Broker Settings

HOST ADDRESS	PORT NUMBER	ENCODING	TRANSPORT	GROUP ID	NODE ID	CLIENT ID	CREDENTIAL ALIAS	USER NAME	PASSWORD
10.56.17.8	8883	SPARKPLUG	TLS	Machfu Edge Gateway	Empty	Empty	Empty	Empty	Empty

MQTT TLS Credentials

CREDENTIAL ALIAS: Select Alias

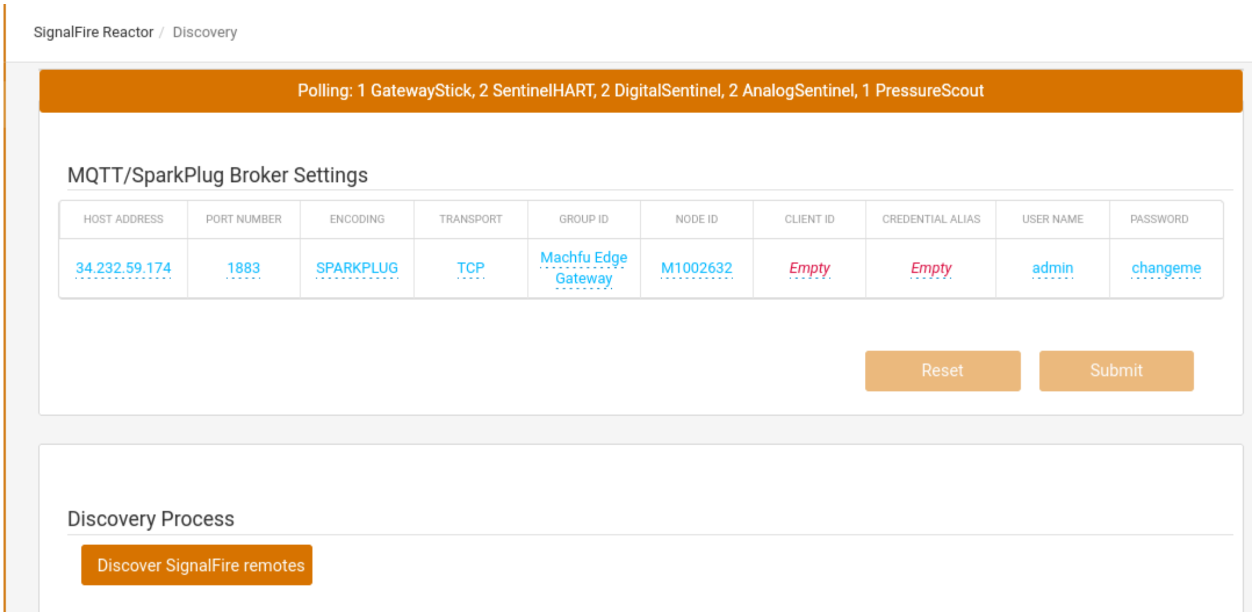
Figure 13: MQTT Broker TLS Settings

3.5.1 Encoding/Presentation

MQTT payloads are published using the SPARKPLUG format by default. Optionally, payloads can be sent in plain-text JSON format.

3.6 SignalFire Discovery

The auto-discovery process for connected SignalFire nodes can be initiated by updating the MQTT settings or by clicking "SignalFire Network Discovery" button.



SignalFire Reactor / Discovery

Polling: 1 GatewayStick, 2 SentinelHART, 2 DigitalSentinel, 2 AnalogSentinel, 1 PressureScout

MQTT/SparkPlug Broker Settings

HOST ADDRESS	PORT NUMBER	ENCODING	TRANSPORT	GROUP ID	NODE ID	CLIENT ID	CREDENTIAL ALIAS	USER NAME	PASSWORD
34.232.59.174	1883	SPARKPLUG	TCP	Machfu Edge Gateway	M1002632	Empty	Empty	admin	changeme

Discovery Process

Figure 14: Starting Discovery process

When the discovery operation is started, the user is notified of the status of operations at the top of the page. The display messages are in the following format:

- Starting SignalFire Network Discovery
- Checking connected SignalFire remotes
- Discovered: 1 Gateway Stick, X Nodes
- Activated Devices
- Polling: 1 Gateway Stick, X Nodes

3.6.1 Supported Device Types

The SignalFire devices supported by default are:

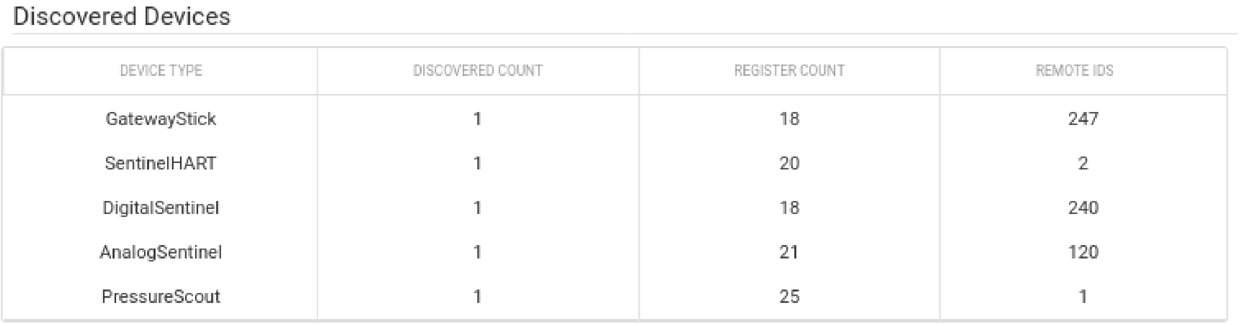
1. Analog Sentinel
2. Digital Sentinel
3. HART Sentinel
4. Pressure Scout

Additional devices can be supported by editing and uploading a new Configuration file. Please contact Machfu Support on how to add more devices to the discovery process.

3.6.2 Discovered Devices

This table shows the basic information about the discovered SignalFire nodes along with the Gateway Stick. These information are:

- Device Type: Type of SignalFire device
- Discovered Count: Number of SignalFire device discovered
- Register Count: Number of Modbus registers configured for the device
- Remote IDs: List of Modbus IDs for the SignalFire device



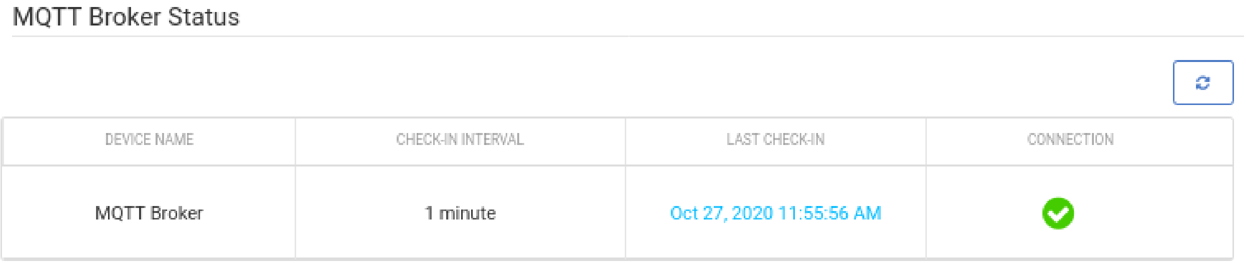
Discovered Devices

DEVICE TYPE	DISCOVERED COUNT	REGISTER COUNT	REMOTE IDS
GatewayStick	1	18	247
SentinelHART	1	20	2
DigitalSentinel	1	18	240
AnalogSentinel	1	21	120
PressureScout	1	25	1

Figure 15: Discovered Devices

3.6.2 MQTT Broker Status

This table updates the user with live connection status of the MQTT broker.



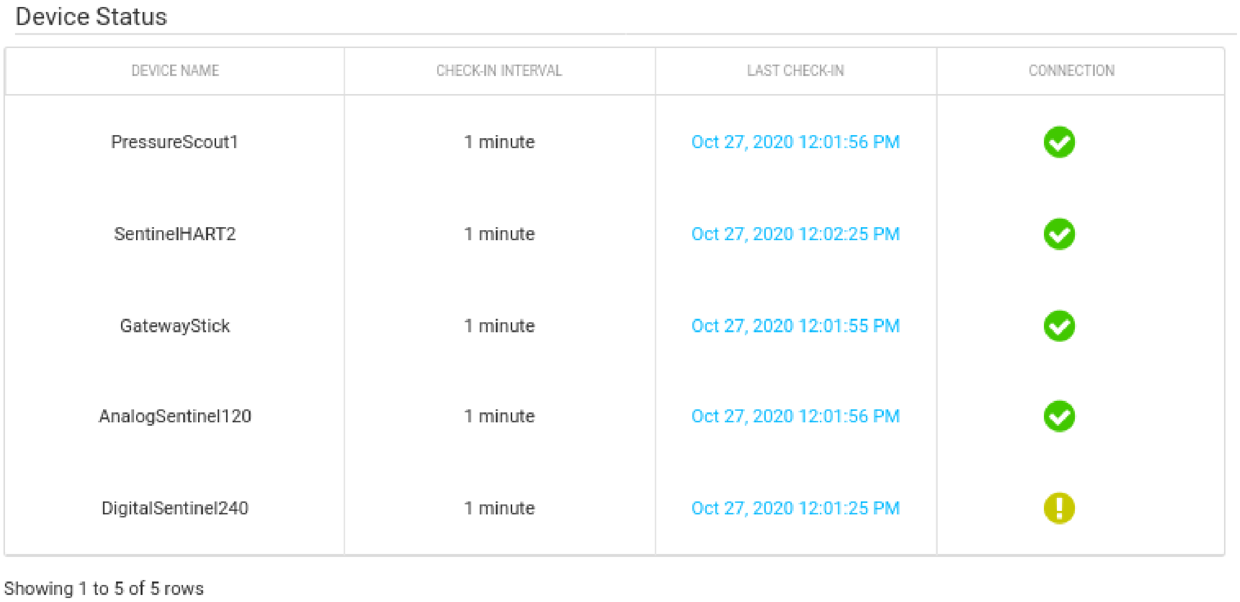
MQTT Broker Status

DEVICE NAME	CHECK-IN INTERVAL	LAST CHECK-IN	CONNECTION
MQTT Broker	1 minute	Oct 27, 2020 11:55:56 AM	✓

Figure 16: MQTT Broker Connection Status

3.6.2 Device Status

This table informs the user of live connection status of the SignalFire Gateway stick and discovered nodes.



Device Status

DEVICE NAME	CHECK-IN INTERVAL	LAST CHECK-IN	CONNECTION
PressureScout1	1 minute	Oct 27, 2020 12:01:56 PM	✓
SentinelHART2	1 minute	Oct 27, 2020 12:02:25 PM	✓
GatewayStick	1 minute	Oct 27, 2020 12:01:55 PM	✓
AnalogSentinel120	1 minute	Oct 27, 2020 12:01:56 PM	✓
DigitalSentinel240	1 minute	Oct 27, 2020 12:01:25 PM	!

Showing 1 to 5 of 5 rows

Figure 17: Connection Status of SignalFire devices

3.6.3 Status Information

The colored icon in the Status column of MQTT Broker & Device Status sections is used to represent the check-in (polling for Modbus, publishing for MQTT) status of each device and broker. The represented status are:

- OK sign ✅ displayed as a green icon with a check mark, means that the device is connected and the check-ins are behaving as expected.
- Caution sign ❗ displayed as a yellow icon with exclamation mark symbol, indicating that the device may be connected but there are more failed check-ins than successful in the most recent batch.
- Error sign ❌ displayed as a red icon with the cancel symbol indicating all the most recent check-in operations have failed and the device may not be connected.

4 Downloading TLS Credentials

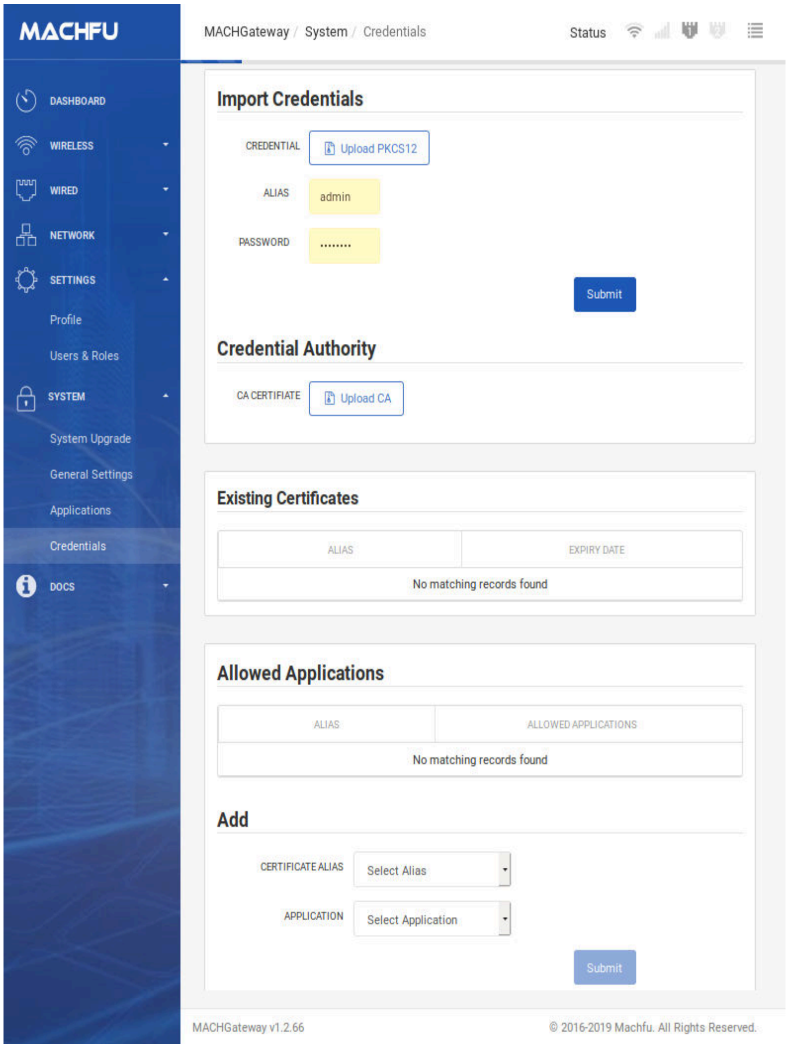
To use SSL/TLS connections to publish data to a MQTT broker using mutual authentication, an appropriate credential set must be uploaded and configured on the Machfu Gateway.

Machfu Gateway only supports enrolling the public, private key in PKCS12 format. If your key pair is in some other format, you may have to use "OpenSSL" to convert it into PKCS12 format.

Example: In case the keys are in PEM format,

```
openssl pkcs12 -export -out <PKCS12 File Name>.pfx -inkey <Private Key File Name> -in <Public Key File Name> -name <Alias>
```

The newly created <PKCS12 File Name>.pfx is in PKCS12 format and contains the key pair.



Import Credentials

CREDENTIAL: Upload PKCS12

ALIAS: admin

PASSWORD: ....

Submit

Credential Authority

CA CERTIFIATE: Upload CA

Existing Certificates

ALIAS	EXPIRY DATE
No matching records found	

Allowed Applications

ALIAS	ALLOWED APPLICATIONS
No matching records found	

Add

CERTIFICATE ALIAS: Select Alias

APPLICATION: Select Application

Submit

MACHGateway v1.2.66

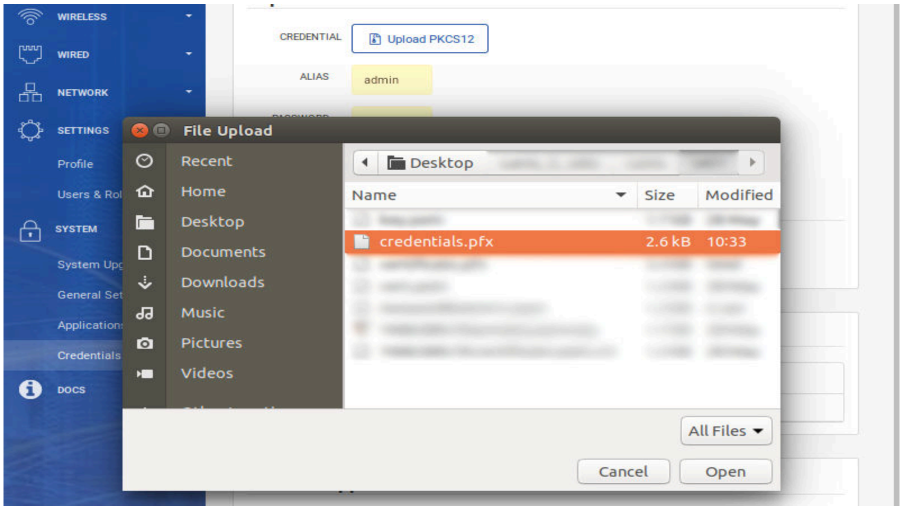
© 2016-2019 Machfu. All Rights Reserved.

Figure 18: Downloading TLS Credentials

4.1 Uploading TLS Credentials

Please follow the steps below to upload a credentials set:

1. Go to the Machfu Gateway web UI.
2. Using the left side navigation menu, select System >> Credentials
3. The “Credentials” page will appear on your browser.
4. In the "Import Credentials" section, click the "Upload PKCS12" button and select the PKCS12 file containing the public and private key you would like to use for the connection.



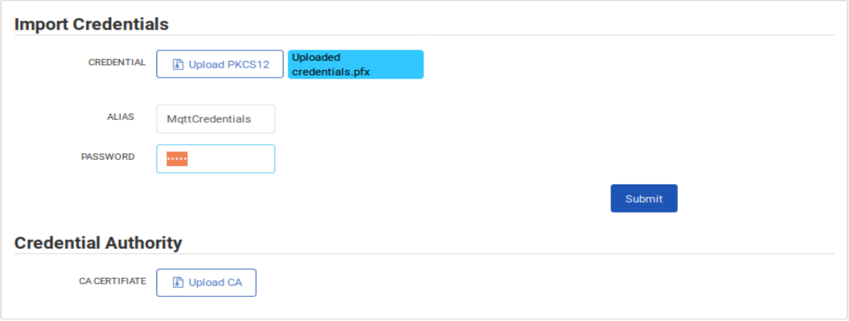
CREDENTIAL Upload PKCS12

ALIAS admin

credentials.pfx 2.6 kB 10:33

Figure 19: PKCS12 File Upload

5. Enter the "Alias" that was used while creating the PKCS12 file.
6. Enter the "Password" that was used while creating the PKCS12 file.



Import Credentials

CREDENTIAL Upload PKCS12 Uploaded credentials.pfx

ALIAS MqttCredentials

PASSWORD •••••

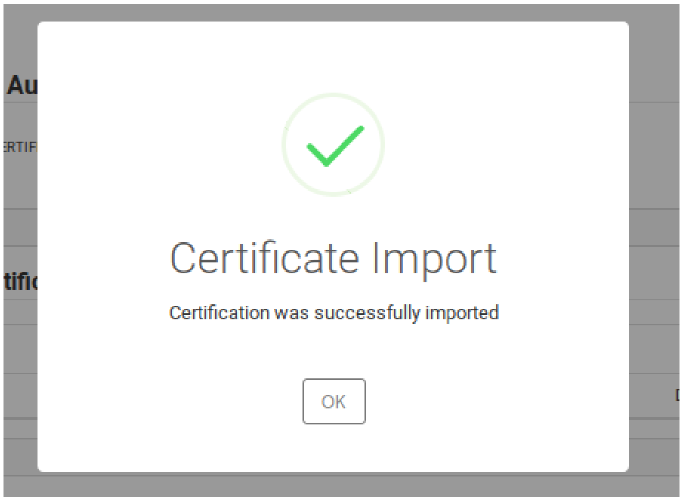
Submit

Credential Authority

CA CERTIFIATE Upload CA

Figure 20: Enter Password under 'Import Credentials'

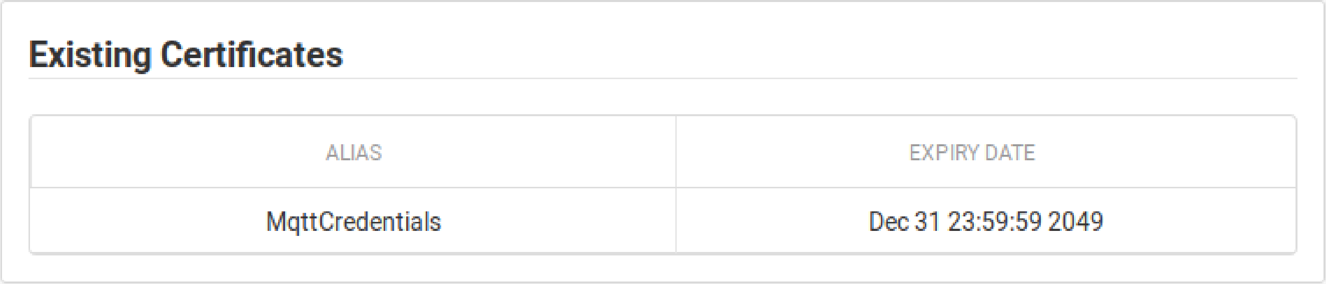
7. Press Submit.
8. If successfully installed, a confirmation message will pop up.
9. The installed credentials are now displayed in the "Existing Certificates" section.



Certificate Import

Certification was successfully imported

Figure 21: Import successful confirmation pop-up window



Existing Certificates

ALIAS	EXPIRY DATE
MqttCredentials	Dec 31 23:59:59 2049

Figure 22: Existing Certificates display

4.2 Upload CA

The Machfu Gateway already comes installed with well-known CAs. In case you are using a self-signed key pair you may need to install the CA. This can be done using the following steps:

1. Click the “TRUSTED” button in the “TYPE” section.
2. Click the "Upload CA" button in the "Credential Authority" section and choose the CA certificate file in PEM format.
3. If successfully installed, a confirmation message will pop up as shown below.

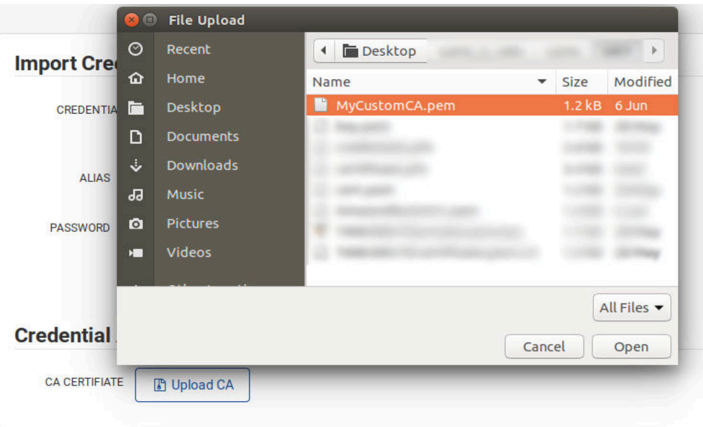
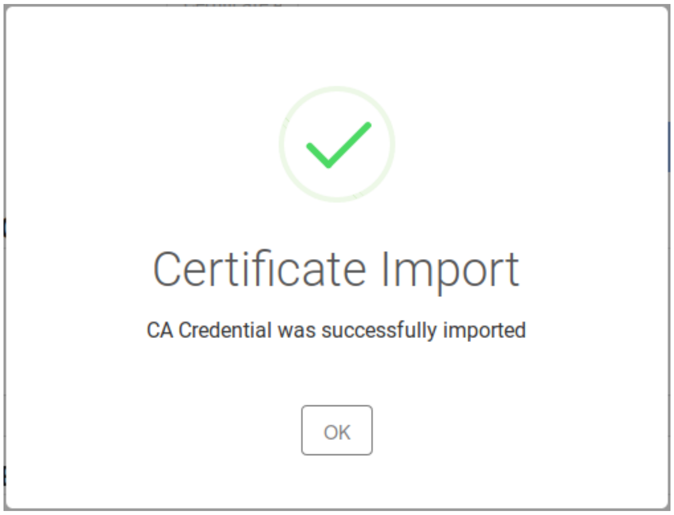


Figure 23: Custom CA File Upload



Certificate Import

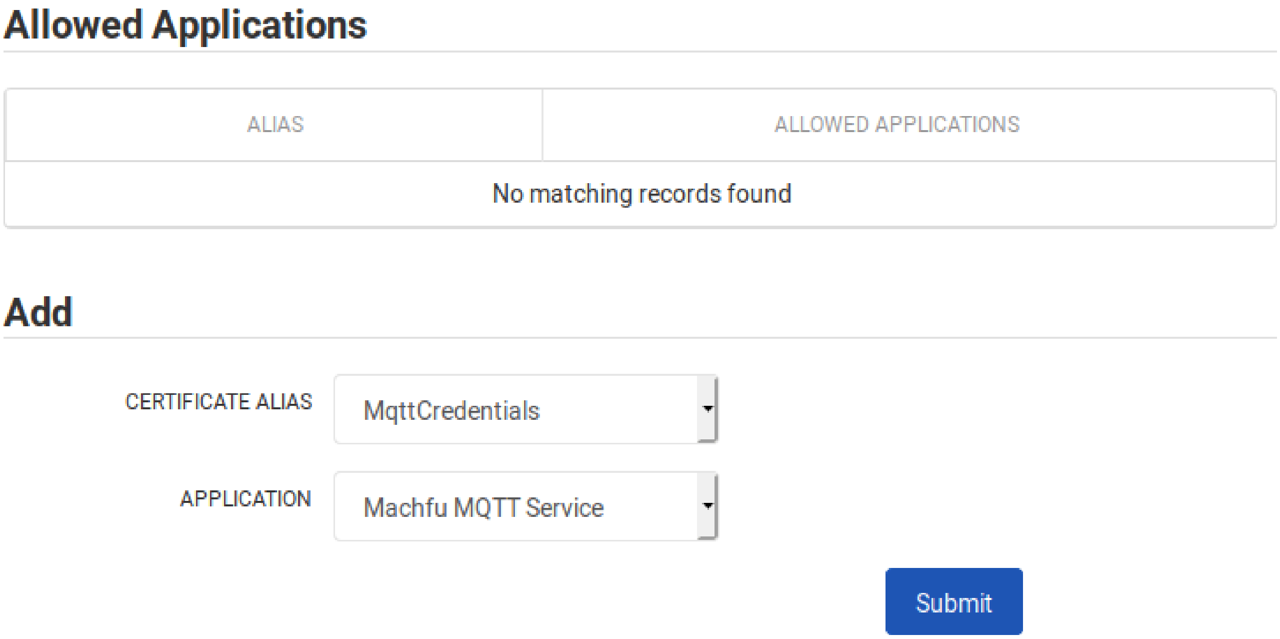
CA Credential was successfully imported

Figure 24: Import successful confirmation pop-up window

4.3 Allowing MQTT Service Access to Credentials

Now that you have installed the credentials, the Machfu MQTT Service must be allowed to access them. This can be done, by following the steps below:

1. In the "Allowed Applications/Add" section, Select the "Alias" installed in the previous section.
2. In the "Application" field select "Machfu MQTT Service".



Allowed Applications

ALIAS	ALLOWED APPLICATIONS
No matching records found	

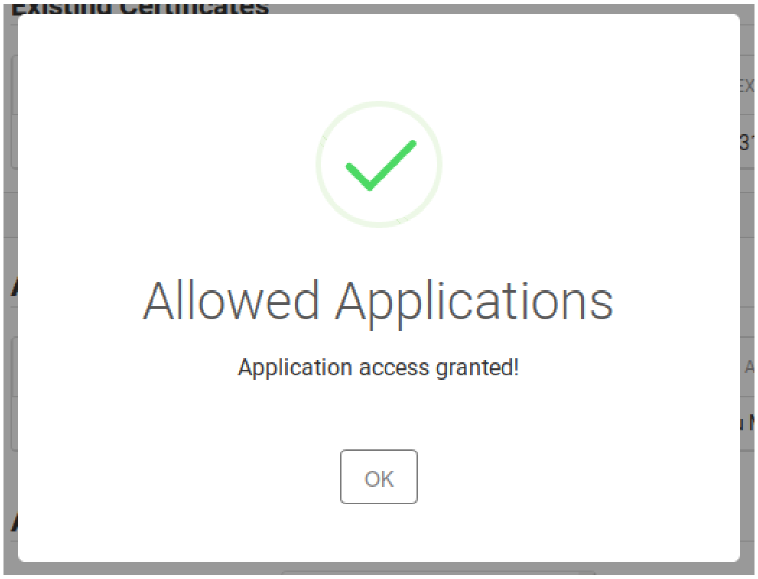
Add

CERTIFICATE ALIAS: MqttCredentials

APPLICATION: Machfu MQTT Service

Figure 25: Allow MQTT Service Access to Credentials

3. Press "Submit"
4. If successfully installed, a confirmation message will pop up.

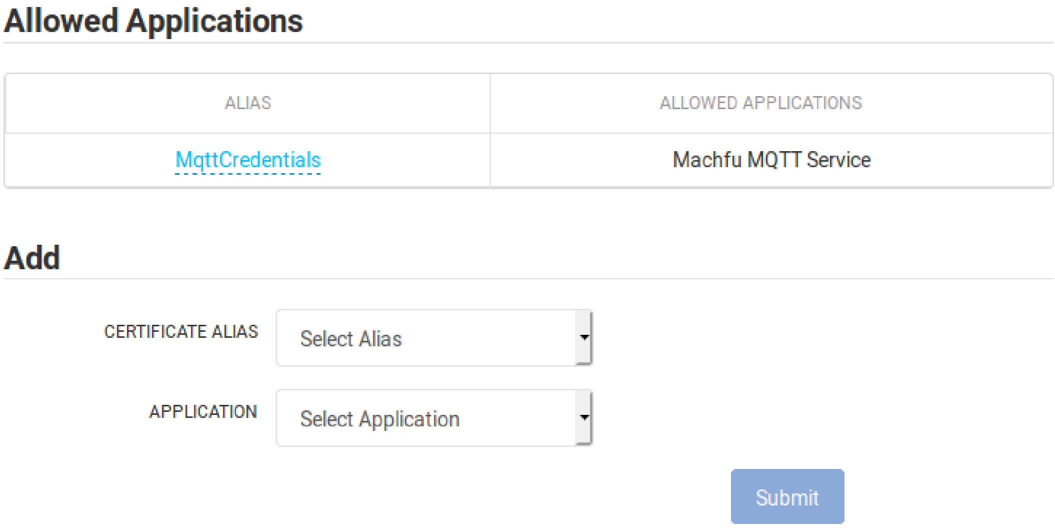


Allowed Applications

Application access granted!

Figure 26: Access granted confirmation

5. Once completed, the alias and application name shall appear in the "Allowed Applications" table.



Allowed Applications

ALIAS	ALLOWED APPLICATIONS
MqttCredentials	Machfu MQTT Service

Add

CERTIFICATE ALIAS: Select Alias

APPLICATION: Select Application

Figure 27: Allowed Application List

5 Appendix A: Setting up Ignition

If you are using the Machfu Gateway with Ignition backend, then this section will help you to set up an Ignition instance.
Install the Ignition platform and the MQTT broker as follows:
This section of the tutorial will provide step by step instructions for installing the Ignition Gateway in an Industrial Application Platform with the following modules:

- Ignition Platform
- MQTT Distributor – An MQTT Server that runs as an Ignition module.
- MQTT Engine – Engine is an MQTT Client that implements the SparkPlugB specification and automatically creates Ignition tag structures for the Edge node, device metadata and process variables.

Upon completion of this tutorial, you will have all the required Ignition components to configure and deploy the Machfu Gateway.

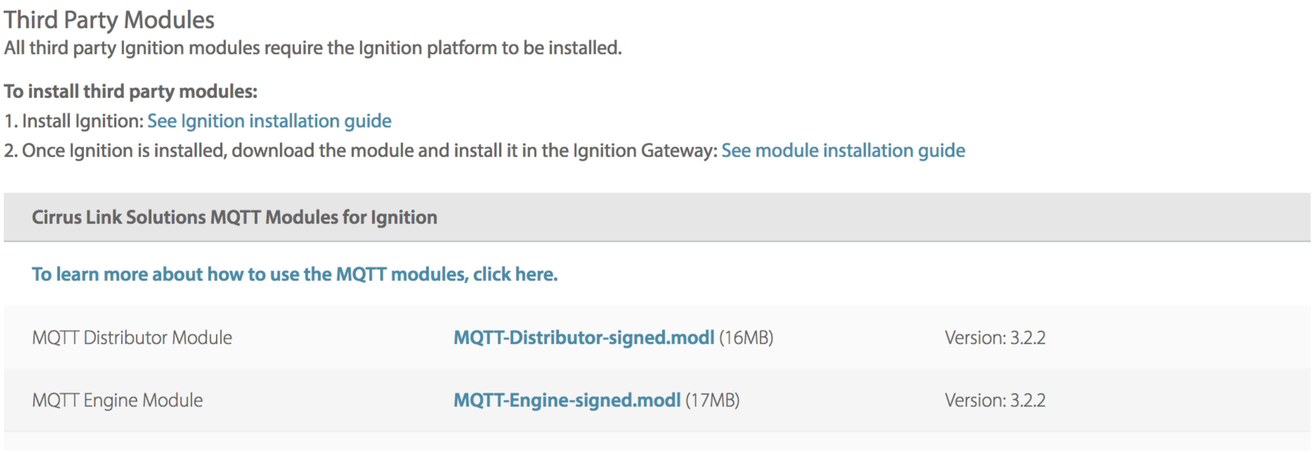
5.1 Step 1: Download and install Ignition

Ignition is an Industrial Application Platform that can be used to create SCADA and HMI solutions. A fully functional Ignition system can be downloaded and run in trial mode. Using Ignition as a tool, the Sparkplug MQTT Modules can be installed to observe everything working. Go to the Inductive Automation download page and download the desired version of the Ignition installer for Windows, Linux or MacOS; inductiveautomation.com/downloads/archive.
Once the Ignition installer has been downloaded, follow the instructions provided in the Inductive Automation page to install and startup Ignition.
(Note: For this test infrastructure, MQTT Distributor will be installed as an Ignition module. Remember to either turn off firewalls or at a minimum allow inbound connections to TCP/IP port #1883 and port #8883, as remote MQTT Clients should be able to establish a TCP/IP socket connection to these ports).

5.2 Step 2: Download and install the Cirrus Link MQTT Modules

Go to the Inductive Automation download inductiveautomation.com/downloads/archive.

Scroll down to the 3rd Party modules section. Find the Cirrus Link modules section and download the MQTT Distributor, MQTT Engine.
The download links should look like the below screenshot:



Third Party Modules

All third party Ignition modules require the Ignition platform to be installed.

To install third party modules:

1. Install Ignition: See Ignition installation guide
2. Once Ignition is installed, download the module and install it in the Ignition Gateway: See module installation guide

Cirrus Link Solutions MQTT Modules for Ignition

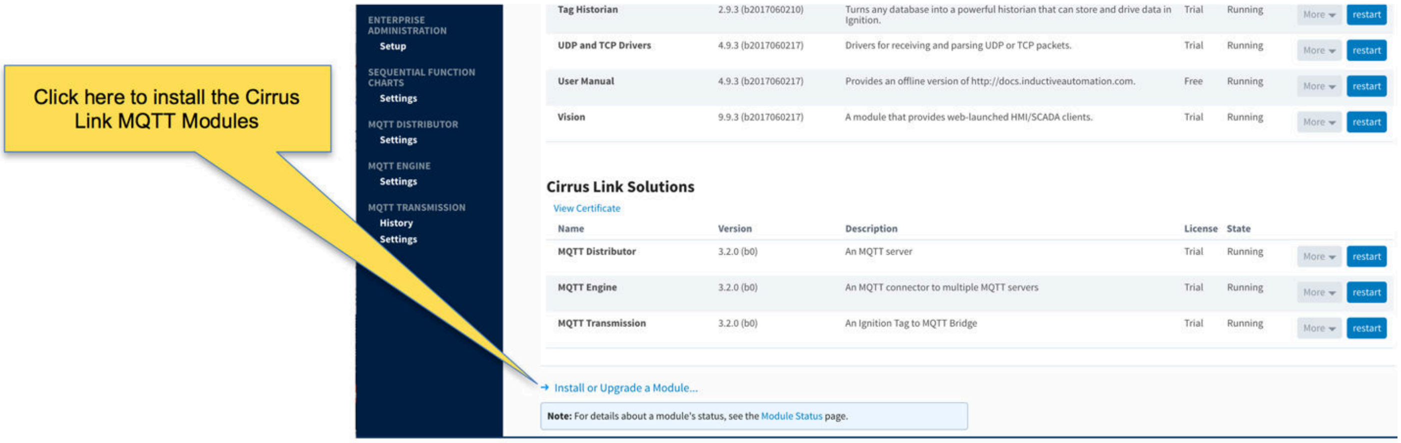
To learn more about how to use the MQTT modules, click here.

MQTT Distributor Module	**MQTT-Distributor-signed.modl** (16MB)	Version: 3.2.2
MQTT Engine Module	**MQTT-Engine-signed.modl** (17MB)	Version: 3.2.2

Figure 28: Links to download MQTT modules

5.3 Step 3: Install the MQTT Modules

Once you have Ignition installed and running, and the MQTT Distributor and MQTT Engine downloaded, browse to the Ignition Gateway console (e.g. http://localhost:8088). Login using the default credentials of admin/password. Click on Configuration tab and then click on the Modules tab on the left side of the page. Scroll down to the bottom of the Modules section and click on the Download/Upgrade modules button. When prompted, select the MQTT Distributor module from the file browser and install it. Do the same for the MQTT Engine. When complete, the Ignition Gateway Web UI module section should look like the below screenshot:



Click here to install the Cirrus Link MQTT Modules

Tag Historian	2.9.3 (b2017060210)	Turns any database into a powerful historian that can store and drive data in Ignition.	Trial	Running
UDP and TCP Drivers	4.9.3 (b2017060217)	Drivers for receiving and parsing UDP or TCP packets.	Trial	Running
User Manual	4.9.3 (b2017060217)	Provides an offline version of http://docs.inductiveautomation.com.	Free	Running
Vision	9.9.3 (b2017060217)	A module that provides web-launched HMI/SCADA clients.	Trial	Running

Cirrus Link Solutions

View Certificate

Name	Version	Description	License	State
MQTT Distributor	3.2.0 (b0)	An MQTT server	Trial	Running
MQTT Engine	3.2.0 (b0)	An MQTT connector to multiple MQTT servers	Trial	Running
MQTT Transmission	3.2.0 (b0)	An Ignition Tag to MQTT Bridge	Trial	Running

→ Install or Upgrade a Module...

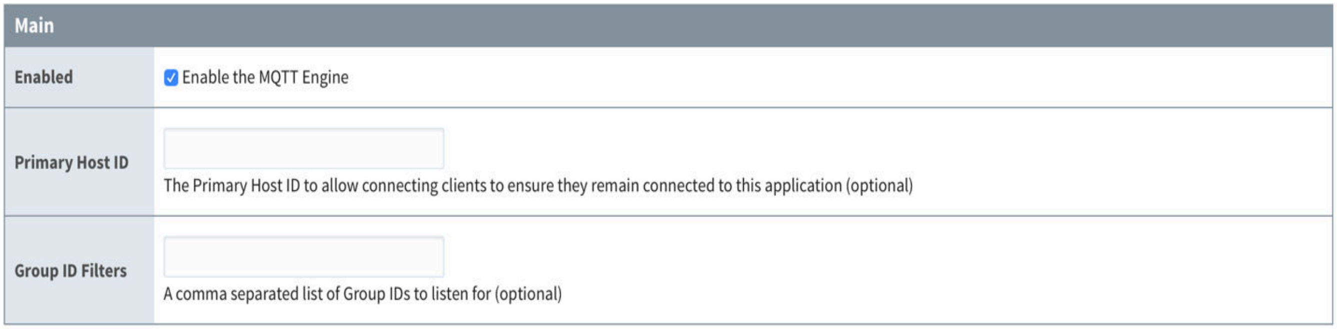
Note: For details about a module's status, see the Module Status page.

Figure 29: MQTT Modules Installation/Upgrade link

5.4 Step 4: Use Ignition Designer to Examine the Initial Tag Structure

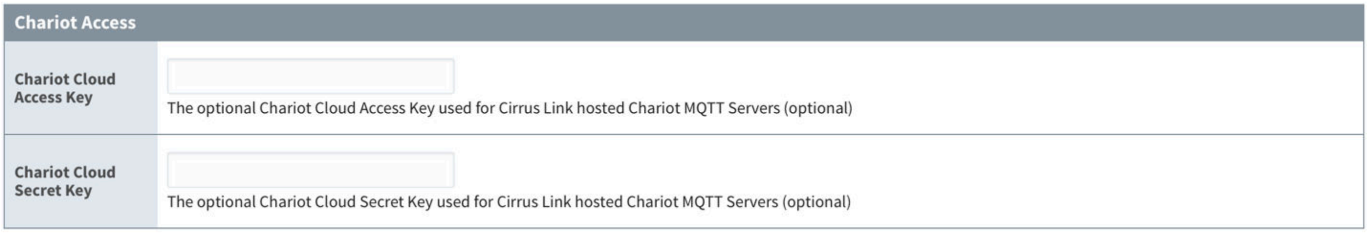
By default, MQTT Distributor, MQTT Engine, and MQTT Transmission are all configured out of the box to connect locally with each other in the exact same architecture of this tutorial. You can examine the MQTT setting of the Ignition client by opening the MQTT Engine configuration tab in the Ignition Gateway console.

5.4.1 MQTT Engine Settings



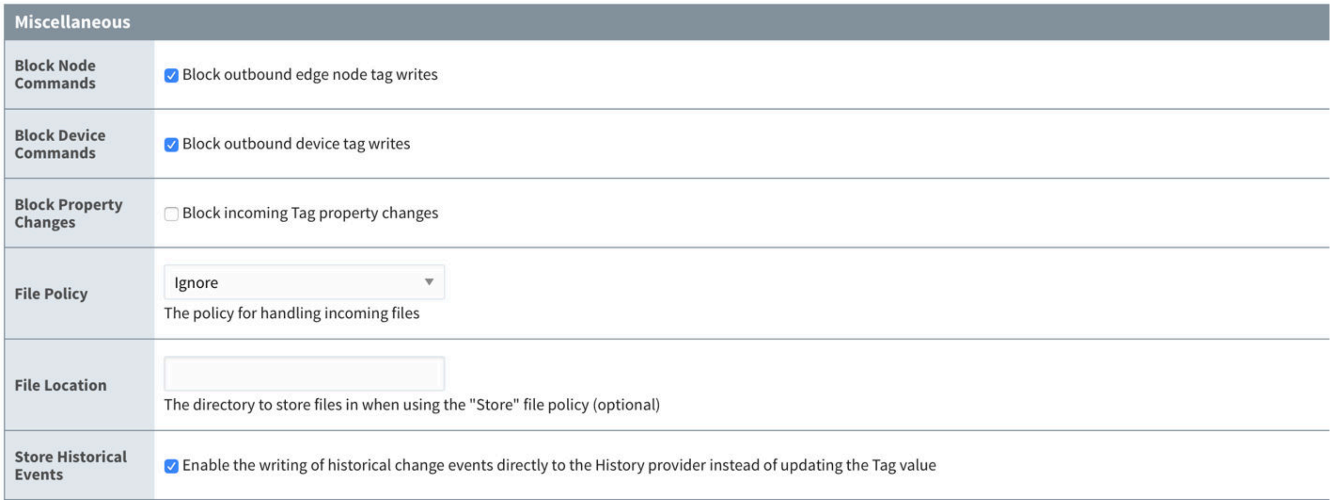
Main	
Enabled	☑ Enable the MQTT Engine
Primary Host ID	The Primary Host ID to allow connecting clients to ensure they remain connected to this application (optional)
Group ID Filters	A comma separated list of Group IDs to listen for (optional)

Figure 30: MQTT Engine Settings section 1



Chariot Access	
Chariot Cloud Access Key	The optional Chariot Cloud Access Key used for Cirrus Link hosted Chariot MQTT Servers (optional)
Chariot Cloud Secret Key	The optional Chariot Cloud Secret Key used for Cirrus Link hosted Chariot MQTT Servers (optional)

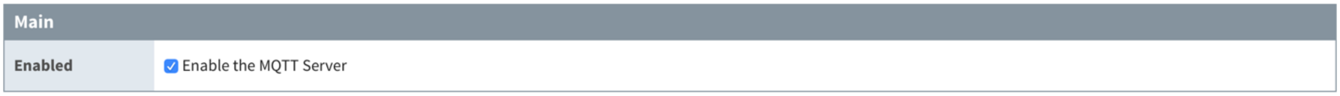
Figure 31: MQTT Engine Settings section 2



Miscellaneous	
Block Node Commands	☑ Block outbound edge node tag writes
Block Device Commands	☑ Block outbound device tag writes
Block Property Changes	☐ Block incoming Tag property changes
File Policy	Ignore ▾ The policy for handling incoming files
File Location	The directory to store files in when using the "Store" file policy (optional)
Store Historical Events	☑ Enable the writing of historical change events directly to the History provider instead of updating the Tag value

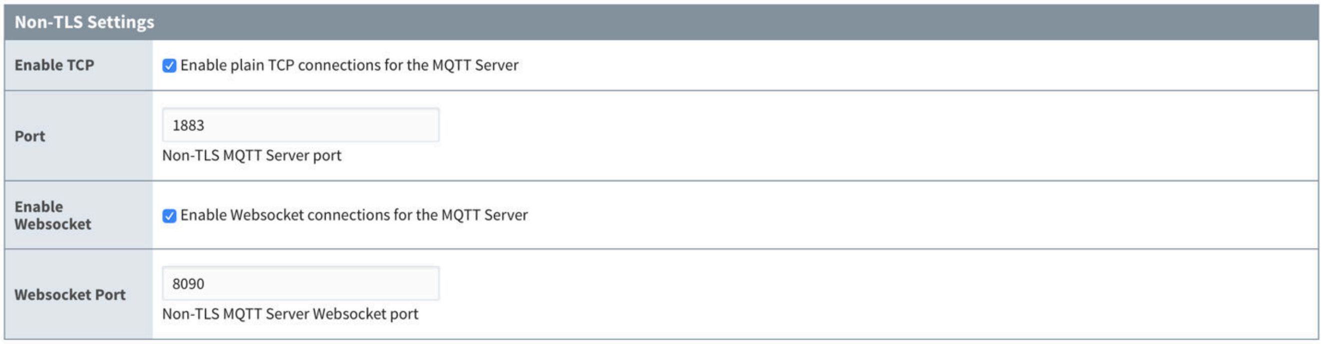
Figure 32: MQTT Engine Settings section 3

5.4.2 MQTT Distributor Settings



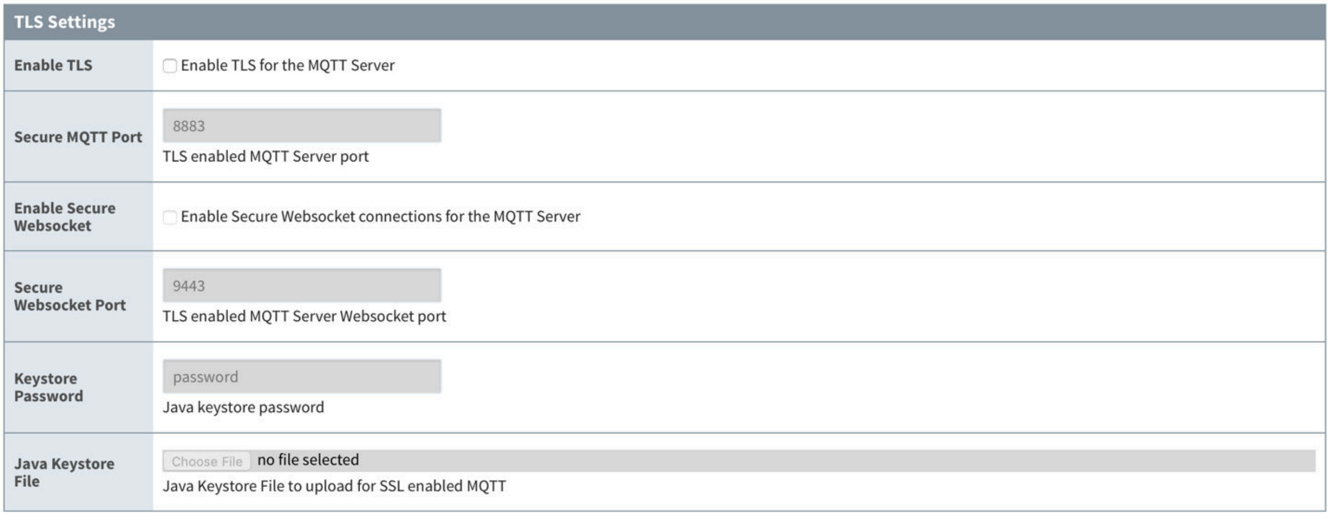
Main	
Enabled	☑ Enable the MQTT Server

Figure 33: MQTT Distributor Settings section 1



Non-TLS Settings	
Enable TCP	☑ Enable plain TCP connections for the MQTT Server
Port	1883 Non-TLS MQTT Server port
Enable Websocket	☑ Enable Websocket connections for the MQTT Server
Websocket Port	8090 Non-TLS MQTT Server Websocket port

Figure 34: MQTT Distributor Settings section 2



TLS Settings	
Enable TLS	□ Enable TLS for the MQTT Server
Secure MQTT Port	8883 TLS enabled MQTT Server port
Enable Secure Websocket	□ Enable Secure Websocket connections for the MQTT Server
Secure Websocket Port	9443 TLS enabled MQTT Server Websocket port
Keystore Password	password Java keystore password
Java Keystore File	Choose File no file selected Java Keystore File to upload for SSL enabled MQTT

Figure 35: MQTT Distributor Settings section 3

5.5 Step 6: Observe MQTT data

With Ignition running and the Distributor and Engine loaded, now we can open the Ignition Designer to observe some of the MQTT topology. Regardless of the OS Ignition is running on, there is a “Launch Designer” button on the Ignition Gateway Console. From here you can launch your Designer on any machine. This is shown below. The default credentials for the designer are the same as the Gateway Console, admin/password. Once you have logged into the Designer enter a new project name and open the project. The project name that we used for this tutorial is simply called “Tutorial1”.

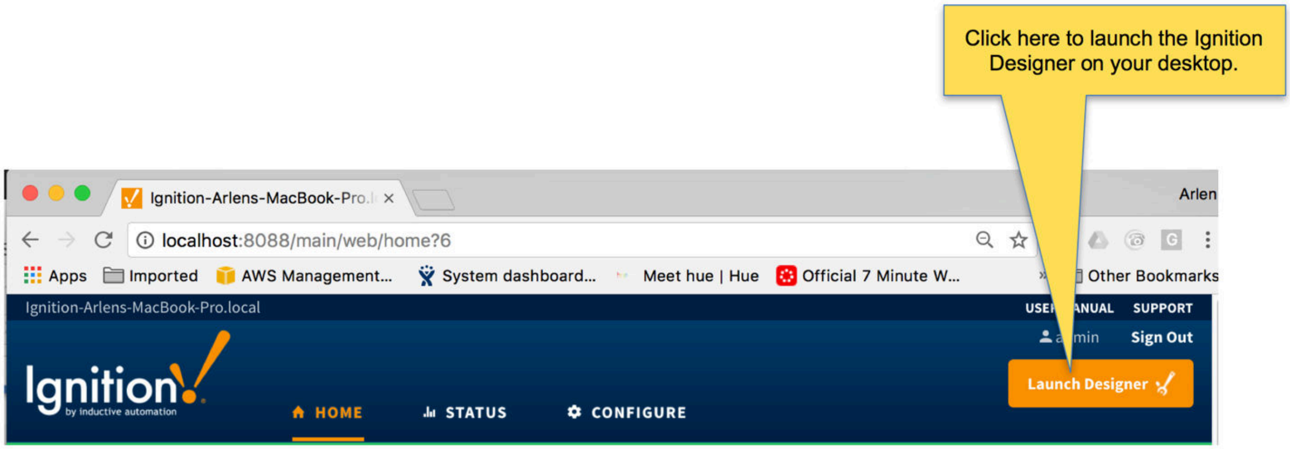

Click here to launch the Ignition Designer on your desktop.

Figure 36: Link to launch Ignition Designer

After the Designer opens, you will see the default Designer screen as shown below.

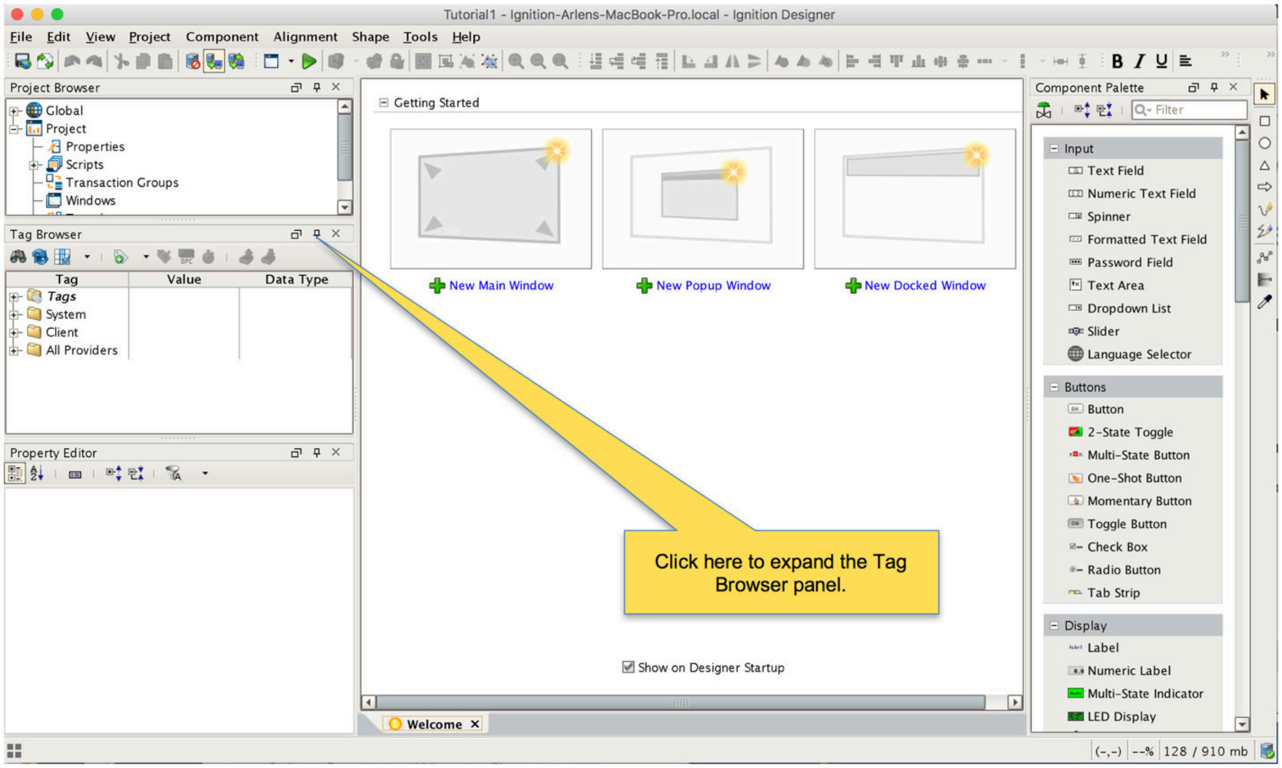
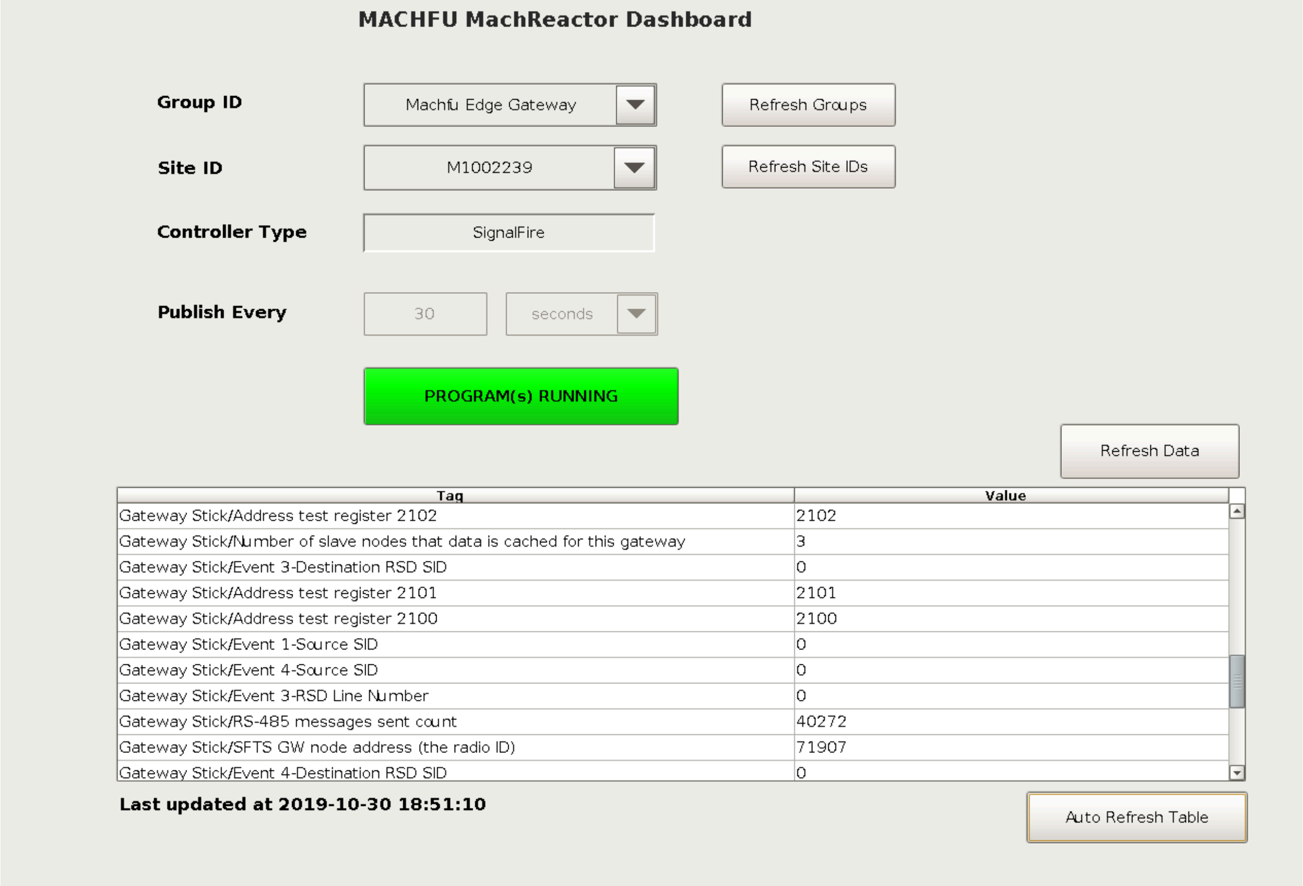

Click here to expand the Tag Browser panel.

Figure 37: Default Designer Screen

Once the MQTT Engine module is installed in Ignition, a new “Tag Provider” folder is created called “MQTT Engine”. This MQTT Engine folder will contain both process variable tags from field devices as well as metrics and diagnostics information from the MQTT infrastructure itself. At this point in the tutorial, we can use the tag browser to start looking at the simple MQTT infrastructure that we have running already. In the tag structure below using the default MQTT Distributor and MQTT Engine configurations, you can see the metrics provided for the MQTT client connection from the MQTT Engine to the MQTT Distributor.



MACHFU MachReactor Dashboard

Group ID: Machfu Edge Gateway

Site ID: M1002239

Controller Type: SignalFire

Publish Every: 30 seconds

PROGRAM(s) RUNNING

Tag	Value
Gateway Stick/Address test register 2102	2102
Gateway Stick/Number of slave nodes that data is cached for this gateway	3
Gateway Stick/Event 3-Destination RSD SID	0
Gateway Stick/Address test register 2101	2101
Gateway Stick/Address test register 2100	2100
Gateway Stick/Event 1-Source SID	0
Gateway Stick/Event 4-Source SID	0
Gateway Stick/Event 3-RSD Line Number	0
Gateway Stick/RS-485 messages sent count	40272
Gateway Stick/SFTS GW node address (the radio ID)	71907
Gateway Stick/Event 4-Destination RSD SID	0

Last updated at 2019-10-30 18:51:10

Figure 38: Ignition Dashboard

6 Appendix B: MQTT Topic Structure

MQTT Topics are structured in a hierarchy like the folders and files in a file system using the forward slash (/) as a delimiter.

By using the hierarchical system mentioned above, the RPC structure acquires a user friendly self-descriptive naming convention.

Topic names are:

- Case sensitive
- use UTF-8 strings.
- Must consist of at least **one valid character**.

The default SignalFire Gateway Stick topics for JSON encoding are structured in the following way:
Group ID/
Node ID/
GatewayStick/

The default SignalFire Remotes topics for JSON encoding are structured in the following way:
Group ID/
Node ID/
[Remote] /
Slave ID/

The default SignalFire topics for SparkPlug encoding are structured in the following way:
Group ID/
Node ID/
SignalFire/Gateway Stick/ [Remote + Slave ID] /MachReactor/
Program/
Value/
Enabled

REVISION HISTORY

Revision	Description	Date
0.1	Initial release	10/21/2019
1.0	Second release for review	10/31/2019
1.1	Connectivity to the Cloud	11/14/2019
2.0	Phase 1 release	10/27/2020
2.1	Added more updates	01/20/2021
2.2	Added SignalFire branded images	02/04/2021
3	Revisions per customer feedback	03/08/2021

Technical Support and Contact Information

SignalFire Telemetry
140 Locke Dr., Suite B
Marlborough, MA 01749
(978) 212-2868
support@signal-fire.com