ELPRO Technologies
29 Lathe Street
Virginia QLD 4014
Australia
tel: +61 7 3352 8600
www.elpro.com.au

MQTT Gateway Configuration Guide

Contents

Introduction

The ELPRO MQTT gateway provides the user with a highly configurable, powerful and easy to use protocol gateway to allow the transport of field process input or output variables using MQTT or MQTT with SparkplugB extensions.

The MQTT Gateway feature is available either through firmware upgrade or in new units for 215U-2-BGN, 415U-E-Cx, 415U-2-Cx and 915U-2 units.

For the application each process variable can be linked to a MQTT topic and payload.   Process variable can be sourced from/to local inputs, outputs, or external PLC/SmartSensors/Controllers via Modbus TCP or RTU internal gateway.

The MQTT/SparkplugB data can there be transported to the broker/server via licenced/unlicenced radio, WiFi (802.11) or 4G/LTE cellular depending on application requirements.

Designed with industrial applications in mind, redundancy and data security are important with the facility for up to 4 server/brokers to be configured allowing redundancy. Further data protection is available by using built in message queuing (historian store & forward) which will store event data if the unit is in communication failure with the broker/server and will automatically send stored data when the link is reinstated.

The sections below provide a guide to the system design, configuration and diagnostics when using the ELPRO MQTT gateway with standard MQTT protocol or with MQTT-SparkplugB applications.

ELPRO products that support MQTT gateway as outlined below:

Condor 415U-2
150/450/950MHz

Condor 415U-E
150/450/950MHz

Condor 215U-2
WiFi™ IO

ELPRO 915U-2
900MHz FHSS

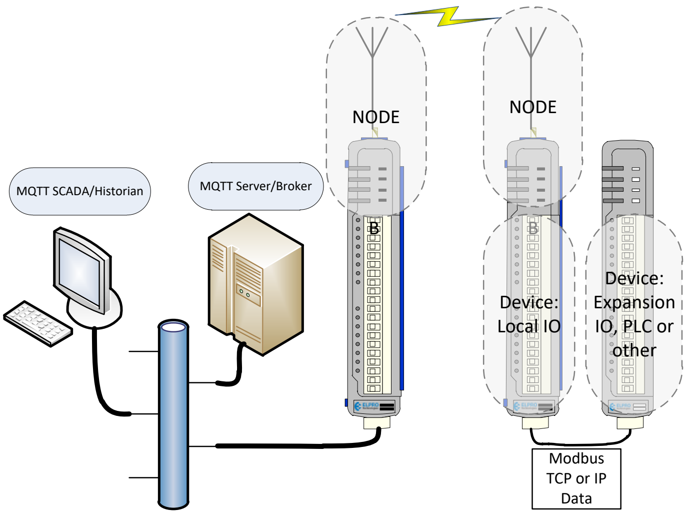
System Design

A typical application using MQTT will consist of a central server or broker and connected clients or nodes. Each of the client nodes in the typical industrial automation system will be connected to a mix of SCADA or Historian, Enterprise systems or various devices such as sensors, actuators, PLCs, controllers in the plant. Many standard industrial field devices do not support MQTT protocol directly and require a gateway to provide connectivity into MQTT based systems.

The gateway unit is comprised of the MQTT protocol gateway, the Node and the connected devices. The node will provide the MQTT or MQTT with Sparkplug protocol support, connectivity status/statics, diagnostics and system information.

Devices are connected to the Node either physically for external PLCS, smart sensors, etc or logically for local IO.

Typical system over is presented in the diagram below:



The ELPRO wireless gateway can be conveniently located and connected directly to the IO points or field PLCs, controllers, or smart sensors in the network. The ELPRO wireless network transports the MQTT data through the network to the broker as standard IP protocol.

This provides a powerful and robust system where the MQTT gateways can be distributed across the network very easily thus simplifying the configuration and removing the central gateway as a single point of failure.

The ELPRO MQTT gateway can operate in two protocol modes which allow for use of standard or plain MQTT or to use MQTT with SparkplugB.

The configuration of the MQTT gateway allows for the setup up to 4 broker/servers, MQTT topic/payload structure, MQTT plain or MQTT SparkplugB and message store & forward (queuing). The connected devices can also be setup to support the reading of inputs or controlling outputs in process systems.

The ELPRO MQTT gateway is integrated in the ELPRO industrial wireless units for 215U-2, 415U-2, 415U-E and 915U-2. This gateway can be installed as a simple firmware upgrade for 215U, 415U and the 915U-2. Note that the 915U-2 does not support TLS (SSL).

The MQTT gateway can be used in conjunction with the existing Modbus RTU/TCP gateway to interface to external PLCs, pump controllers, smart sensors, and large variety of other industrial devices.

Standard MQTT and MQTT Sparkplug

Standard MQTT is a very simple and lightweight publish/subscribe protocol which is event-based operating via a central broker or server. MQTT devices that have data to send, is published to the broker using a topic. Users or consumers of this data will subscribe to the topic and will then receive updates on any changes of the source data.

This protocol is very useful in gathering real time information across large networks in the Information Technology (IT) systems. For industrial applications there is often the need to not just monitor but also control plant. For these Operational Technology (OT) applications there is a need for more structure. Sparkplug provides OT centric namespace topic definition and IT centric payloads with MQTT state management.

Sparkplug provides:

- Simple, light weight and flexible event driven communications for OT applications
- State management with extensions for Birth, Death, Command and Data message types
- Auto discover of tags
- Store and forward (queuing) of data when communications is lost to allow backfilling of data
- Open standard

MQTT with sparkplug bridges the gap between IT and OT and thus allow increased access to data for all collection systems and enabling intelligent control of plant and equipment.

MQTT Message, Topic Structure and Payload Format

MQTT protocol messages consist of a Topic and a Data Payload. The topic is a unique descriptive index which also allows grouping. The topic uses a special character '/' to separate layers within the structure which can be configured to group functions.

The payload is a JSON based description of the message data. JSON is a data interchange format that uses human readable text to transport objects which consist of an attribute and value pairs.

MQTT is a very open standard and this does allow a huge amount of flexibility in the Topic and Payload structures. If this application is using standard MQTT then it's recommended to plan out this structure to allow simple access to data by using topics to group elements which have natural commonality.

The ELPRO MQTT gateway provides configurable, multilevel MQTT topics for use with standard MQTT and Sparkplug. Topic structure includes elements related to the Node and any connected devices.

Topic structure for plain MQTT will follow similar style to that of Sparkplug (EoN and device) which allows very simple switching between MQTT-plain and MQTT-Sparkplug protocols.

Topic namespace and payload names are case sensitive.

Suggested MQTT gateway Topic Structure:

[Owner]/[Group]/[Node]/[Device]/[configurable]

Payload structure:

{"timestamp":[linux EPOCH time ms], "DataValueLabel":value}

Note that for each payload there can be multiple time stamped DataValueLabel/value sets with a single message transmission.

Standard (Plain) Topic and payload

Configured for standard MQTT the gateway uses a similar format and structure to sparkplug operation and includes timestamp and data values. See MQTT example below.

Topic: ELPRO/FLOOD/GATEWAY/Georges Crossing/Register

Payload: {"timestamp":954711743792, "River Level":17.61, "Battery Voltage":13.8356}

Sparkplug Topic and payload

Sparkplug is commonly used in OT applications and provides significant advantages with a state-based protocol additions when working with industrial devices.

Sparkplug defines the structure of the topic (Namespace) and the payload which includes message timestamp. The metrics are structure to include elements for timestamp, data point name, an alias reference, data type and value. The final part of the payload is a message sequence number with sequence 0 reserved for special messages.

The payload is encoded as part of the Sparkplug standard.

The example used below here is for a sparkplugB application which will have the structure of spBv1.0/<Group ID>/<message type>/<Node ID>/<Device>

<Group ID>: Can be considered the name of the system or application.

<message type>: Defined as the type of message which can be a Birth, Death, Data or CMD (command) from Node or Device.

<Node ID>: This is the name or identification of the edge of network (EoN) connection. For the MQTT gateway here the edge of network can be via the ELPRO wireless or the Ethernet port depending on the network configuration.

<Device ID>: Device name or identification of a sensor, PLC, local inputs/outputs which is logically (local inputs/outputs) or physically connected to the Node.

Sparkplug Example:

Topic: spBv1.0/ELPRO-WaterCo/DDATA/LatheSt Res/Local Inputs

Payload JSON:

```
{"timestamp":947124413831,"metrics":[
        {"name":"IO-Digital/Pump Running","alias":10002,
        "timestamp":947124413831,
        "dataType":"Boolean","value":true}],"seq":67}
```

SparkplugB defines the data types, JSON structure and encoding of payloads.

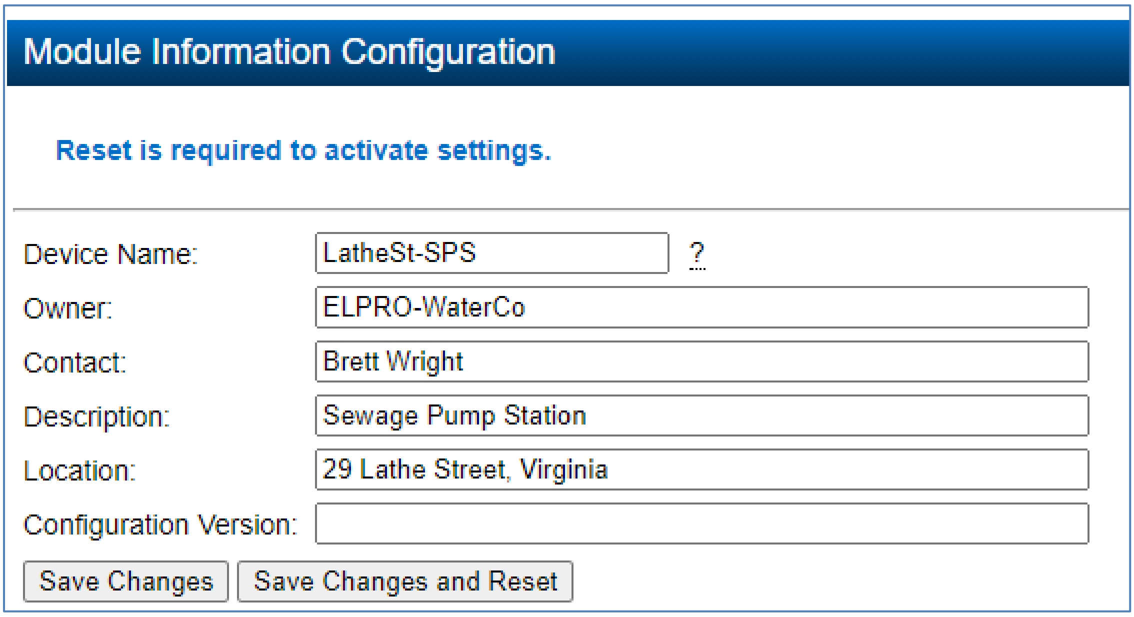
Gateway Configuration

Module Information

Once the ELPRO wireless units' normal networking and configuration is setup via the Quick Start menu, the MQTT gateway and IO can be configured.

Start by selecting the Module Information at the bottom of the web menu under the Unit Information section. The Module Information is used to provision the initial basic configuration items for the unit.

The MQTT gateway will use some of the items here to form the basis of the topic to be used. Other items that will be reported are parts of the unit's "System Info" on startup as a "Birth Message" and on Node updates.



Module Information Configuration

Reset is required to activate settings.

Device Name:	LatheSt-SPS ?
Owner:	ELPRO-WaterCo
Contact:	Brett Wright
Description:	Sewage Pump Station
Location:	29 Lathe Street, Virginia
Configuration Version:	

Device Name: This is used as the Node ID in the topic. Not to be confused with MQTT Device ID which is configured in the MQTT web page for Sparkplug operation.

Owner: Used as the Group ID in the topic for Sparkplug operation.

Contact: Reported in the Node information message.

Description: Site description. reported in the Node information message.

Location: Address, location information or even google maps short cut is useful.

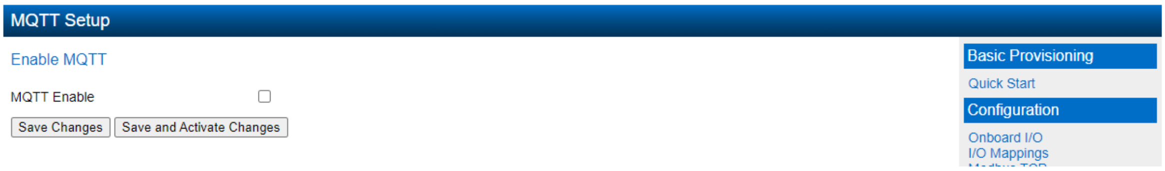
Configuration Version: Free to use, not reported by unit.

MQTT Broker Configuration

MQTT Nodes communicate with a central server or broker. Each MQTT node in the network needs to be connected to this Broker which manages the published and subscribed communications flows by Nodes in the system.

The connection to the broker is made using a transport control protocol, either TCP or TLS (SSL). MQTT message are transported through this connection. For additional security when using plain TCP protocol over a public network (cellular) it's recommended to use a secure tunnel for this connection which can be OpenVPN, IPSec, GRE or similar.

The configuration of the broker setting is accessed through the MQTT Setup web page in the configuration menu.



MQTT Setup

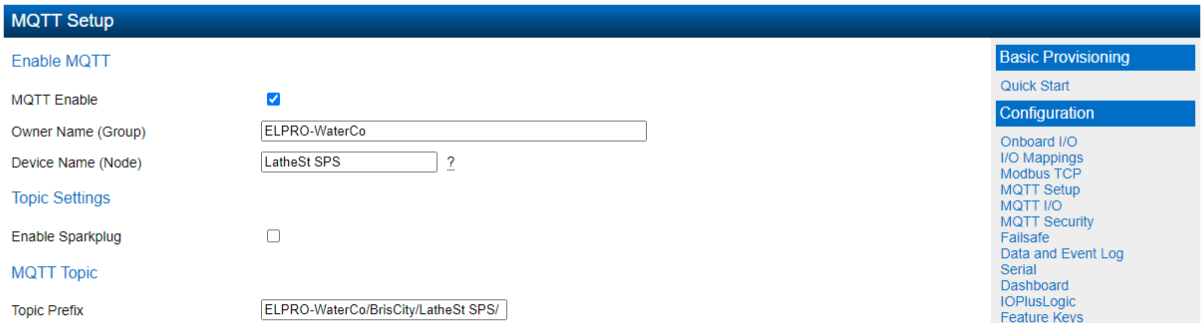
Enable MQTT

MQTT Enable □

MQTT gateway is enabled by checking the MQTT Enable on the MQTT Setup Configuration webpage menu. Then you can either select Sparkplug to enable SparkplugB MQTT extension or leave it off to use standard MQTT. Depending on the requirements of the application.

Owner Name (Group) and Device Name (Node) configuration items are pulled from the Module Information as entered above. Further edits can be made here if required.

When using standard MQTT, first configure the MQTT TOPIC prefix, then any attached logical devices and broker connection. The MQTT Topic Prefix can take any form using / symbol for logical system separations. It is useful to use a topic prefix that includes Group Name and Node name. Additional devices will be added to this topic tree.



MQTT Setup

Enable MQTT

MQTT Enable ☑

Owner Name (Group) ELPRO-WaterCo

Device Name (Node) LatheSt SPS ?

Topic Settings

Enable Sparkplug □

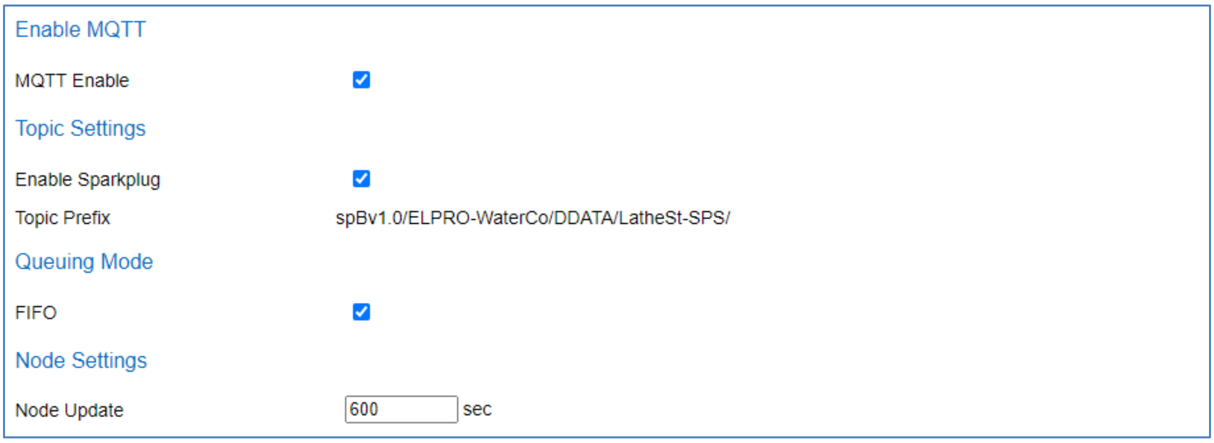
MQTT Topic

Topic Prefix ELPRO-WaterCo/BrisCity/LatheSt SPS/

With sparkplug enabled, the topic structure will be automatically populated to conform with the standard, “spBv1.0/GROUP/STATE/NODE”. For the ELPRO gateway:

- GROUP: Uses the units Owner, configured in Module Information.
- NODE: Uses the units Device Name, configured in Module Information Configuration or through quick start.
- STATE: is not configurable and part of the sparkplug standard.

Start from the top of the Web page by configuring the basic MQTT parameters. See example page below.



Enable MQTT

MQTT Enable ☑

Topic Settings

Enable Sparkplug ☑

Topic Prefix spBv1.0/ELPRO-WaterCo/DDATA/LatheSt-SPS/

Queuing Mode

FIFO ☑

Node Settings

Node Update 600 sec

Enable MQTT: Select this tick box to enable the MQTT gateway.

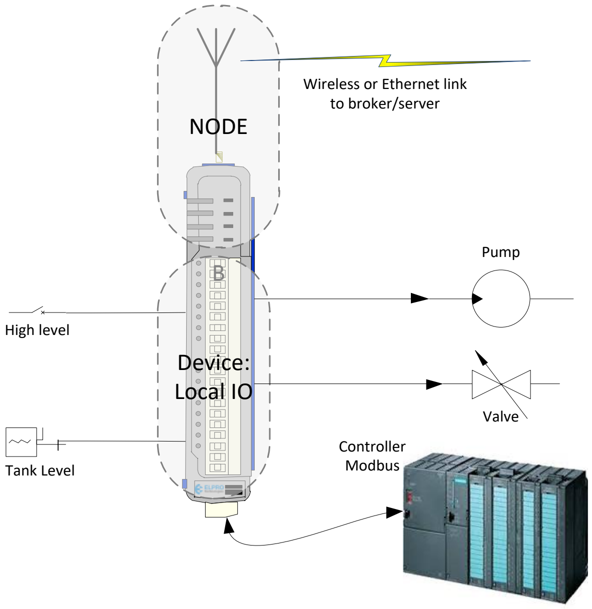
Enable Sparkplug: Select this tick box if you are using a SCADA or host visualization Software package, or any other Sparkplug B enabled connection.

MQTT Topic: Configuration box is shown when sparkplug is not enabled to allow configuration of the namespace topic prefix which will be used as base for topic in this gateway. For example, “Group Name/Node Name” is a good format to follow. Input configuration table will provide the final part of topic used to send a data packet.

Queuing Mode: Queuing or Store-and-Forward historization is used by the gateway to store messages if there is a break in communications channel with the broker or server. To adjust the emptying mode for Queuing or Store-and-Forward historization, select either FIFO (First In First Out) or LIFO (Last In First Out).

Device Configuration

Device Configuration below provides additional grouping to be provided within the topic structure for local IO, diagnostics and external devices connected to the MQTT node. Device Name is used in the MQTT IO configuration to select the IO directly for publishing (inputs) or subscription (outputs).

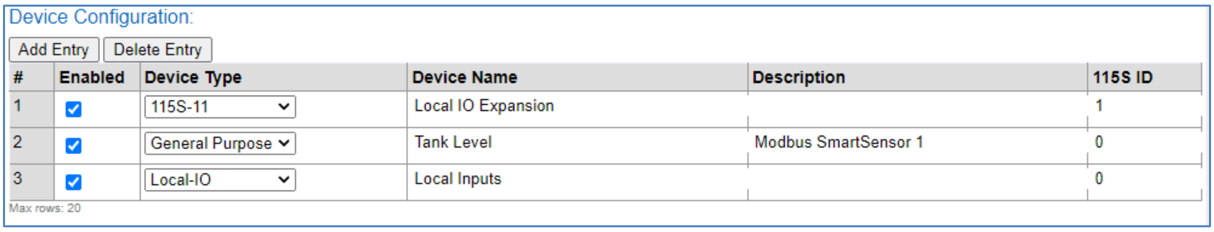


In diagram above it can be seen that the Node provides the connectivity to the broker and there are several devices connected to the Node such as direct inputs and outputs via on-board IO and remote Controller via Modbus gateway. In this example, the local onboard IO could be broken up into three logical devices.

1. Direct inputs for monitoring tank level and high-level float switch
2. Direct outputs to control Pump and proportional valve.
3. Modbus connection to external PLC controller

Using the device configuration table, the application can be broken into natural groupings by function which will link back into the controller application such as SCADA in an organized way.

The configuration for each of the device definitions is completed using the table below:



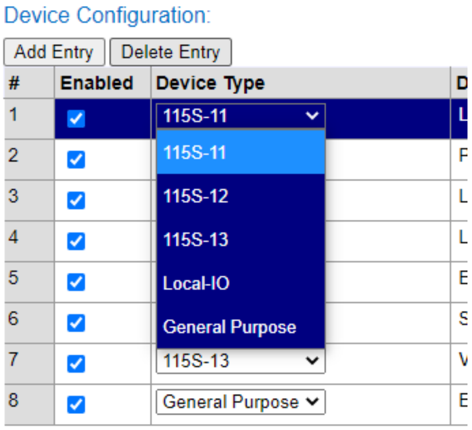
Device Configuration:

#	Enabled	Device Type	Device Name	Description	115S ID
1	☑	115S-11	Local IO Expansion		1
2	☑	General Purpose	Tank Level	Modbus SmartSensor 1	0
3	☑	Local-IO	Local Inputs		0

Max rows: 20

Typical devices are the local inputs and outputs of the wireless unit and external controllers or sensors connected via a Modbus gateway. The user can also use these device configurations to logically separate function into a separate topic stream.

Device Type can be selected to be 115S-11, 115S-12, 115S-13, Local-IO or General Purpose. Device configuration pull down menu can be seen below.



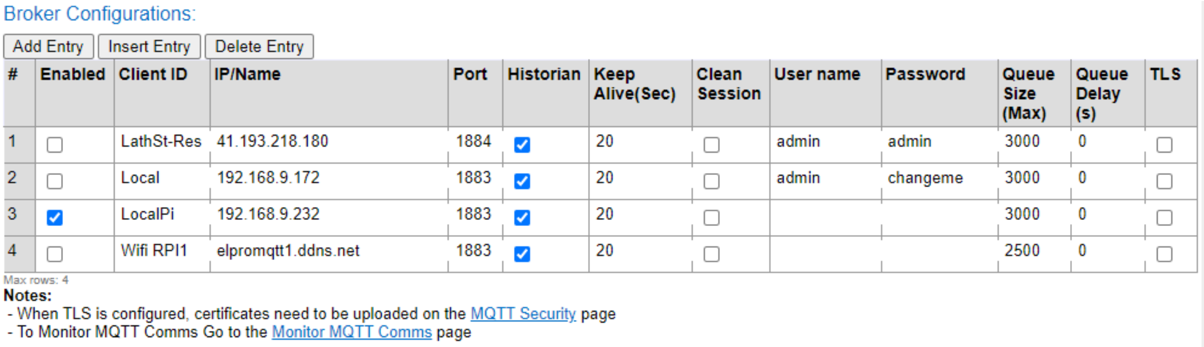
Device Configuration:

#	Enabled	Device Type	D
1	☑	115S-11	L
2	☑		F
3	☑		L
4	☑		L
5	☑		E
6	☑		S
7	☑	115S-13	V
8	☑	General Purpose	E

For 115S Modbus serial expansion units a slave address must be configured, on the far right of the configuration line. Other Device Types do not need a slave address and this should set to zero.Once configured, the device names will now be available for selection in the MQTT IO configuration (see below).

Broker Configuration

The broker configuration table allows multiple broker configurations to be made for up to 4 broker/server connections (maximum). Brokers can be configured using either hostname or IP address. Connections can be made with simple username and password or with security (TLS) by uploading certificate data (see MQTT security page below).



Broker Configurations:

#	Enabled	Client ID	IP/Name	Port	Historian	Keep Alive(Sec)	Clean Session	User name	Password	Queue Size (Max)	Queue Delay (s)	TLS
1	☐	LathSt-Res	41.193.218.180	1884	☑	20	☐	admin	admin	3000	0	☐
2	☐	Local	192.168.9.172	1883	☑	20	☐	admin	changeme	3000	0	☐
3	☑	LocalPi	192.168.9.232	1883	☑	20	☐			3000	0	☐
4	☐	Wifi RPI1	elpromqtt1.ddns.net	1883	☑	20	☐			2500	0	☐

Max rows: 4

Notes:
- When TLS is configured, certificates need to be uploaded on the MQTT Security page
- To Monitor MQTT Comms Go to the Monitor MQTT Comms page

Enabled: Enable this broker connection. Can be used to disable inactive broker connections.

Client ID: Broker Client ID. This is required if the broker/server has client IDs enabled in its configuration. This MUST be unique in this broker . If broker/server has client IDs disabled, then it can be blank.

IP/Name: Broker/server hostname (DNS) or IP address.

Port: TCP port to use on the broker. Default is 1883.

Historian: Enables tagging of messages when they have been stored during a period when gateway is disconnected from broker/server.

Keep Alive: Connection check (PING) to the broker/server. For fast connections (ethernet, WiFi or cellular) this can be short, 20-60 seconds. For slower radio links then use a larger value of 180-600 seconds. If the broker/server does not receive within the configured keep alive time, then it will disconnect and send the configured last will and testament. Note: If using Cellular keep in mind that using fast keep alive times will increase overall cellar data throughput.

Clean Session: When enabled the broker will remove all configuration, subscribed topics and messages (QoS 1 or 2) when the link is lost to the MQTT gateway. Leave off to preserve data for a persistent data configuration.

Username: Broker/server connection username to use.

Password: Broker/server connection password to use.

Queue Size: The maximum size of the message queue when unit is storing messages. The maximum number of messages is 10000 and shared across all configured brokers.

Queue Delay: Upon broker/server communication reconnection after an outage this setting will limit the rate that messages are sent out over the network. For radio networks this setting should be set to stop unit from swamping network with radio traffic. A typical value might be 5 seconds for a low-speed radio network.

TLS: Check this box to make the broker connection using Transport Layer Security (TLS). Note certificate are required to be entered into the MQTT Security web page.

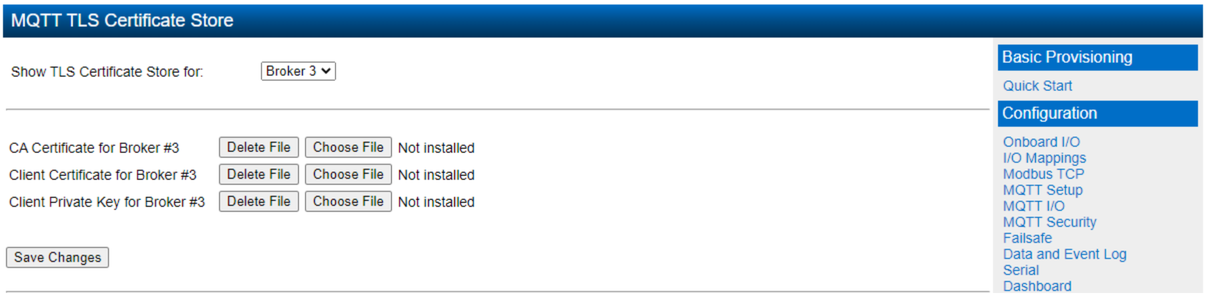
Once the configuration is completed, press save and activate to store the configuration. The gateway will immediately attempt to make a connection to the broker. The status of this connection can be viewed by accessing the connectivity web page (or through registers) and broker connections will be found at the bottom of this page below any wireless connections.

MQTT TLS Certificate Store Configuration

If the MQTT or Sparkplug broker/server requires TLS or SSL security for the connection, then the TLS checkbox in the broker configuration must be checked.

The correctly configured SSL certificates also need to be loaded into the unit using the MQTT TLS Certificate Store to allow the unit to make the secure connection. Click on “MQTT Security” on the right menu to access the MQTT TLS Certificate store.

You should contact your IT department to provide the required CA Certificate, Client Certificate and Client Private Key files to be loaded into the unit below.



MQTT TLS Certificate Store

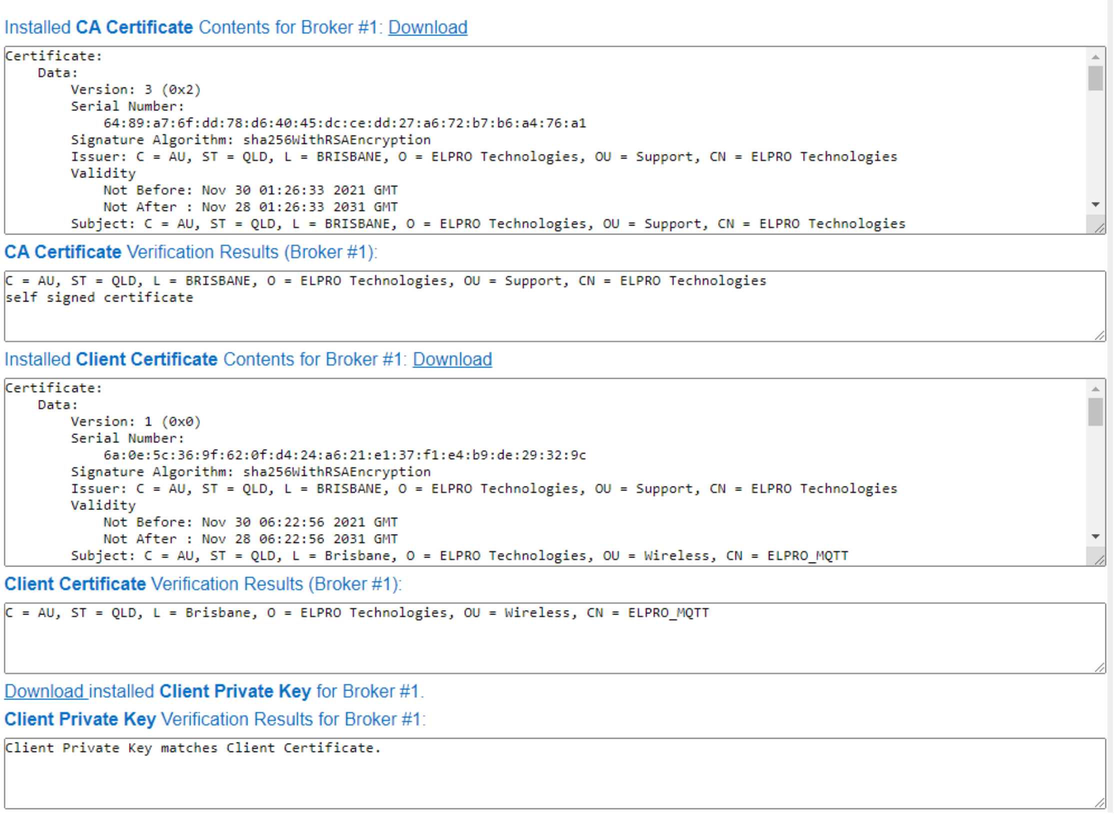
Show TLS Certificate Store for: Broker 3

CA Certificate for Broker #3 — Not installed

Client Certificate for Broker #3 — Not installed

Client Private Key for Broker #3 — Not installed

Once the files are loaded then click on “Save Changes” and the unit will perform some basic checks to validate the certificates. If this is successful there you will see the following diagnostic output showing no errors as below.



Installed **CA Certificate** Contents for Broker #1: Download

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            64:89:a7:6f:dd:78:d6:40:45:dc:ce:dd:27:a6:72:b7:b6:a4:76:a1
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = AU, ST = QLD, L = BRISBANE, O = ELPRO Technologies, OU = Support, CN = ELPRO Technologies
        Validity
            Not Before: Nov 30 01:26:33 2021 GMT
            Not After : Nov 28 01:26:33 2031 GMT
        Subject: C = AU, ST = QLD, L = BRISBANE, O = ELPRO Technologies, OU = Support, CN = ELPRO Technologies
```

CA Certificate Verification Results (Broker #1):

```
C = AU, ST = QLD, L = BRISBANE, O = ELPRO Technologies, OU = Support, CN = ELPRO Technologies
self signed certificate
```

Installed **Client Certificate** Contents for Broker #1: Download

```
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number:
            6a:0e:5c:36:9f:62:0f:d4:24:a6:21:e1:37:f1:e4:b9:de:29:32:9c
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = AU, ST = QLD, L = BRISBANE, O = ELPRO Technologies, OU = Support, CN = ELPRO Technologies
        Validity
            Not Before: Nov 30 06:22:56 2021 GMT
            Not After : Nov 28 06:22:56 2031 GMT
        Subject: C = AU, ST = QLD, L = Brisbane, O = ELPRO Technologies, OU = Wireless, CN = ELPRO_MQTT
```

Client Certificate Verification Results (Broker #1):

```
C = AU, ST = QLD, L = Brisbane, O = ELPRO Technologies, OU = Wireless, CN = ELPRO_MQTT
```

Download installed **Client Private Key** for Broker #1.

Client Private Key Verification Results for Broker #1:

```
Client Private Key matches Client Certificate.
```

MQTT IO Configuration

Once the MQTT Broker Configuration is complete and necessary TLS Certificates are loaded, The MQTT IO Configuration can be completed.

IO Configuration table below allows the mapping of local inputs, diagnostics and connected devices (ALERT and ALERT2 data sources) to be mapped to Topic/payload MQTT data.

There are two configuration tables, first one allowing the configuration of inputs values to be published and second for outputs that are required to be set by creating subscriptions to broker.

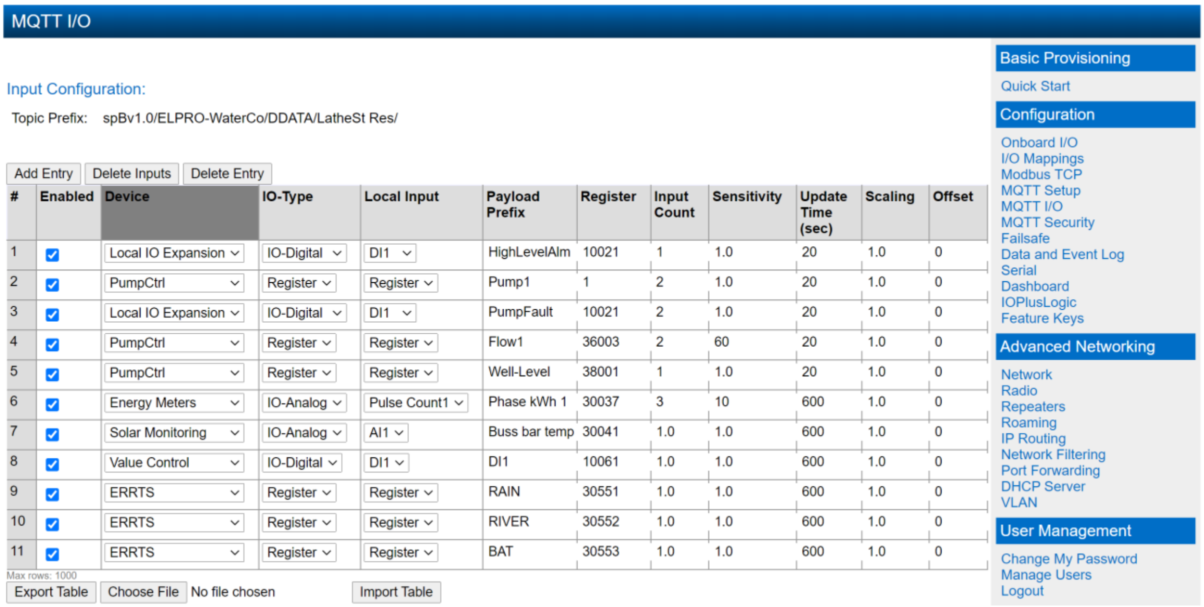
Configuring Inputs to Publish

Configure inputs to publish MQTT (or Sparkplug) data to a broker through the input configuration table as seen below. This input tables allows configuration of the topic and the source input (plain MQTT) or data and name (SparkplugB) of the payload to be sent to the broker.

The dark grey heading row of the Input Configuration table (pictured below) indicates configuration items for the Topic. On the right side is payload source data and name definition items. Selecting a row in the table will display the full topic below the table for reference.

When configuring multiple inputs to configure it’s a good practice to press save often as there is an activity timeout on configuration menus.

Input are configured using the Input Configuration table as below.



Input Configuration:

Topic Prefix: spBv1.0/ELPRO-WaterCo/DDATA/LatheSt Res/

#	Enabled	Device	IO-Type	Local Input	Payload Prefix	Register	Input Count	Sensitivity	Update Time (sec)	Scaling	Offset
1	☑	Local IO Expansion	IO-Digital	DI1	HighLevelAlm	10021	1	1.0	20	1.0	0
2	☑	PumpCtrl	Register	Register	Pump1	1	2	1.0	20	1.0	0
3	☑	Local IO Expansion	IO-Digital	DI1	PumpFault	10021	2	1.0	20	1.0	0
4	☑	PumpCtrl	Register	Register	Flow1	36003	2	60	20	1.0	0
5	☑	PumpCtrl	Register	Register	Well-Level	38001	1	1.0	20	1.0	0
6	☑	Energy Meters	IO-Analog	Pulse Count1	Phase kWh 1	30037	3	10	600	1.0	0
7	☑	Solar Monitoring	IO-Analog	AI1	Buss bar temp	30041	1.0	1.0	600	1.0	0
8	☑	Value Control	IO-Digital	DI1	DI1	10061	1.0	1.0	600	1.0	0
9	☑	ERRTS	Register	Register	RAIN	30551	1.0	1.0	600	1.0	0
10	☑	ERRTS	Register	Register	RIVER	30552	1.0	1.0	600	1.0	0
11	☑	ERRTS	Register	Register	BAT	30553	1.0	1.0	600	1.0	0

Max rows: 1000

No file chosen

Detailed description of each configuration item is below:

Enabled: Tick box used to enable or disable MQTT configuration line.

Device: Device name is configured in the MQTT broker page and is selected from this pull down. The Device is added to the topic prefix at top of table to form the overall topic to this payload.

IO-Type: The source inputs or registers used for the payload are selected by IO-Type and Local Input name. The IO-Type is used to select the I/O type and is dependent on the device table configuration. Below is a table of the available IO-Type for each device type. There are other derived inputs available which can be accessed directly using the Register, IO-Type setting.

Device Type	IO-Type
115S-11	IO-Digital, IO-Analog
115S-12	IO-Digital, IO-Analog
115S-13	IO-Digital
Local-IO	IO-Digital, IO-Analog, Diagnostics
General Purpose	Register

Local Input: The local input configuration is used to select the actual Input, output or register to source the input data to send in the MQTT message payload. Below is a table outlining each local input available for IO-Type setting.

MQTT Device Type Index Index	General Purpose Register	Local-IO			115S-11		115S-12		115S-13 IO-Digital
		IO-Digital	IO-Analog	Diagnostics	IO-Digital	IO-Analog	IO-Digital	IO-Analog	
0	Register	DI1	AI1(0-20mA)	Battery V	DI1	Pulse Count1	DI1	AI1	DI1
1		DI2	AI2(0-20mA)	Supply V	DI2	Pulse Count2	DI2	AI2	DI2
2		DI3	AI3(0-20mA)	RSSI	DI3	Pulse Count3	DI3	AI3	DI3
3		DI4	AI4(0-20mA)	Link Count	DI4	Pulse Count4	DI4	AI4	DI4
4		DI5	VSupply	Tx Fail	DI5	Pulse Rate1	DI5	AI5	DI5
5		DI6	24V	Link Uptime	DI6	Pulse Rate2	DI6	AI6	DI6
6		DI7	Vbatt	Link Status	DI7	Pulse Rate3	DI7	AI7	DI7
7		DI8	VExt	Channel Ultilization	DI8	Pulse Rate4	DI8	AI8	DI8
8			AI1(0-20V)	VSWR	DI9				
9			AI2(0-20V)		DI10				
10			AI3(0-5V)		DI11				
11			AI4(0-5V)		DI12				
12			PRate1		DI13				
13			PRate2		DI14				
14			PRate3		DI15				
15			PRate4		DI16				

Payload Prefix: Enter here the name of the input that is to be used in the MQTT message. If an input count of greater than 1 is used, then a count number will be appended to the payload name. For a payload name that contains a number as the last digit, then this input count will increment from this number. For example: Local Input is DI3 (digital input 3), Payload Prefix = Pump Running1 and count of 4 will produce 4 MQTT messages with payloads of Pump Running1, Pump Running2, Pump Running3 and Pump Running4, collecting their values from DI3 through to DI6 respectively.

Register: When “Local Input” is selected to any value other than “Register” this value will preload with the correct register value. For physical inputs on the unit this value cannot be changed. When “Local Input” is set to “Register” , this should be entered as the register required to be sent. Please refer to product user manual for a guide to available registers.

Note: For Sparkplug applications the register is used as the alias and therefore must be unique.

Input Count: Input Count allows a block of values to be referenced and sent together in a single MQTT message..

Sensitivity: The sensitivity is used for register or analog input values as the threshold required for this input to change before a message is transmitted. The sensitivity should be used carefully to manage the message traffic on the wireless network. If the sensitivity is set to zero, then changes will not be transmitted as value changes. The Update time can be used to send regular updates of the value. If a scale value is used then the sensitivity is applied after scaling.

Update Time (sec): Update time is used to allow a regular update messages to transmitted for the configured input value. If the update time is configured to zero then there will be no updates transmitted. The update time is particularly useful for analog or register values where these is a sensitivity configured to send a current value can capture any small changes inside the sensitivity threshold.

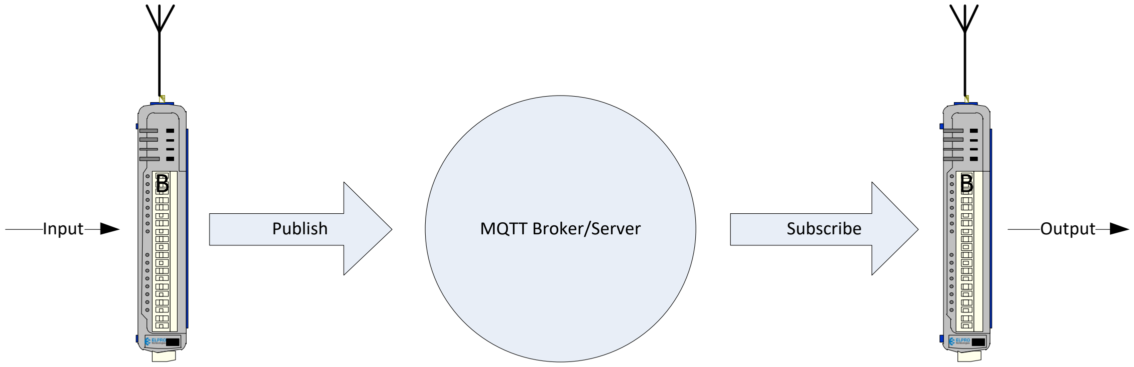
Scaling: This allows the value to be scaled before sending it as an MQTT message. A Scaling value of 1.0 sends the data unmodified using the value type as it is stored in the device. For a scaling value other than 1.0 the MQTT gateway will convert the value type to floating point and apply the scaling value.

Offset: This value is used to apply a fixed value offset to the input or register. If the Offset value is an integer (and the scaling value is 1.0) then the value type is not changed. If the Offset does use decimal places then the type is converted to a float automatically.

Below the Input Configuration table there are three buttons which allow the export and import of the table data using CSV formatted file. This can be useful where there is a large amount of data to enter. See separate section in this document for more detail on this feature.

Configuring Outputs to Subscribe

To configure outputs for use with MQTT or Sparkplug the unit needs to be setup with a subscribe topic and to have the output configured . The subscribe topic and payload data needs to match the input publish topic containing the value to be written to the output. If there is a mismatch in data type then the output will not be set, and an error will be logged in the system log (see below)



To allow for control with MQTT based SCADA systems using Sparkplug, the subscribe topic must include DCMD (publish for input would use DDATA) .

There is an option to automatically configure a publish topic on the output value. This allows the source unit to receive the state of the output once it is set to the new value. This option will default ”ON” when sparkplug is enabled and requires topic structure to include DCMD in the topic as indicated in example below.

Example for Sparkplug:

Subscribe Topic: spBv1.0/ELPRO-WaterCo/DCMD/LatheSt Res/Local-IO/

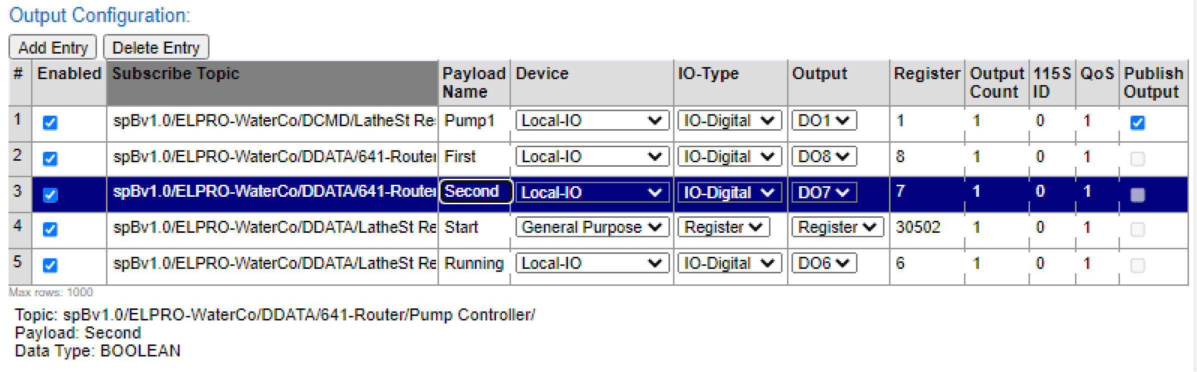
For Sparkplug applications which require a direct input to output sent (which is not though SCADA or other controller), use a topic that matches the input publish topic including the DDATA as shown in the example below. When using DDATA, the auto publish feature is not available, so output publish requires a separate Input Configuration to publish value of output.

Example for Sparkplug input to output direct:

Subscribe Topic:

spBv1.0/ELPRO-WaterCo/DDATA/641-1-Router/Pump controller

Output configuration table and description of each configuration item is below.



Output Configuration:

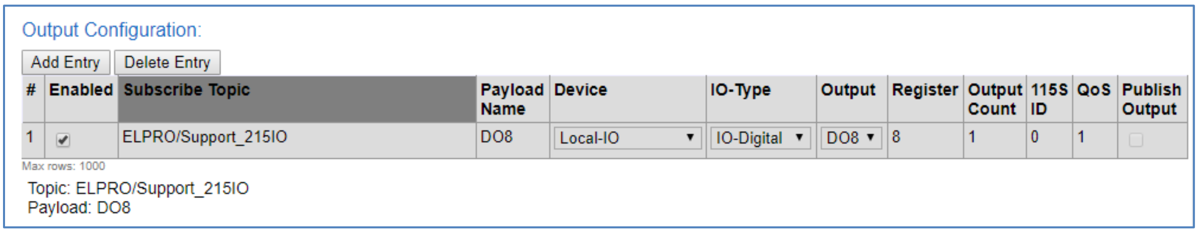
#	Enabled	Subscribe Topic	Payload Name	Device	IO-Type	Output	Register	Output Count	115S ID	QoS	Publish Output
1	☑	spBv1.0/ELPRO-WaterCo/DCMD/LatheSt Re	Pump1	Local-IO	IO-Digital	DO1	1	1	0	1	☑
2	☑	spBv1.0/ELPRO-WaterCo/DDATA/641-Router	First	Local-IO	IO-Digital	DO8	8	1	0	1	☐
3	☑	spBv1.0/ELPRO-WaterCo/DDATA/641-Router	Second	Local-IO	IO-Digital	DO7	7	1	0	1	☐
4	☑	spBv1.0/ELPRO-WaterCo/DDATA/LatheSt Re	Start	General Purpose	Register	Register	30502	1	0	1	☐
5	☑	spBv1.0/ELPRO-WaterCo/DDATA/LatheSt Re	Running	Local-IO	IO-Digital	DO6	6	1	0	1	☐

Max rows: 1000

Topic: spBv1.0/ELPRO-WaterCo/DDATA/641-Router/Pump Controller/
Payload: Second
Data Type: BOOLEAN

Example for Standard (Plain) MQTT

Subscribe Topic: mosquitto_pub -h elpromqtt1.ddns.net -t ELPRO/Support_215IO -m "{\"DO8\":on}"



Output Configuration:

#	Enabled	Subscribe Topic	Payload Name	Device	IO-Type	Output	Register	Output Count	115S ID	QoS	Publish Output
1	☑	ELPRO/Support_215IO	DO8	Local-IO	IO-Digital	DO8	8	1	0	1	☐

Max rows: 1000

Topic: ELPRO/Support_215IO
Payload: DO8

Output Configuration

Subscribe Topic: Enter the topic for the value that is published to the broker from another source.

Payload Name: For the topic that the output is subscribed this is the NAME of the value to be used to write to the output. For digital outputs, the values of 0 or 1, on or off and true or false are accepted.

Device: Logical device name in unit where the output is located.

IO-Type: Logical IO type of where of the output group to refine the output list.

Output: Select the output to be used.

Register: Internal register which can be overwritten if “Register” IO-Type is used. For physical outputs this is fixed.

Output Count: If the count is great than 1 then the payload name will auto-increment similar to the input configuration. For example, a count of 2 and payload Name of Pump3 will accept values of Pump3 and Pump4 to write to sequential outputs.

QoS: This must match the setting of the subscribed topic at source unit. Usually set to 1.

Publish Output: In Sparkplug applications where there is a SCADA connected there may be a requirement for the output state to be published back to the broker to confirm the output change. If this function is required, then select the “Publish Output” option in the configured output configuration row. For Sparkplug this will be enabled by default.

Queuing

Queuing or Historian store-and-forward is a mode that allows the remote node to be able to hold messages when there is a break in communications and then transmit these once communications is reestablished. This will allow the historical data to be “back filled” to prevent the loss of data.

The ELPRO queued data is stored in RAM memory and will be lost if there is a loss of power to the unit. If the application is mission critical, it is recommended to use an external back up battery which can be charged by the unit to retain power during AC failure.

When using queuing, consideration needs to be made to the speed of the network connection path to the broker. The queue delay is configured is seconds and allows for delivery of queued data to be rate limited for slower networks.

The MQTT gateway also uses a concentration process to consolidate data payloads for common topics into a single message. This significantly improves the efficiency and speed of the queue emptying process.

If the broker/server connection is via ethernet, 802.11 WiFi or cellular network, this is very fast and Queue delay can be set to zero. For slower links such as 415U a queue delay of 5 seconds is recommended to limit impact to normal system traffic.

If the Historian flag is ticked in the broker configuration, queued data will use the isHistorical flag and set to True during a loss of link with broker/server to indicate that data is not real time.

SSL Certificates Setup

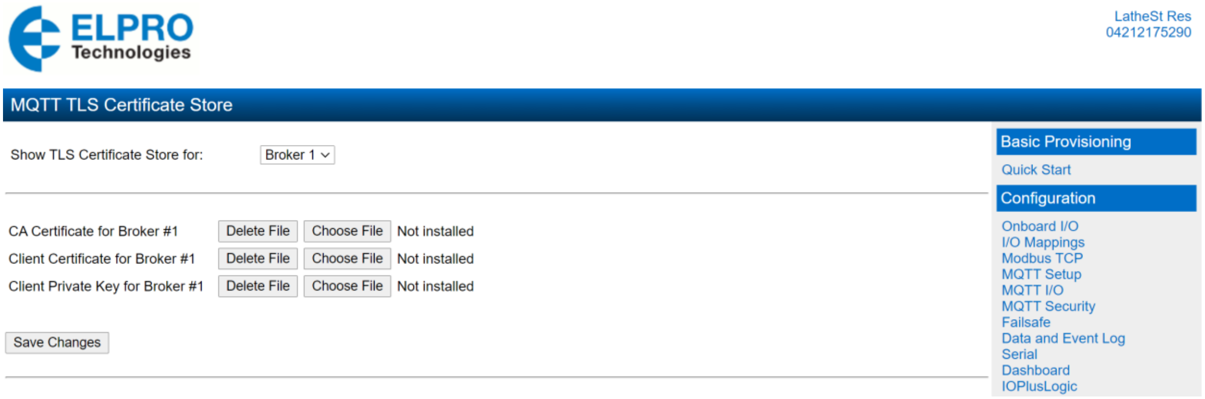
It is very important to consider security of data connection with brokers with any application. The ELPRO 215U, 415U and 915U off over the air wireless encryption using industry standard WPA2 (215U) or AES-256 encryption (415/915U). This encryption provides security of data transported over the network for the wireless links. In applications where the base radio is connected to a “on-premise” broker/server and this network connection is considered to be secure then additional security on the connection to the broker using TLS is usually not necessary.

In applications where the connection to broker/server is on a public network or additional security is required by policy then the ELPRO MQTT gateway can be configured to use a TLS connection.

To use TLS with the MQTT gateway broker/server connection there are 3 certifications required:

- Certificate Authority (CA)
- Client Certificate (using CA above)
- Client Private Key

These certificates and private key need to be provided in x.509 format as a file that can be uploaded into the unit. To upload the certificates, navigate to the MQTT Security web page as seen below.



MQTT TLS Certificate Store

Show TLS Certificate Store for: Broker 1

CA Certificate for Broker #1	Delete File	Choose File	Not installed
Client Certificate for Broker #1	Delete File	Choose File	Not installed
Client Private Key for Broker #1	Delete File	Choose File	Not installed

For each file first select the “Choose File” button, navigate to the folder which holds the certificates and select the correct certificate to load.

Once the CA, Client certificates and the private key is loaded the unit will preform some checks and report any errors found.

With the certificates loaded; go to the MQTT Setup page, configure the broker settings and select the TLS tick box to complete the connection setup. Then press the “Save and Activate” button to finalise the configuration. The MQTT gateway will then restart and immediately try to make a connection to the broker.

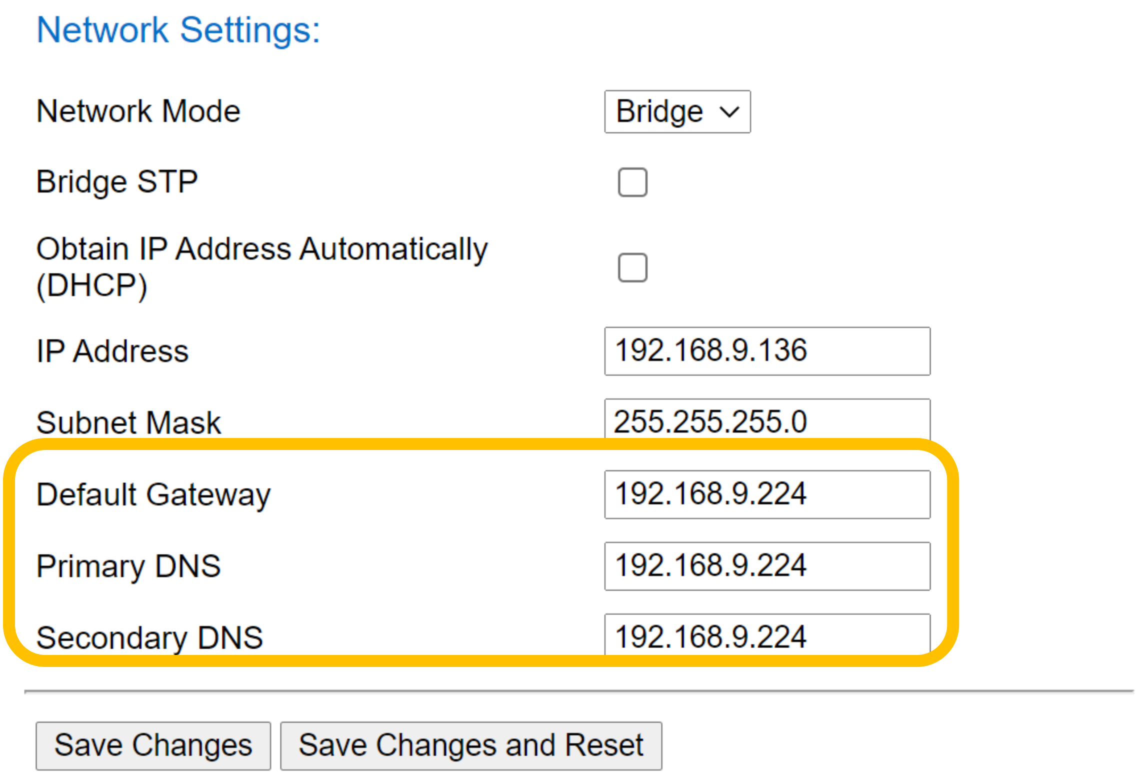
The status of the connection can be found on the “Connectivity” web page, in Network Diagnostics.

If the MQTT gateway fails to make a connection, then check the System log found in System Tools or Statistics pages for any TLS or SSL errors.

Domain Name Server Setup

If the MQTT application requires a domain name to be used, i.e. the broker is connected through a Dynamic cellular IP address then the Domain Name Server (DNS) will need to be entered under the IP/Name field in the broker configuration table to allow the ELPRO wireless unit to use the domain name instead of an IP address.

The configuration for the DNS setup can be found in the “Network” web page under Advanced Networking section. Configuration menu and will appear as below:



Network Settings:

Network Mode	Bridge
Bridge STP	□
Obtain IP Address Automatically (DHCP)	□
IP Address	192.168.9.136
Subnet Mask	255.255.255.0
Default Gateway	192.168.9.224
Primary DNS	192.168.9.224
Secondary DNS	192.168.9.224

For the unit to be able to cross reference the domain name configured, it must be able to make a connection to the to a Domain Name Server (DNS). There are three configuration items required:

1. Default Gateway IP address: This is the IP address of the gateway to network or Wide Area Network where the DNS is located
2. Primary DNS: IP address of the first DNS server to make a connection with.
3. Secondary DNS: IP address of the second DNS server to make connection with. If there is no secondary server address, then use same IP as primary.

Once this configuration is made then press "Save Changes and Reset” to restart the unit and initiate the connection to the DNS.

In the configuration example above there is a Cellular router providing the WAN connection for the MQTT gateway to the brokers. In this case this cellular router has been configured to provide a DNS.

If the unit with the MQTT gateway is connected to network where there is a gateway to the internet then google is a convenient option for DNS servers. Then use IP addresses 8.8.8.8 and 8.8.4.4 for primary and secondary DNS addresses.

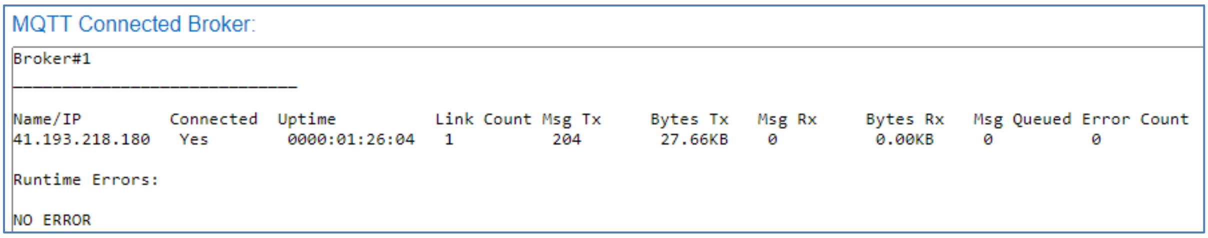
Diagnostics

The MQTT/Sparkplug gateway has a number of useful diagnostic features built into the unit to allow fast diagnosis of issues or facilitate site commission during the installation of new equipment or features.

Broker Connectivity Status

Connectivity web page will show the broker/server status and statistics at the bottom of page, below the wireless connection status.

Typical connectivity status is show below:



MQTT Connected Broker:

```
Broker#1
_____________________________

Name/IP         Connected  Uptime          Link Count Msg Tx     Bytes Tx   Msg Rx     Bytes Rx   Msg Queued Error Count
41.193.218.180   Yes        0000:01:26:04   1          204        27.66KB    0          0.00KB     0          0

Runtime Errors:

NO ERROR
```

This gives a view of the broker connection state, uptime, number of times the link has been re-established, Tx/Rx messages, Tx/Rx bytes, queue size and message error count.

The connectivity statistics are also available in local registers as well sent as part of the Node System Info/Diagnostics and will be sent as part of the Node status information send on Node Birth.

MQTT Broker Status Registers

Connectivity state and statistics are available as MQTT/Sparkplug Node diagnostics and through internal registers in the unit.

Below is an outline of internal registers:

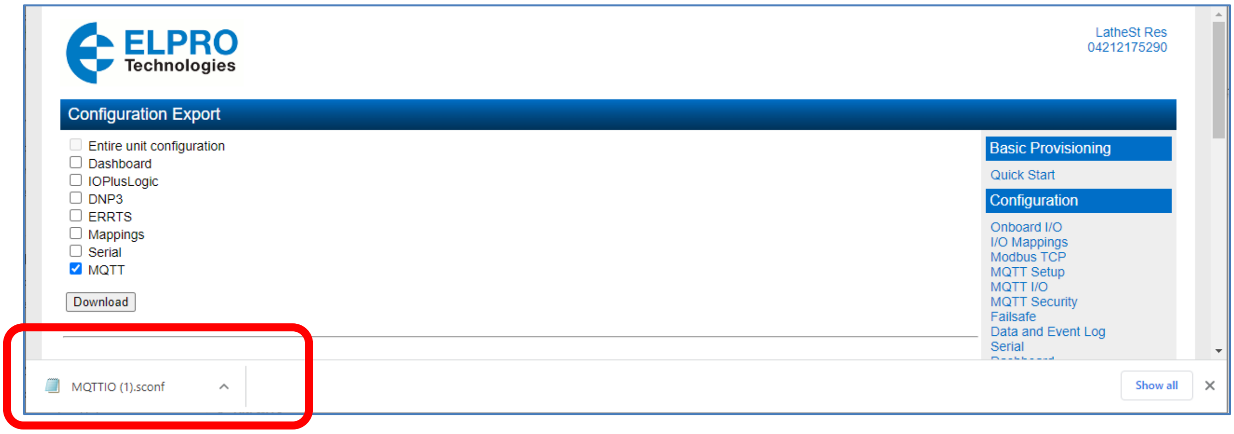
Register		
30430		Broker #1
30445		Broker #2
30460		Broker #3
30475		Broker #4
Register offset for each broker		
	0	Broker Connection Status: 0-No, 1-Yes
	1	Link Count: number of times link to broker has been made
	2	Current link connection up time (seconds, 32 bit - 2 words)
	4	Number of MQTT packets transmitted (32 bit - 2 words)
	6	Number of MQTT packets received (32 bit - 2 words)
	8	Number of MQTT packets stored in queue
	9	Number of bytes transmitted (32 bit - 2 words)

Saving and loading the configuration

Configuration of MQTT gateway can be saved and loaded through the System tools, Configuration Export web page.

Full configuration or only MQTT items can be downloaded. This is useful if several sites have a similar configuration and only the MQTT data can be downloaded and then uploaded into several sites as required.

Click Download button and configuration file will download and be available in Download folder.



Configuration Export

- □ Entire unit configuration
- □ Dashboard
- □ IOPlusLogic
- □ DNP3
- □ ERRTS
- □ Mappings
- □ Serial
- ☑ MQTT

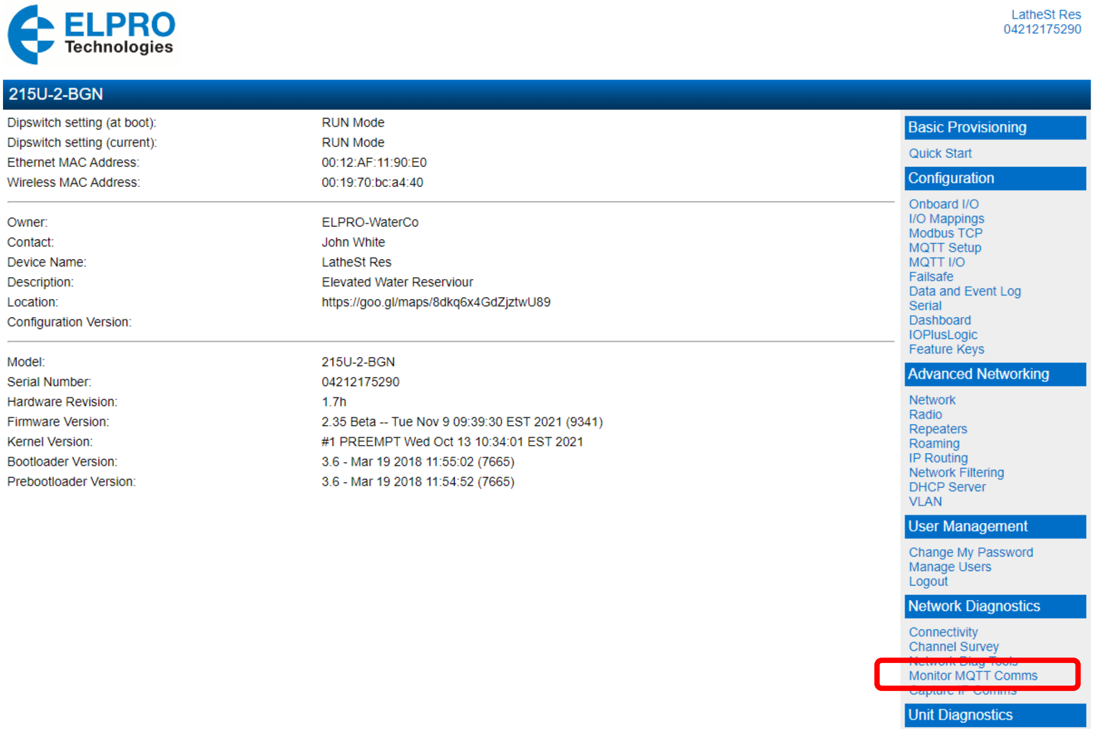
Download

MQTTIO (1).sconf

Warning: If this configuration file is to be used to program into a different unit connected to same broker, then ensure that the ClientID used in the broker configuration in MQTT Setup web page is changed so that it is unique in the system. Failing to do so will lead to the broker denying the connection.

Monitoring MQTT and SparkplugB Communications

The MQTT gateway includes the ability to display the MQTT or SparkplugB messages allowing easy system site commissioning verification and debugging. This feature is access through the web menu in the Networking Diagnostics. See web page screen shot below.



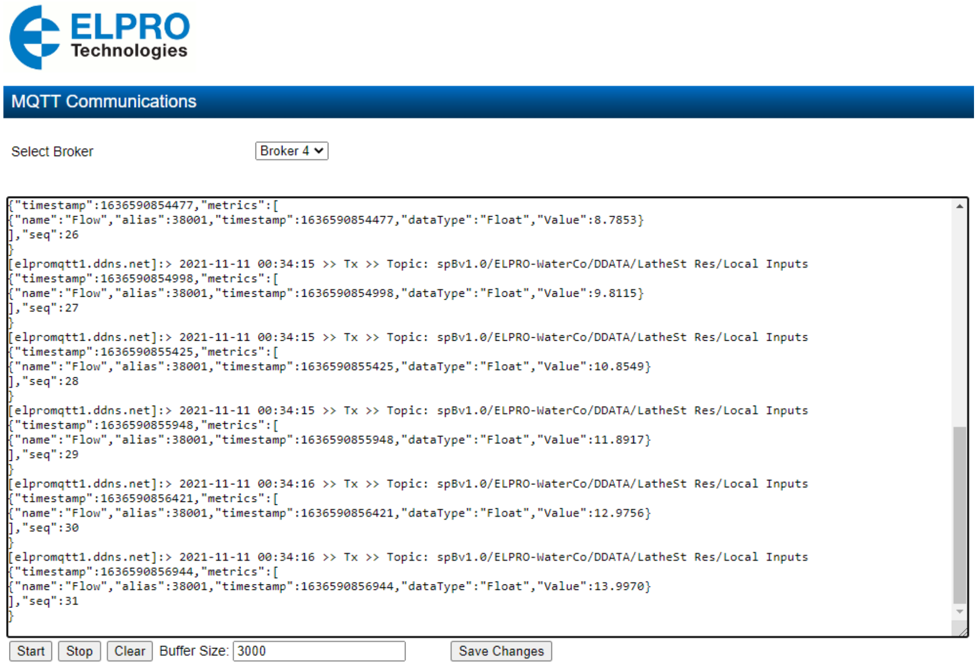
215U-2-BGN

Dipswitch setting (at boot):	RUN Mode
Dipswitch setting (current):	RUN Mode
Ethernet MAC Address:	00:12:AF:11:90:E0
Wireless MAC Address:	00:19:70:bc:a4:40
Owner:	ELPRO-WaterCo
Contact:	John White
Device Name:	LatheSt Res
Description:	Elevated Water Reserviour
Location:	https://goo.gl/maps/8dkq6x4GdZjztwU89
Configuration Version:	
Model:	215U-2-BGN
Serial Number:	04212175290
Hardware Revision:	1.7h
Firmware Version:	2.35 Beta -- Tue Nov 9 09:39:30 EST 2021 (9341)
Kernel Version:	#1 PREEMPT Wed Oct 13 10:34:01 EST 2021
Bootloader Version:	3.6 - Mar 19 2018 11:55:02 (7665)
Prebootloader Version:	3.6 - Mar 19 2018 11:54:52 (7665)

A web page as shown below will be displayed. Select the broker that is required to be display the communications for. Note that the communications shown is only between the wireless unit and the broker, but whole system.

The communications monitoring will automatically Start when the web page is opened and can be stopped by clicking the Stop button or screen can be cleared by clicking the Clear button.

When selecting a different broker, then press save changes to restart communication monitoring.



Select Broker: Broker 4

```
{"timestamp":1636590854477,"metrics":[
{"name":"Flow","alias":38001,"timestamp":1636590854477,"dataType":"Float","Value":8.7853}
],"seq":26
}
[elpromqtt1.ddns.net]:> 2021-11-11 00:34:15 >> Tx >> Topic: spBv1.0/ELPRO-WaterCo/DDATA/LatheSt Res/Local Inputs
{"timestamp":1636590854998,"metrics":[
{"name":"Flow","alias":38001,"timestamp":1636590854998,"dataType":"Float","Value":9.8115}
],"seq":27
}
[elpromqtt1.ddns.net]:> 2021-11-11 00:34:15 >> Tx >> Topic: spBv1.0/ELPRO-WaterCo/DDATA/LatheSt Res/Local Inputs
{"timestamp":1636590855425,"metrics":[
{"name":"Flow","alias":38001,"timestamp":1636590855425,"dataType":"Float","Value":10.8549}
],"seq":28
}
[elpromqtt1.ddns.net]:> 2021-11-11 00:34:15 >> Tx >> Topic: spBv1.0/ELPRO-WaterCo/DDATA/LatheSt Res/Local Inputs
{"timestamp":1636590855948,"metrics":[
{"name":"Flow","alias":38001,"timestamp":1636590855948,"dataType":"Float","Value":11.8917}
],"seq":29
}
[elpromqtt1.ddns.net]:> 2021-11-11 00:34:16 >> Tx >> Topic: spBv1.0/ELPRO-WaterCo/DDATA/LatheSt Res/Local Inputs
{"timestamp":1636590856421,"metrics":[
{"name":"Flow","alias":38001,"timestamp":1636590856421,"dataType":"Float","Value":12.9756}
],"seq":30
}
[elpromqtt1.ddns.net]:> 2021-11-11 00:34:16 >> Tx >> Topic: spBv1.0/ELPRO-WaterCo/DDATA/LatheSt Res/Local Inputs
{"timestamp":1636590856944,"metrics":[
{"name":"Flow","alias":38001,"timestamp":1636590856944,"dataType":"Float","Value":13.9970}
],"seq":31
}
```

Buffer Size: 3000

If the communications messages need to be saved for future reference, then simply select and copy the messages required and paste into Notepad, word or other documenting software as required.

Message are displayed and show these parts:

> Broker Name or IP Address, date/time, Tx (transmitted) or Rx (received), MQTT Topic and MQTT Payload (automatic decoding of sparkplugB)

Note that this diagnostic will only show MQTT or SparkplugB message traffic and does not show raw MQTT protocol such as conneciton

Input Configuration CSV Export-Import

On the MQTT I/O configuration page there is the facility to be able export and import the input publish table as a CSV file. This is a function which is useful when there is a large number of input mappings or for duplicating configurations for multiple sites with minor differences between sites.

Warning:

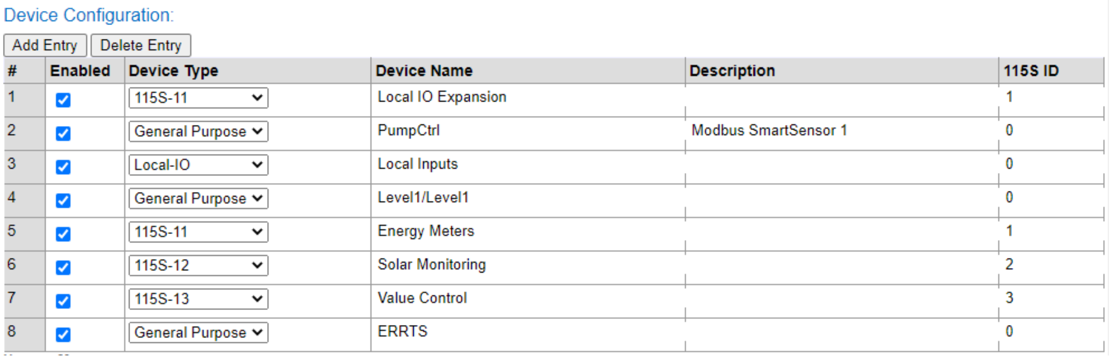
> This function should be used with care and intended for use by expert users and the uploading of incorrect configuration data files can cause unintended results.
>
> When using the upload of CSV input configuration, it is recommended that all input mapping be tested thoroughly as part of the site commissioning process.

A template of the CSV file can easily be obtained by exporting an empty configuration which will provide the table headings to start building the configuration. Below is an example of a input configuration and detailed explanation of each column and data requirements:

Enabled	Device	IO-Type	Local Inpu	Payload Prefix	Register	Input Count	Sensitivity	Update Time	Scaling	Offset
1	0	0	0	HighLevelAlm	10021	1	1	20	1	0
1	1	0	0	Pump1	1	2	1	20	1	0
1	0	0	0	PumpFault	10021	2	1	20	1	0
1	1	0	0	Flow1	36003	2	60	20	1	0
1	1	0	0	Well-Level	38001	1	1	20	1	0
1	4	1	0	Phase kWh 1	30037	3	10	600	1	0
1	5	1	0	Buss bar temp	30041	1	1	600	1	0
1	6	0	0	DI1	10061	1	1	600	1	0
1	7	0	0	RAIN	30551	1	1	600	1	0
1	7	0	0	RIVER	30552	1	1	600	1	0
1	7	0	0	BAT	30553	1	1	600	1	0

Enabled: Either 0 or 1 to disable or enable this inputs mapping line.

Device: This value is a reference to the Device table configured in the MQTT Setup page. The value used here is the line number minus 1. For example for line 2 of the configuration above references the device type PumpCtrl in device configuration shown below. This is line 2, which would require a reference of 1 to be used in input table.



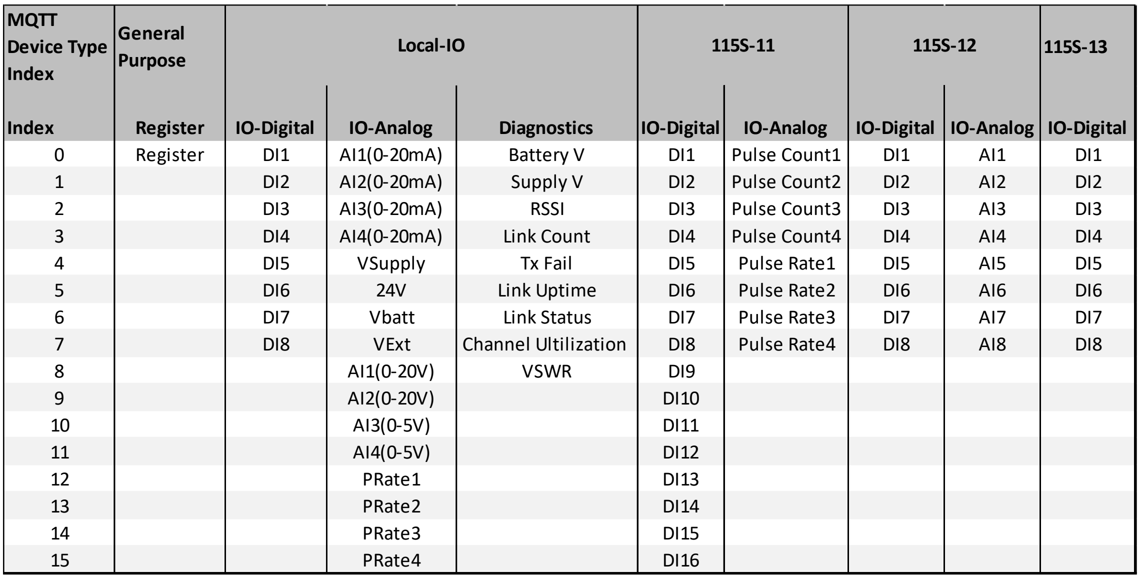
Device Configuration:

Add Entry Delete Entry

#	Enabled	Device Type	Device Name	Description	115S ID
1	☑	115S-11	Local IO Expansion		1
2	☑	General Purpose	PumpCtrl	Modbus SmartSensor 1	0
3	☑	Local-IO	Local Inputs		0
4	☑	General Purpose	Level1/Level1		0
5	☑	115S-11	Energy Meters		1
6	☑	115S-12	Solar Monitoring		2
7	☑	115S-13	Value Control		3
8	☑	General Purpose	ERRTS		0

IO-Type: This field is dependent on the device Type used.

- Local IO and 115S use 0 for IO-Digital and 1 for IO-Analog
- General Purpose use 0 for Register



MQTT Device Type Index	General Purpose	Local-IO			115S-11		115S-12		115S-13
Index	Register	IO-Digital	IO-Analog	Diagnostics	IO-Digital	IO-Analog	IO-Digital	IO-Analog	IO-Digital
0	Register	DI1	AI1(0-20mA)	Battery V	DI1	Pulse Count1	DI1	AI1	DI1
1		DI2	AI2(0-20mA)	Supply V	DI2	Pulse Count2	DI2	AI2	DI2
2		DI3	AI3(0-20mA)	RSSI	DI3	Pulse Count3	DI3	AI3	DI3
3		DI4	AI4(0-20mA)	Link Count	DI4	Pulse Count4	DI4	AI4	DI4
4		DI5	VSupply	Tx Fail	DI5	Pulse Rate1	DI5	AI5	DI5
5		DI6	24V	Link Uptime	DI6	Pulse Rate2	DI6	AI6	DI6
6		DI7	Vbatt	Link Status	DI7	Pulse Rate3	DI7	AI7	DI7
7		DI8	VExt	Channel Ultilization	DI8	Pulse Rate4	DI8	AI8	DI8
8			AI1(0-20V)	VSWR	DI9				
9			AI2(0-20V)		DI10				
10			AI3(0-5V)		DI11				
11			AI4(0-5V)		DI12				
12			PRate1		DI13				
13			PRate2		DI14				
14			PRate3		DI15				
15			PRate4		DI16				

Local Input: This reference is dependent on input type and is the index number starting from zero for the pull-down list that appears in the menu configuration page.

Payload Prefix: Text field that is used for payload data entry. Do not use # or + as they are illegal characters.

Register: Unit register number reference for input data. NOTE: the register value is used as the Sparkplug alias and must be unique.

Input Count: Number of input registers used allowing multiple outgoing MQTT messages to be generated.

Sensitivity: The sensitivity is used for register or analog input values as the threshold required for this input to change before a message is transmitted. The sensitivity should be used carefully to manage the message traffic on the wireless network. If the sensitivity is set to zero, then changes will not be transmitted as value changes. The Update time can be used to send regular updates of the value. If a scale value is used then the sensitivity is applied after scaling.

Update Time (sec): Update time is used to allow a regular update messages to transmitted for the configured input value. If the update time is configured to zero then there will be no updates transmitted. The update time is particularly useful for analog or register values where these is a sensitivity configured to send a current value can capture any small changes inside the sensitivity threshold.

Scaling: For a scaling value for anything other than 1.0 the MQTT gateway will convert the value type to the floating point and apply the scaling value.

Offset: This value is used to apply a fixed value offset to the input or register. If the Offset value is an integer, then the value type is not changed. If the Offset does use decimal places, then the type is converted to a float automatically.

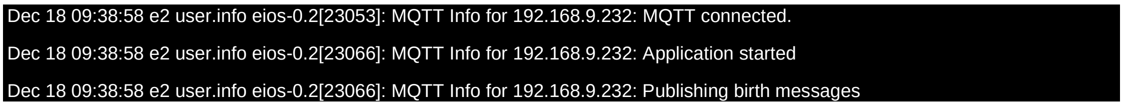
Statistics

The unit Statics web page can provide additional useful information to help diagnose connectivity issues with broker/server or MQTT issues.

Navigate to Unit Diagnostics/Statistics to view web page which includes interface details, System log file, Routes, IP Statistics, TCP/UDP Statistics and others.

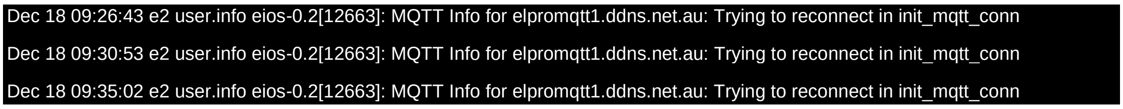
For MQTT gateway the most useful are the system log file which will give a view of broker connectivity attempts, MQTT data parsing problem and other errors that might occur.

For example a failure to connect to broker the system log file will have multiple attempts to connect:



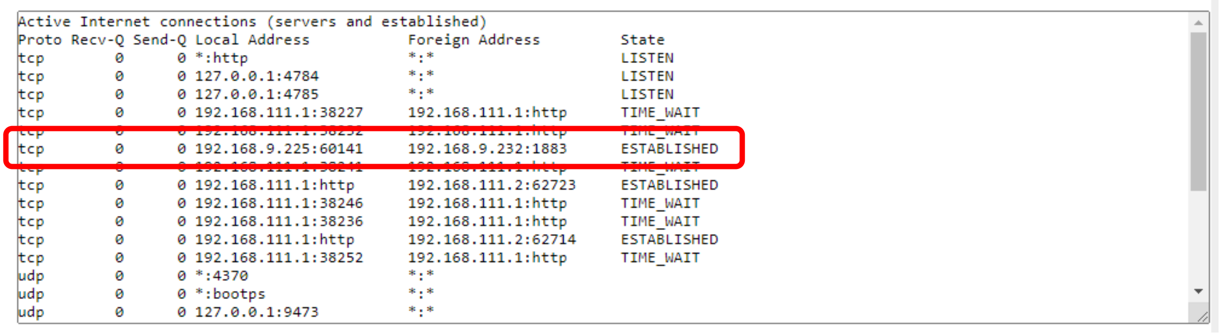
```
Dec 18 09:38:58 e2 user.info eios-0.2[23053]: MQTT Info for 192.168.9.232: MQTT connected.
Dec 18 09:38:58 e2 user.info eios-0.2[23066]: MQTT Info for 192.168.9.232: Application started
Dec 18 09:38:58 e2 user.info eios-0.2[23066]: MQTT Info for 192.168.9.232: Publishing birth messages
```

Ar



```
Dec 18 09:26:43 e2 user.info eios-0.2[12663]: MQTT Info for elpromqtt1.ddns.net.au: Trying to reconnect in init_mqtt_conn
Dec 18 09:30:53 e2 user.info eios-0.2[12663]: MQTT Info for elpromqtt1.ddns.net.au: Trying to reconnect in init_mqtt_conn
Dec 18 09:35:02 e2 user.info eios-0.2[12663]: MQTT Info for elpromqtt1.ddns.net.au: Trying to reconnect in init_mqtt_conn
```

A good broker connection will also be seen in the TCP/UDP Statistics:



```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 *:http                  *:*                     LISTEN
tcp        0      0 127.0.0.1:4784          *:*                     LISTEN
tcp        0      0 127.0.0.1:4785          *:*                     LISTEN
tcp        0      0 192.168.111.1:38227     192.168.111.1:http      TIME_WAIT
tcp        0      0 192.168.111.1:38232     192.168.111.1:http      TIME_WAIT
tcp        0      0 192.168.9.225:60141     192.168.9.232:1883      ESTABLISHED
tcp        0      0 192.168.111.1:38241     192.168.111.1:http      TIME_WAIT
tcp        0      0 192.168.111.1:http      192.168.111.2:62723     ESTABLISHED
tcp        0      0 192.168.111.1:38246     192.168.111.1:http      TIME_WAIT
tcp        0      0 192.168.111.1:38236     192.168.111.1:http      TIME_WAIT
tcp        0      0 192.168.111.1:http      192.168.111.2:62714     ESTABLISHED
tcp        0      0 192.168.111.1:38252     192.168.111.1:http      TIME_WAIT
udp        0      0 *:4370                  *:*
udp        0      0 *:bootps                *:*
udp        0      0 127.0.0.1:9473          *:*
```

Reference Section

Diagnostic Registers full listing for MQTT Gateway

Register		
30430	Broker 1	Broker Connection Status: 0-No, 1-Yes
30431		Link Count: number of times link to broker has been made
30432		Current link connection up time (seconds, 32 bit - 2 words)
30434		Number of MQTT packets transmitted (32 bit - 2 words)
30436		Number of MQTT packets received (32 bit - 2 words)
30438		Number of MQTT packets stored in queue
30439		Number of bytes transmitted (32 bit - 2 words)
30441		Number of bytes received (32 bit - 2 words)
30443-30444		Reserved for future use
30445	Broker 2	Broker Connection Status: 0-No, 1-Yes
30446		Link Count: number of times link to broker has been made

30447		Current link connection up time (seconds, 32 bit - 2 words)
30449		Number of MQTT packets transmitted (32 bit - 2 words)
30451		Number of MQTT packets received (32 bit - 2 words)
30453		Number of MQTT packets stored in queue
30454		Number of bytes transmitted (32 bit - 2 words)
30456		Number of bytes received (32 bit - 2 words)
30458-30459		Reserved for future use
30460	Broker 3	Broker Connection Status: 0-No, 1-Yes
30461		Link Count: number of times link to broker has been made
30462		Current link connection up time (seconds, 32 bit - 2 words)
30464		Number of MQTT packets transmitted (32 bit - 2 words)
30466		Number of MQTT packets received (32 bit - 2 words)
30468		Number of MQTT packets stored in queue
30469		Number of bytes transmitted (32 bit - 2 words)
30471		Number of bytes received (32 bit - 2 words)
30473-30474		Reserved for future use
30475	Broker 4	Broker Connection Status: 0-No, 1-Yes
30476		Link Count: number of times link to broker has been made
30477		Current link connection up time (seconds, 32 bit - 2 words)
30479		Number of MQTT packets transmitted (32 bit - 2 words)
30481		Number of MQTT packets received (32 bit - 2 words)
30483		Number of MQTT packets stored in queue
30484		Number of bytes transmitted (32 bit - 2 words)
30486		Number of bytes received (32 bit - 2 words)
30488-30489		Reserved for future use

Reported Node Metrics

The list of metrics below is published by the module on Birth and as part of regular updates, providing unit information and diagnostics.

{"metrics":[

"System Info/Serial Number",

"System Info/Owner",

"System Info/Contact",

"System Info/Device Name",

"System Info/Description",

"System Info/Location",

Configuration from
Module Information

"System Info/Host Firmware Version",

"System Info/Radio Firmware Version",

"System Info/MAC Address",

"System Info/IP Address",

Unit Info and
configured items

"Diagnostics/Battery Voltage",

"Diagnostics/Supply Voltage",

"Diagnostics/RSSI",

"Diagnostics/Link Count",

"Diagnostics/Tx % Failed Count",

"Diagnostics/Unit Uptime",

"Diagnostics/Module Uptime",

"Diagnostics/Channel utilization",

Unit and Wireless Link
Health and statistics

"Diagnostics/Antenna VSWR fault",

"Broker Diagnostics/IP Address",

"Broker Diagnostics/Connected",

"Broker Diagnostics/Connection Count",

"Broker Diagnostics/Uptime",

"Broker Diagnostics/Messages Tx",

"Broker Diagnostics/Messages Rx",

"Broker Diagnostics/Messages Queue"]}

MQTT Broker/server
health

Birth Messages

Sparkplug has an extension to standard MQTT to provide state information on the connected unit.

When a unit is restarted, after loss of connection or when changes are made to the configuration, Birth messages are sent which provide configuration information for each of the configured published and subscribed data. Birth messages are sent from the Node (wireless unit) and then for each configured Device that is connected to the Node.

Examples of Node and Device Birth Messages below:

Logged Data "Historian" Flagged in Broker configuration

When using sparkplug, during a loss of communications the MQTT messages will include a flag "isHistorical" set to true, to indicate that data is stale. See example below:

{"timestamp":947361963544,"metrics":[

{"name":"Level","alias":38001,

"timestamp":947361963544,"dataType":"Float",**"isHistorical":true**,"value":8.325978},

{"name":"Level","alias":38001,

"timestamp":947361963486,"dataType":"Float",**"isHistorical":true**,"value":8.57652}],"seq":4}

Analog Input with High level Setpoint Alarm

With this configuration its very easy to setup a high level alarm, run a pump or similar applications. In this case tank high level alarm.

Alarm setpoint is configured to trigger ON (true) at 18.0mA and clear OFF (false) at 17.0mA.

See SparkplugB messages below:

Topic: spBv1.0/ELPRO-WaterCo/DDATA/LatheSt Res/Tank Level

Payloads:

```
{"timestamp":947366507207,"metrics":[{"name":"Level","alias":38001,
        "timestamp":947366507207,"dataType":"Float","value":17.29269}],"seq":30}
{"timestamp":947366507873,"metrics":[{"name":"Level","alias":38001,
        "timestamp":947366507873,"dataType":"Float","value":17.830229}],"seq":31}
{"timestamp":947366508131,"metrics":[{"name":"Level High","alias":10009,
        "timestamp":947366508131,"dataType":"Boolean","value":true}],"seq":32}
{"timestamp":947366508541,"metrics":[{"name":"Level","alias":38001,
        "timestamp":947366508541,"dataType":"Float","value":18.35204}],"seq":33}
{"timestamp":947366509208,"metrics":[{"name":"Level","alias":38001,
        "timestamp":947366509208,"dataType":"Float","value":18.860807}],"seq":34}
{"timestamp":947366509831,"metrics":[{"name":"Level","alias":38001,
        "timestamp":947366509831,"dataType":"Float","value":19.363665}],"seq":35}
{"timestamp":947366511877,"metrics":[{"name":"Level","alias":38001,
        "timestamp":947366511878,"dataType":"Float","value":18.836416}],"seq":36}
{"timestamp":947366512491,"metrics":[{"name":"Level","alias":38001,
        "timestamp":947366512491,"dataType":"Float","value":18.343716}],"seq":37}
{"timestamp":947366513156,"metrics":[{"name":"Level","alias":38001,
        "timestamp":947366513156,"dataType":"Float","value":17.834644}],"seq":38}
{"timestamp":947366513773,"metrics":[{"name":"Level","alias":38001,
        "timestamp":947366513773,"dataType":"Float","value":17.299173}],"seq":39}
{"timestamp":947366514183,"metrics":[{"name":"Level High","alias":10009,
        "timestamp":947366514183,"dataType":"Boolean","value":false}],"seq":40}
{"timestamp":947366514440,"metrics":[{"name":"Level","alias":38001,
        "timestamp":947366514440,"dataType":"Float","value":16.792973}],"seq":41}
```

CSV Table Examples

File template column names:

Enabled	Device	IO-Type	Local Input	Payload Prefix	Register	Input Count	Sensitivity	Update Time (sec)	Scaling	Offset
1	0	0	0	HighLevelAlm	10021	1	1	20	1	0
1	1	0	0	Pump1	1	2	1	20	1	0
1	0	0	0	PumpFault	10021	2	1	20	1	0
1	1	0	0	Flow1	36003	2	60	20	1	0
1	1	0	0	Well-Level	38001	1	1	20	1	0
1	4	1	0	Phase kWh 1	30037	3	10	600	1	0
1	5	1	0	Buss bar temp	30041	1	1	600	1	0
1	6	0	0	DI1	10061	1	1	600	1	0
1	7	0	0	RAIN	30551	1	1	600	1	0
1	7	0	0	RIVER	30552	1	1	600	1	0
1	7	0	0	BAT	30553	1	1	600	1	0

Device Type to IO-Type Index Lookup:

MQTT Device Type Index	General Purpose	Local-IO			115S-11		115S-12		115S-13
Index	Register	IO-Digital	IO-Analog	Diagnostics	IO-Digital	IO-Analog	IO-Digital	IO-Analog	IO-Digital
0	Register	DI1	AI1(0-20mA)	Battery V	DI1	Pulse Count1	DI1	AI1	DI1
1		DI2	AI2(0-20mA)	Supply V	DI2	Pulse Count2	DI2	AI2	DI2
2		DI3	AI3(0-20mA)	RSSI	DI3	Pulse Count3	DI3	AI3	DI3
3		DI4	AI4(0-20mA)	Link Count	DI4	Pulse Count4	DI4	AI4	DI4
4		DI5	VSupply	Tx Fail	DI5	Pulse Rate1	DI5	AI5	DI5
5		DI6	24V	Link Uptime	DI6	Pulse Rate2	DI6	AI6	DI6
6		DI7	Vbatt	Link Status	DI7	Pulse Rate3	DI7	AI7	DI7
7		DI8	VExt	Channel Ultilization	DI8	Pulse Rate4	DI8	AI8	DI8
8			AI1(0-20V)	VSWR	DI9				
9			AI2(0-20V)		DI10				
10			AI3(0-5V)		DI11				
11			AI4(0-5V)		DI12				
12			PRate1		DI13				
13			PRate2		DI14				
14			PRate3		DI15				
15			PRate4		DI16				

For more information, please contact your local ELPRO reseller or ELPRO sales, sales@elpro.com.au. Technical support can be obtained by contacting support@elpro.com.au or visit the ELPRO web site www.elpro.com.au

References:

- ELPRO 215U-2/415U/915U-2 User Manual: https://elpro.com.au/resources/user-manuals/
- MQTT Standard Specification: https://mqtt.org/mqtt-specification/
- Sparkplug B Specification: https://projects.eclipse.org/projects/iot.tahu
- MQTT and Sparkplug B videos and information on Automation and SCADA: https://inductiveautomation.com/resources/video