MQTT + SSL (MQTTS)

Cvision

History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date (dd/mm/yyyy)** | **Author** | **Comment** |
| V1.0 | 02/22/2016 | Phuc Pham | Initial |
|  |  |  |  |
|  |  |  |  |

Executive summary

To define the API for MQTTS library

Reference

Contents

[I. Introduction 3](#_Toc443983109)

[1. MQTT 3](#_Toc443983110)

[2. TLS/SSL 3](#_Toc443983111)

[3. MQTTS 3](#_Toc443983112)

[II. Application programing interface (API) 3](#_Toc443983113)

[1. mqtts\_set\_clientid 3](#_Toc443983114)

[2. mqtts\_set\_callbackhandler 3](#_Toc443983115)

[3. mqtts\_set\_malloc 3](#_Toc443983116)

[4. mqtts\_set\_free 3](#_Toc443983117)

[5. mqtts\_set\_logger 3](#_Toc443983118)

[6. mqtts\_init 4](#_Toc443983119)

[7. mqtts\_subscribe 4](#_Toc443983120)

[8. mqtts\_unsubscribe 4](#_Toc443983121)

[9. mqtts\_control 4](#_Toc443983122)

[10. mqtts\_main 4](#_Toc443983123)

[11. mqtts\_publish 4](#_Toc443983124)

[III. How to use this library 4](#_Toc443983125)

# Introduction

## MQTT

* MQTT is a machine-to-machine (M2M)/"Internet of Things" connectivity protocol. It was designed as an extremely lightweight publish/subscribe messaging transport. Refer <http://mqtt.org>
* The PAHO MQTT project (<http://www.eclipse.org/paho> ) supports 2 licenses: Eclipse Public License v1.0 (<http://www.eclipse.org/legal/epl-v10.html>) and Eclipse Distribution License v1.0 (<http://www.eclipse.org/org/documents/edl-v10.php>)

## TLS/SSL

* Refer: <https://en.wikipedia.org/wiki/Transport_Layer_Security>
* The MBED TLS (<https://tls.mbed.org> ) supports 2 licenses GPL and APACHE.

## MQTTS

* The purpose of this library:
  + Support 2-way communication over network (MQTT).
  + Support security for the communication (TLS/SSL).
* MQTTS is MQTT that is built up on TLS/SSL to create the secure channel for the communication over network

# Application programing interface (API)

All API is declared in mqtts-main/mqtts.h:

## mqtts\_set\_clientid

To set the mqtt client ID for application. The max client ID length is 64 bytes by default that can be changed in mqtts.h : MQTT\_CLIENTID\_LENGTH

## mqtts\_set\_callbackhandler

To set the callback handler to handle the incoming message from the pre-subscribed channel

## mqtts\_set\_malloc

To set the malloc function that is used in library

## mqtts\_set\_free

To set the free function that is used in library

## mqtts\_set\_logger

To set the logger function that is used in library

## mqtts\_init

To init the mqtt connection to the mqtt server:port in parameter

## mqtts\_subscribe

To subscribe the topic. By default it supports to subscribe maximum 8 topic at the same time that can be changed in mqtts.h : MAX\_SUB\_TOPIC\_SUPPORT . The maximum topic length is 64 bytes by default that can be changed in mqtts.h : MQTT\_TOPIC\_LENGTH

## mqtts\_unsubscribe

To unsubscribe the topic

## mqtts\_control

To send the command to control the MQTTS library when running. Currently, it supports 2 commands: MQTTS\_CMD\_START, MQTTS\_CMD\_STOP

## mqtts\_main

The main loop of MQTTS library that will listen and handle all incoming MQTT message

## mqtts\_publish

To publish the message to the topic . The maximum length of data will be published is 256 bytes.

# How to use this library

Fill all needed information before start the library, refer linux-test.c:

*client\_id\_generator(client\_id);*

*device\_topic\_generator(topic\_id);*

*printf("set all MQTTS items\n");*

*mqtts\_set\_clientid(client\_id);*

*mqtts\_set\_callbackhandler(&mqtt\_process\_command);*

*mqtts\_set\_malloc(&malloc);*

*mqtts\_set\_free(&free);*

*mqtts\_set\_logger(&camera\_logger);*

*printf("Start the mqtt thread\n");*

*mqtts\_init(MQTTS\_SERVER, MQTTS\_PORT);*

*mqtts\_subscribe(topic\_id);*

*mqtts\_unsubscribe(topic\_id);*

*mqtts\_subscribe(topic\_id);*

*mqtts\_subscribe("abc123");*

*mqtts\_subscribe("def567");*

*mqtts\_control(MQTTS\_CMD\_START);*

*mqtts\_main();*