

Integrating ORB-X1 With SCADA Server Via MQTT

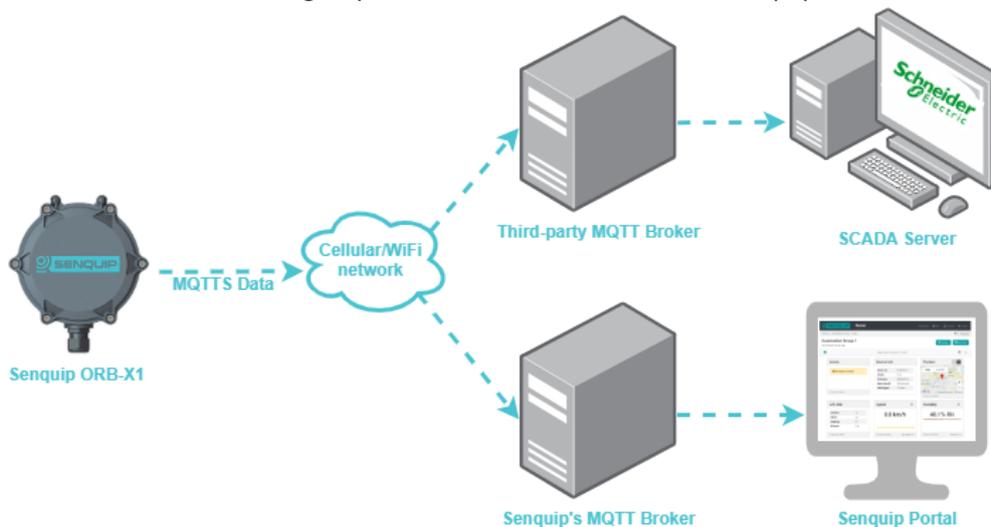
Introduction

The ORB-X1 supports several commonly used communication protocols to interface with field devices and communicate with a top-end server. This application note details how to integrate the ORB-X1 with a SCADA server using secured MQTT communications. MQTT stands for MQ Telemetry Transport and is a lightweight, publish-subscribe network protocol that transports messages between devices. The MQTT data is secured using client and server TLSv1.2 certificates. In the MQTT protocol devices publish data under a 'topic', which other devices can then subscribe to. The data is published to a common central server called a 'Broker'. The Broker receives data and will send it to all devices 'subscribed' to that topic.

This document will outline how to configure an MQTT broker secured using TLSv1.2, how to generate TLS certificates for devices, how to provide those certificates to the ORB-X1 and GeoSCADA, and how to configure GeoSCADA as an MQTT subscriber to access the ORB's data.

Upon completion of this document you will have:

- An MQTT broker that can communicate securely with any device that supports MQTT
- A SCADA server communicating with the broker to access device data
- An ORB-X1 communicating to your secure MQTT broker and Senquip's secure broker



Requirements

The following are required for integration of the ORB-X1 into an existing SCADA system:

- SCADA server that supports MQTTS using TLSv1.2 (we will use GeoSCADA)
- Access to an ORB-X1

Document Number **Revision**
 APN0007 1.0

Prepared By
 HDW

Approved By
 NGB

Title
 Integrating ORB-X1 With SCADA Server Via MQTT

Page
 2 of 8

- Server with network connection to SCADA server and ORB-X1 (can be the same server as SCADA server) to use as MQTT broker

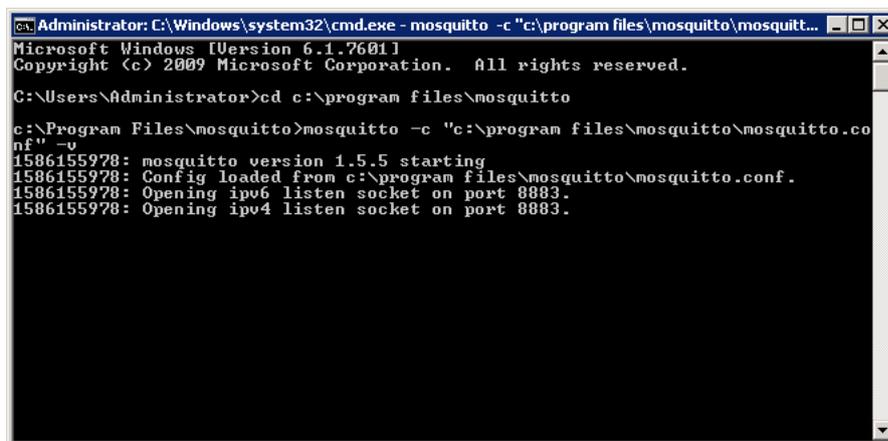
MQTT Broker Configuration

This section outlines how to set up a secure MQTT broker using the open source Mosquitto MQTT broker and securing the broker using TLSv1.2. Any broker that supports TLSv1.2 can be used. If you already have access to an MQTT broker using TLSv1.2 you can skip this section.

1. Download and install the latest version of Mosquitto for your operating system on your server from <https://mosquitto.org/download/>. It is recommended that you install this as a service so that it will start automatically.
2. Follow the steps outlined here <http://www.steves-internet-guide.com/mosquitto-tls/> to use OpenSSL to create a Certificate Authority (CA), generate server keys and certificates, and configure MQTT to require certificates. For ease of configuration, below are the contents of a typical Mosquitto configuration file. By default, this is located at "C:\Program Files\mosquitto\mosquitto.conf".

```
port 8883
cafile C:\Program Files\mosquitto\certs\ca.crt
keyfile C:\Program Files\mosquitto\certs\server.key
certfile C:\Program Files\mosquitto\certs\server.crt
tls_version tlsv1.2
require_certificate true
use_identity_as_username true
```

3. Test your broker by starting it. To do this, open Command Prompt and navigate to mosquitto's install location using the 'cd' command. E.g. "cd c:\program files\mosquitto\" if using default installation location. Then enter "mosquitto -c "<configuration file path>" -v". This starts mosquitto using the specified configuration file in 'Verbose Mode', which will display some diagnostic information about the broker in the CMD window:



```
Administrator: C:\Windows\system32\cmd.exe - mosquitto -c "c:\program files\mosquitto\mosquitto.conf" -v
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd c:\program files\mosquitto
c:\Program Files\mosquitto>mosquitto -c "c:\program files\mosquitto\mosquitto.conf" -v
1586155978: mosquitto version 1.5.5 starting
1586155978: Config loaded from c:\program files\mosquitto\mosquitto.conf.
1586155978: Opening ipv6 listen socket on port 8883.
1586155978: Opening ipv4 listen socket on port 8883.
```

Document Number **Revision**
 APN0007 1.0

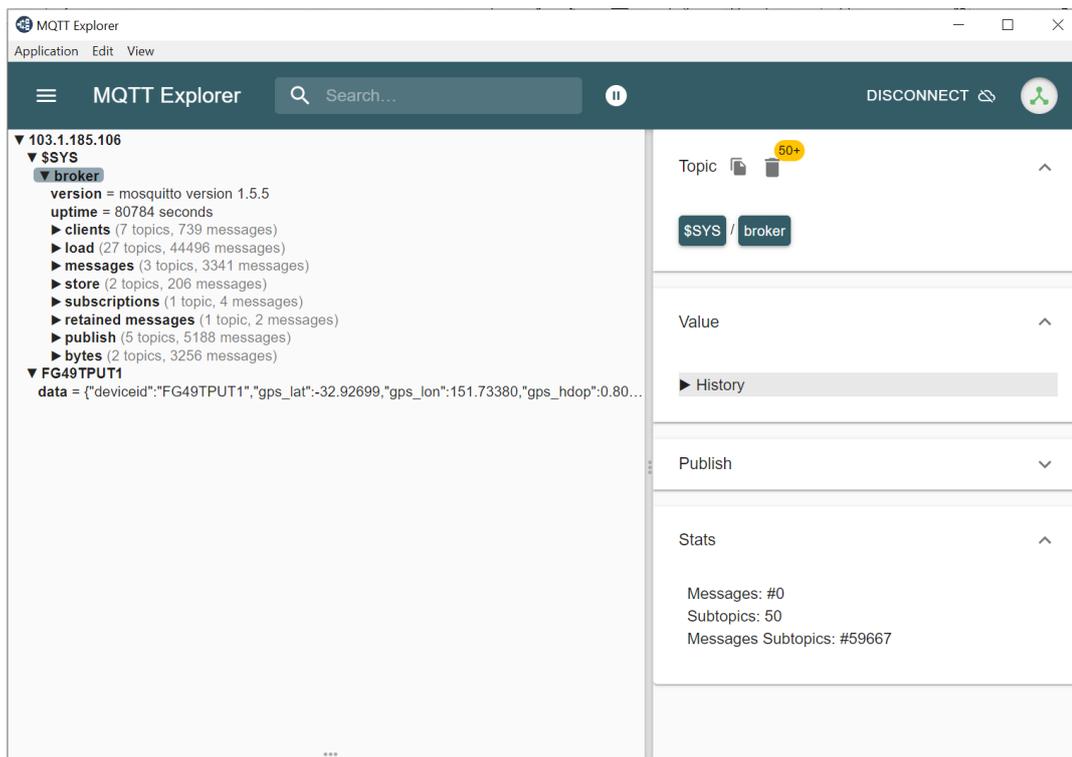
Prepared By
 HDW

Approved By
 NGB

Title
 Integrating ORB-X1 With SCADA Server Via MQTT

Page
 3 of 8

4. Follow the steps here <http://www.steves-internet-guide.com/creating-and-using-client-certificates-with-mqtt-and-mosquitto/> to create client certificates using the Certificate Authority created earlier. A unique client certificate should be issued for each MQTT device that will connect to the broker.
5. It is recommended that you configure the MQTT broker to start on boot.
6. Test the broker and client certificate by installing MQTT-Explorer from here <http://mqtt-explorer.com/>. Enter the host name/IP and port number. Go to 'Advanced' > 'Certificates' and add the certificates generated in (4). The image below shows a successful connection to a Broker. The '\$SYS' topic contains information on the broker, and the 'FG49TPUT1' topic contains data from an ORB-X1 device that is connected to the Broker.



7. If you are unsuccessful in connecting to the broker, observe the Command Prompt Window on the MQTT Broker server for feedback. E.g. if you attempt to connect to the Broker without supplying certificates the following is displayed:

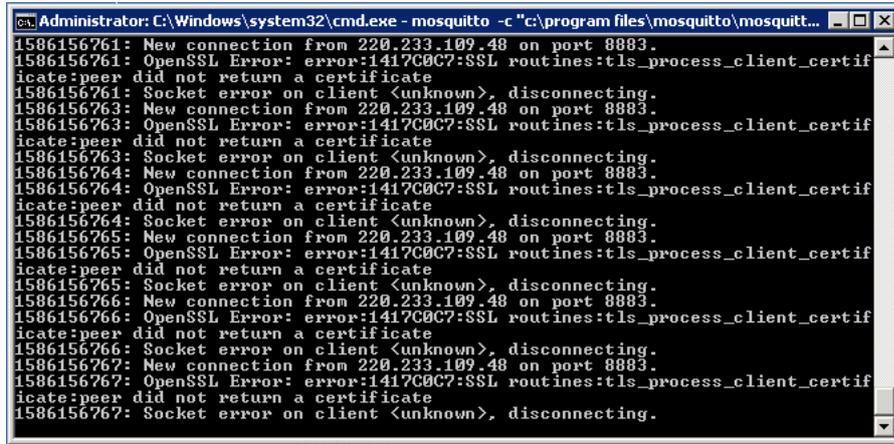
Document Number **Revision**
APN0007 1.0

Prepared By
HDW

Approved By
NGB

Title
Integrating ORB-X1 With SCADA Server Via MQTT

Page
4 of 8



```
Administrator: C:\Windows\system32\cmd.exe - mosquitto -c "c:\program files\mosquitto\mosquitto.conf"
1586156761: New connection from 220.233.109.48 on port 8883.
1586156761: OpenSSL Error: error:1417C0C7:SSL routines:tls_process_client_certificate:peer did not return a certificate
1586156761: Socket error on client <unknown>, disconnecting.
1586156763: New connection from 220.233.109.48 on port 8883.
1586156763: OpenSSL Error: error:1417C0C7:SSL routines:tls_process_client_certificate:peer did not return a certificate
1586156763: Socket error on client <unknown>, disconnecting.
1586156764: New connection from 220.233.109.48 on port 8883.
1586156764: OpenSSL Error: error:1417C0C7:SSL routines:tls_process_client_certificate:peer did not return a certificate
1586156764: Socket error on client <unknown>, disconnecting.
1586156765: New connection from 220.233.109.48 on port 8883.
1586156765: OpenSSL Error: error:1417C0C7:SSL routines:tls_process_client_certificate:peer did not return a certificate
1586156765: Socket error on client <unknown>, disconnecting.
1586156766: New connection from 220.233.109.48 on port 8883.
1586156766: OpenSSL Error: error:1417C0C7:SSL routines:tls_process_client_certificate:peer did not return a certificate
1586156766: Socket error on client <unknown>, disconnecting.
1586156767: New connection from 220.233.109.48 on port 8883.
1586156767: OpenSSL Error: error:1417C0C7:SSL routines:tls_process_client_certificate:peer did not return a certificate
1586156767: Socket error on client <unknown>, disconnecting.
```

Document Number **Revision**
APN0007 1.0

Prepared By
HDW

Approved By
NGB

Title
Integrating ORB-X1 With SCADA Server Via MQTT

Page
5 of 8

ORB-X1 Configuration

To configure an ORB-X1 to use a third-party Broker, complete the following steps.

1. Browse to <http://portal.senquip.com>, log in, and select the ORB you wish to connect to a Broker.
2. Go to Settings > General and confirm that your device is using firmware version 2020022710 or newer.
3. In settings, go to Endpoint > MQTT
4. Enter your Broker's IP address or host name and port under the 'Broker Address' field separated by a colon. E.g. *MyBroker.com:8883*, or *10.1.1.1:8883*.
5. Set the Client ID. This is used by the broker to identify the device and should be unique per client. It is typically set to the ORB's ID.
6. Generate a private key and client certificate for this device (if using OpenTLS and Mosquitto see instructions for how to do this here: <http://www.steves-internet-guide.com/creating-and-using-client-certificates-with-mqtt-and-mosquitto/>)
7. Click 'Configure MQTTS' and add the CA Certificate, Client Private Key, and Client Certificate to the ORB. Click Apply when ready.

Document Number **Revision**
APN0007 1.0

Prepared By
HDW

Approved By
NGB

Title
Integrating ORB-X1 With SCADA Server Via MQTT

Page
6 of 8

Configure MQTTS

Configure the device to connect to a private MQTT broker securely with TLS v1.2. All certificates and keys loaded here are sent to the device using a secure connection and stored in an encrypted section of device memory.

Certificates and keys must be in Base-64 encoded X.509 format. Certificates signed with SHA1 are not supported.

Pressing Apply will clear and update all existing MQTT settings, including previously loaded certificates.

Broker Address

Client ID

Username

Password

CA Certificate ca.crt

Client Certificate client.crt

Client Private Key client.key

8. The next time the ORB connects to the Senquip portal these certificates and settings will be loaded in and the ORB will connect to your Broker.

The ORB-X1 publishes its data to a topic with the ORB's device ID. In this topic is a subtopic called 'data' where the ORB's data is published in the JSON format. A sample message in this format can be seen below. This data was published to *FG49TPUT1/data* on the Broker.

```
{  
  "deviceid": "FG49TPUT1",  
  "humidity": 63.27,  
  "vreg": 0.00,  
  "vsys": 3.31,  
  "vbat": 0.00,  
  "vin": 0.00,  
  "analog1": 0.00,  
  "analog2": 0.00,  
  "roll": 2.5,  
  "pitch": 86.5,  
}
```

Document Number **Revision**
 APN0007 1.0

Prepared By
 HDW

Approved By
 NGB

Title
 Integrating ORB-X1 With SCADA Server Via MQTT

Page
 7 of 8

```

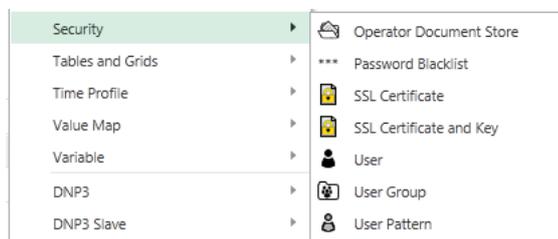
    "angle":87.5,
    "motion":12,
    "pressure":102.01,
    "ambient":24.37,
    "movement_hrs":0.75,
    "ts":1586288847.7,
    "time":1586288848,
    "tsformat":"07/04/2020 19:47:27"
}
    
```

GeoSCADA Configuration

Please note, although some versions of ClearSCADA do support MQTT, the driver for it must be installed separately, has limited functionality, and cannot use the required TLS certificate formats. If you have ongoing ClearSCADA support with Schneider Electric, upgrading to GeoSCADA is free and strongly recommended if using any MQTT devices.

To access the ORB's data as points in GeoSCADA you will need a Broker object, a KeyStore object to hold the Client Key and Client Certificate, a TrustStore object to hold the CA Certificate, an MQTT device, and several MQTT points. The following steps outline how to add and configure these objects.

1. Generate a private key and client certificate for the SCADA server (if using OpenTLS and Mosquitto see instructions for how to do this here: <http://www.steves-internet-guide.com/creating-and-using-client-certificates-with-mqtt-and-mosquitto/>)
2. Open ViewX and add one 'SSL Certificate' object by right clicking on a folder in the database, then *Create New > Security > SSL Certificate*. Name this object 'TrustStore'. Right click on it and select 'Import certificate...' and find the CA Certificate. Give it a description such as 'CA Certificate'.



3. Now add an 'SSL Certificate and Key' object and name it KeyStore. Import the certificate as in (2) and select the Client Certificate and Client Key.
4. Add an MQTT 'Broker' object to your database by right clicking on a folder then *Create New > MQTT > Broker*. Enter the Broker details and enable SSL. Point the Broker to the TrustStore and KeyStore objects created previously. An example configuration can be seen below:

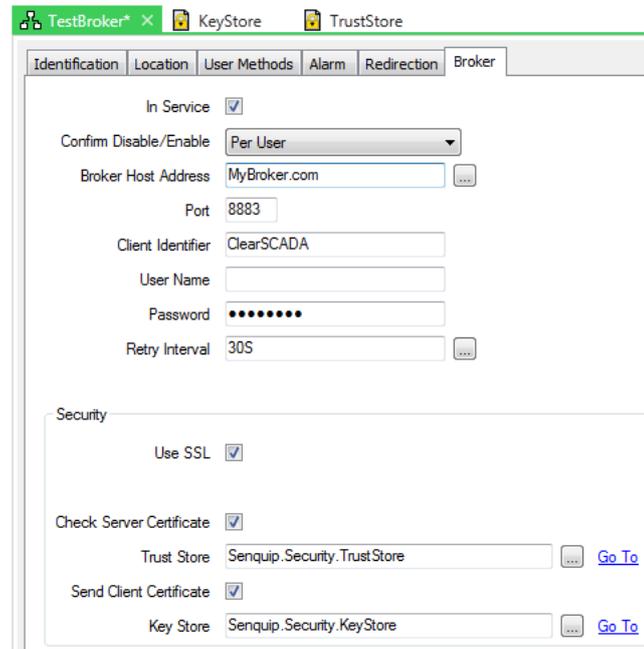
Document Number **Revision**
 APN0007 1.0

Prepared By
 HDW

Approved By
 NGB

Title
 Integrating ORB-X1 With SCADA Server Via MQTT

Page
 8 of 8



5. Add an MQTT device by right clicking on a folder then *Create New > MQTT > JSON Schema A > Device*. Enter the following string in the 'Topic Prefix' field, with your device's ID: *<ORB device ID>/data*
6. Add some MQTT points by right clicking on a folder then *Create New > MQTT > JSON Schema A > Analog/Digital/Counter Point*. For each point set the 'Tag Name' field to the data's label in the JSON data the ORB publishes. I.e. enter 'ambient' to get ambient temperature from the ORB's on board temperature sensor.

Conclusion

Setting up the ORB-X1 to use MQTTS to a third-party broker is quick and easy. If you are using a SCADA server that natively parses JSON data like GeoSCADA it is simple to read, trend, and act upon this data.

If you have any queries about the procedures in this document or would likely to know more about the ORB-X1 please contact Automation Group Support at support@automationgroup.com.au or call 1300 724 743 and select Option 1.