MQTT – Driver Details

V1.0 – November 20th, 2021

CIQProc v5.0.1.2 and later now supports the MQTT protocol. The following describes how to implement the driver. Note if the current CIQProc version is less than 5.0.1.2 you must get a new CIQProc twice in order to get the necessary files. M2MQTT.Net.DLL will be placed in the CIQProc folder.

Testing of this driver has only been implemented on the Raspberry Pi hub with the Zigbee2MQTT program stack. This can be managed on the Raspberry Pi by browsing to its address at port 8080 1http://123.123.123.123:8080 as an example. This document assumes the Raspberry Pi is in place and configured and has all the Devices joined, defined and tested.

Definitions

Broker – The connection point for the driver. This is normally an IP and Port defined in Subsystems table of the ControlIQ database. The default port is 1883. In most cases this will be the same as the Hub, but there is the case where a Broker may be on a public server communicating to a local broker. Note this can be a Named (DNS) broker as well. This should look like 123.123.123.123:1883 or Broker.ControlIQ.Com:1883

Hub – This is the 'Name' of the Zigbee2MQTT in the Raspberry Pi. This is officially called a "Base Topic" (sometimes referred to as a 'Coordinator') but for clarity (and to be synonymous with Home Automation systems) we will refer to it as a 'Hub'. This should be unique at all sites to take advantage of public Brokers. We recommend this be set to the PropertyID defined in the ControlIQ Database (like S378). Keeping this short is a benefit if CIQ is going to have multiple Brokers on the same instance.

Device – The name of a device attached to the Hub. This is the "Friendly Name" assigned in the web interface. This is typically Relay_1, Relay_2, P1A,, etc. It can also be a "Group" defined in the web interface.

Verb – This is the data instance or 'property' that you want to control or monitor. This must be known from the parameters that the device "Exposes" (shown in the web interface).

Groups – Groups can be created to control a number of Devices. Groups can be commanded but cannot be read. For example, you can command a group of lights to be On or Off, but you cannot read back the state of a group.

Note: MQTT Protocol is case-sensitive. You must match the type-case exactly that exists for the Hubs, Devices, Verbs and Groups. Relay_1, relay_1, and RELAY_1 would define 3 different devices.

Firewall considerations:

If the Raspberry Pi is not on the same network as the ControlIQ server, the customer's firewall will have to pass through port 1883 (at a minimum) and port 8080 if the Hub is to be maintained remotely.

In the Raspberry Pi web interface (Hub) you 'Join' and define each Zigbee Device.

For devices with Batteries, you will need to enable the "Retain" feature since these devices cannot be polled directly. They send a message to the Hub when they change state, and the Hub 'retains' the value to be read by ControlIQ. Go to [Web Interface] / Devices / [Friendly name] / Settings. Check the "Retain" option and Click on "Submit"

ControlIQ Database

Subsystems

In the subsystems table, define a new subsystem with the Subsystem Type=MQTT. For this example, we will assign a subsystem Alias = "MQTTS378". In the Settings field, put the IPaddress:port of the Broker.

Personality Field

Hub=S378 – this is optional but saves typing in the points table
QOS=2 – 0,1,2 are the options. 2 is the default.
RefreshInterval=15 [seconds] – time between polls for new data
QuietTime=50 [milliseconds] – delay between its transmission

Points

In the points table, define new points as usual. The driver supports all point types.
Subsystem - enter the alias from the subsystems tables (MQTTS378).
Address – enter the details in the format as Hub.Device.Verb If you omit the Hub, it will use the one defined in the Subsystem.Settings field.
Some Examples of the address field

Type	Address	
DI	S378.P1A.state	(for this device returns On or Off (cannot be set)
AI	S378.P1A.brightness	(for this device returns 0-254 represents (0-100%)
AO	S378.P1A.brightness	(sets the value in the device (see conversion below))
AI	S378.P1A.last_seen	(returns the last date / time the device we seen)
AI	S378.P1A.linkquality	(return the signal strength of the comm to this device)
DO	S378.P1A.brightness	(set the device level (see conversion below))

DO S378.Relay_1.state – Sets the Relay (On/Off)
DI S378.Realy_1.state – Reads the Relay (On/Off)

DI S378.MotionDetector.occupancy (True/False)

AI S378.MotionDetector.temperature
DI S378.MotionDetector.batter_low (True/False)
AI S378.MotionDetector.battery (%)

Remember, all of these addresses can be entered without the “S378.” prefix (Hub) if it is entered in the Subsystem.Personality field, however, the Hub MUST BE included in the address field if there are multiple Hubs on the Broker.

The values read/written will be multiplied by the value in the Points.Conversion field. If there is nothing in the Conversion field a ‘1’ is assumed.

For example:

For a DO / Sched / Logic (digital) point type:
If the conversion field is ‘254’
When the point is Off it will send brightness=0
When the point is On it will send brightness=254 (1 x 254)

For an AO / MVO / Logic (analog) point type
If the conversion field is ‘2.54’
When the value of the point is 0, brightness=0 (Light is Off)
When the value of the point is 100, brightness=254 (100 x 2.54) (Light is On)

For and AI / MVI point type
If the conversion field =’.001’
When the ‘voltage’ is read as 3000, it will return 3.0 (volts) to proc.

If the conversion field =’.3937’
When the ‘brightness’ is read as 254, it will return 100 (%) to proc.

Other Conversions:
If the ‘value’ of the Verb is “On” or “True” the value returned is ‘1’
If the ‘value’ of the Verb is “Off” or “False” the value returned is ‘0’

Many times, “temperature” returns a Celsius reading. In the conversion field put ‘C2F’ to convert the reading to Fahrenheit.

If the Verb is “last_seen”
This will show the elapsed minutes since last communication between the Device and the Hub. (This may be useful for alarming if the duration is too long)

CIQProc

When CIQProc starts, it will create a file in the CIQProc directory that contains the name of the Hub such as: MQTT-Hub-S378.txt. This file contains all of the properties of the Hub. It is useful that you can find all of the devices (Friendly-Names) that are known to the Hub, including their typing case.

As CIQProc is running you can turn on the Driver diagnostics and observe the data flow. This will indicate all of the Verbs that are exposed for each Device (if they are unknown). Note that in some devices (Like P1A) the ‘state’ can be read, but not set.