MQTT

Connecting to the Mosquitto MQTT Broker

Vers. 1.0 – Jul 2021

1. Introduction

All of Infinite's devices that support the MQTT protocol, are capable to connect to any local or remote MQTT Broker. Mosquitto is an open source message broker that implements the MQTT protocol. It is lightweight and suitable for use on all devices ranging from low power single board computers to full servers.

This document is a brief how-to guide for all device communications between Infinite's devices and the Mosquitto MQTT Broker.

2. Installing the Mosquitto MQTT Broker to your server

Installation is a straightforward procedure. Simply download the package (x64 or x32) from here and follow the Eclipse Mosquitto Setup.

Once installed, open a Command Prompt or PowerShell window in the installed directory and type "mosquitto -c mosquitto.conf -v". This command starts the Mosquitto Broker with the settings that are configured in the mosquitto.conf file, in verbose mode. The .conf file is where the TLS, listening port, IPv and many other options are set.

2.1 Configuring TLS on Mosquitto

For a more secure connection we offer TLS support which is the standard in MQTT.

On the Mosquitto side we need to create the Broker certificates and keys. We do that with the commercial-grade TLS toolkit openssl. The easiest way to do that is to simply

install git on your computer and locate the openssl.exe file in this directory:

`C:\Program Files\Git\usr\bin\openssl.exe.`

Open a Command Prompt or PowerShell window in the above directory and type the following commands to create the server certificates and keys:

genrsa -des3 -out ca.key 2048 - creates a key pair for the CA

req -new -x509 -days 1826 -key ca.key -out ca.crt - creates a certificate for the CA

gensra -out server.key 2048 - creates a server key pair

x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt -days 36 - creates the server certificate

(These commands are for testing purposes and should be adjusted for different requirements.)

You can now edit the .conf file with the directories of the above files along with your preferred listener port (8883 for TLS). TLS version should be 1.2.

Follow this detailed tutorial on how to create the server certificates and keys as well as edit the .conf file.

2.2 Creating the Device Certificate and Private Key

Using openssl, we create the device (client) certificate and key using one of the files that we created previously. In an openssl window type the following commands to create the client certificate and private key:

gensra -out client.key 2048 - creates a client private key

req -new -out client.csr -key client.key - creates a certificate request

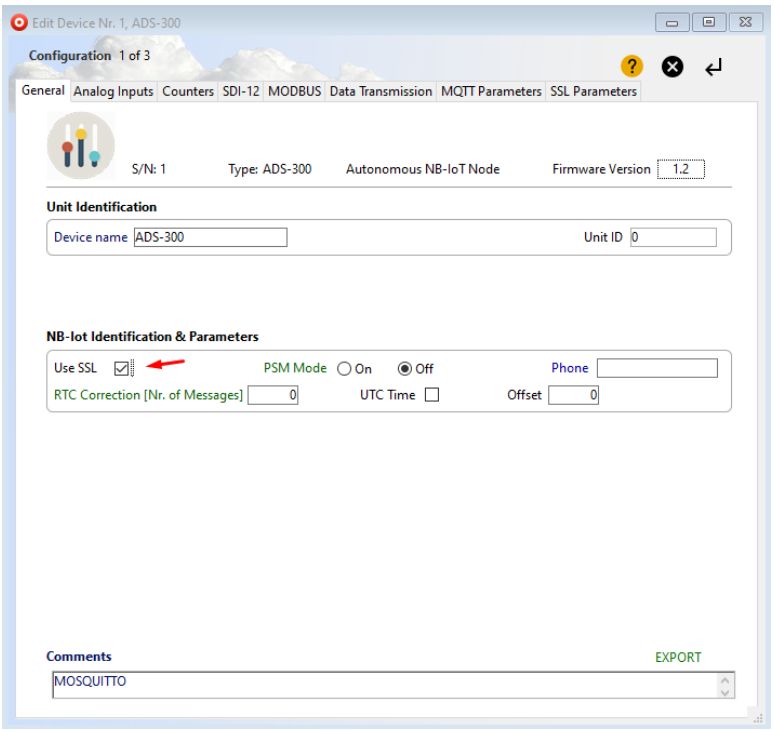
x509 -req -in client.csr -CA ca.crt -Cakey ca.key -CAcreateserial -out client.crt -days 360 - creates client certificate

(These commands are for testing purposes and should be adjusted for different requirements.)

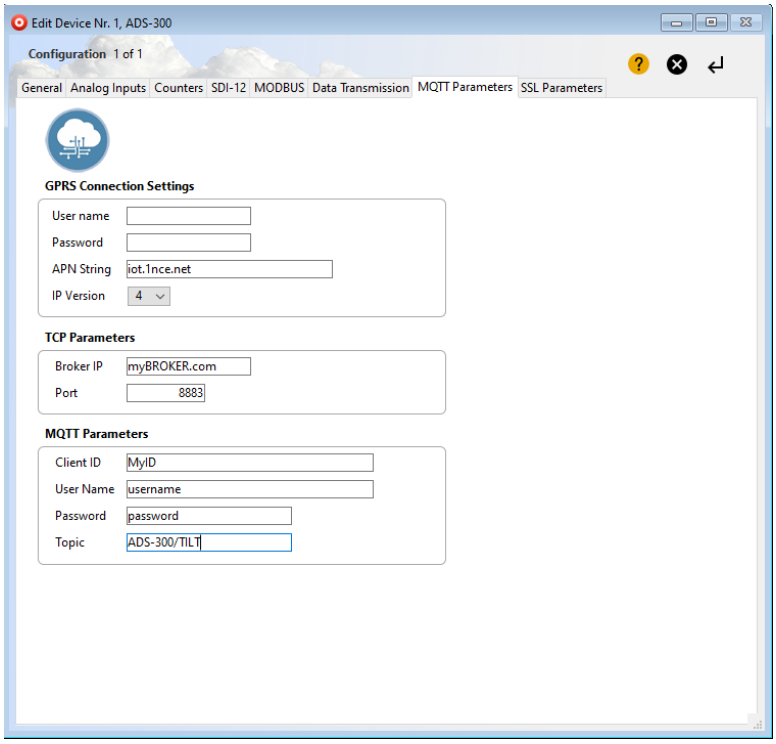
Follow this detailed tutorial on how to create the client certificate and key.

3. Device Configuration with WA Manager

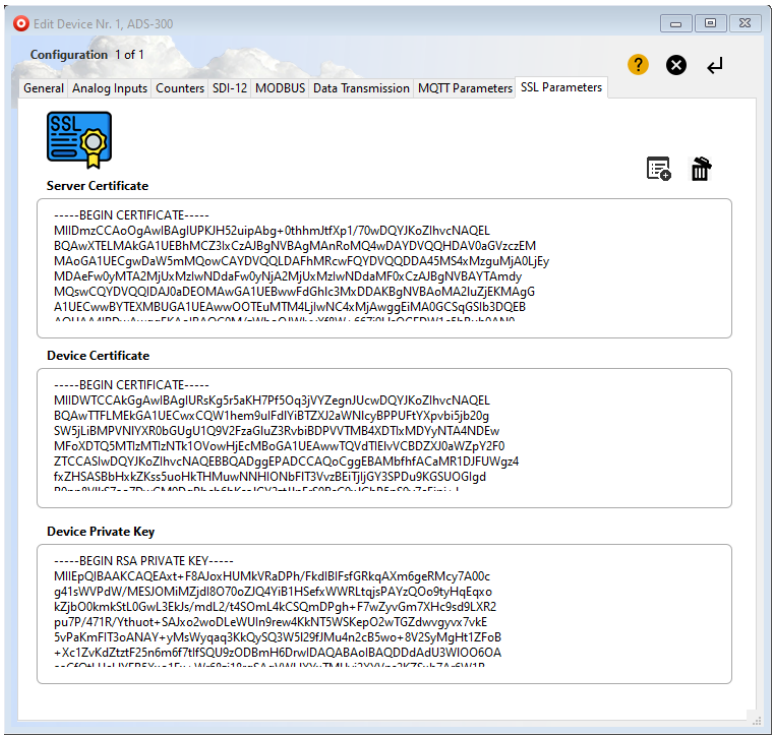
In the Edit Device window in WA Manager, tick the Use SSL box.



Next, we configure the MQTT parameters.



Lastly, in the SSL Parameters tab, we copy and paste the three files needed for the TLS communication: Server Certificate (CA), Device Certificate and Device Private Key.



The Server Certificate is the ca.crt file we created, the Device Certificate is the client.crt file and the Device Private Key is the client,key file. These files should be first opened with Notepad++ and their contents should be copy and pasted in the above tab. All files must be PEM formatted.

Your device can now connect to the Mosquitto Broker and send your encrypted data safely.

Disclaimer:

Mosquitto is an open source (EPL/EDL licensed) message broker that implements the MQTT protocol. All products and software mentioned in this document for educational and demonstration purposes.

Revision: 1.0

Infinite Informatics, Ltd
1, Valaoritou Street
GR-54626 Thessaloniki, Greece
Phone: +30-2310-553545
E: info@indinf.gr
W: www.infinite.com.gr