

# CC31xx MQTT Client

---

## Overview

MQTT(Message Queue Telemetry Transport) protocol is an extremely light weight machine to machine connectivity protocol. It is based on publish/subscribe messaging model and is designed to be used on the top of TCP/IP protocol. Key point of this protocol includes small code footprint and low network bandwidth requirement. Other features include faster response time, low power requirement, ease of scalability. All these advantages makes it an ideal candidate for communication protocol in embedded devices intended to implement IOT(internet of things) applications. More information regarding MQTT protocol can be obtained from the latest MQTT Protocol specification.



Return to CC31xx & CC32xx Home Page



Return to CC31xx Sample Applications

## Application Details

### MQTT Framework

A Simple MQTT infrastructure contains a broker(like a central hub) connected to multiple clients, each of which has the capability of publishing on any topic(token). Broker has the responsibility of sending the message published on any topic to all the subscribers of that topic. This is a very simplistic description of a MQTT network, which has a lot more features and configuration parameters.

### MQTT Library

MQTT library abstracts out the underlying intricacies of MQTT network and provides you with intuitive and easy to use APIs to implement the MQTT protocol on CC3100 device.

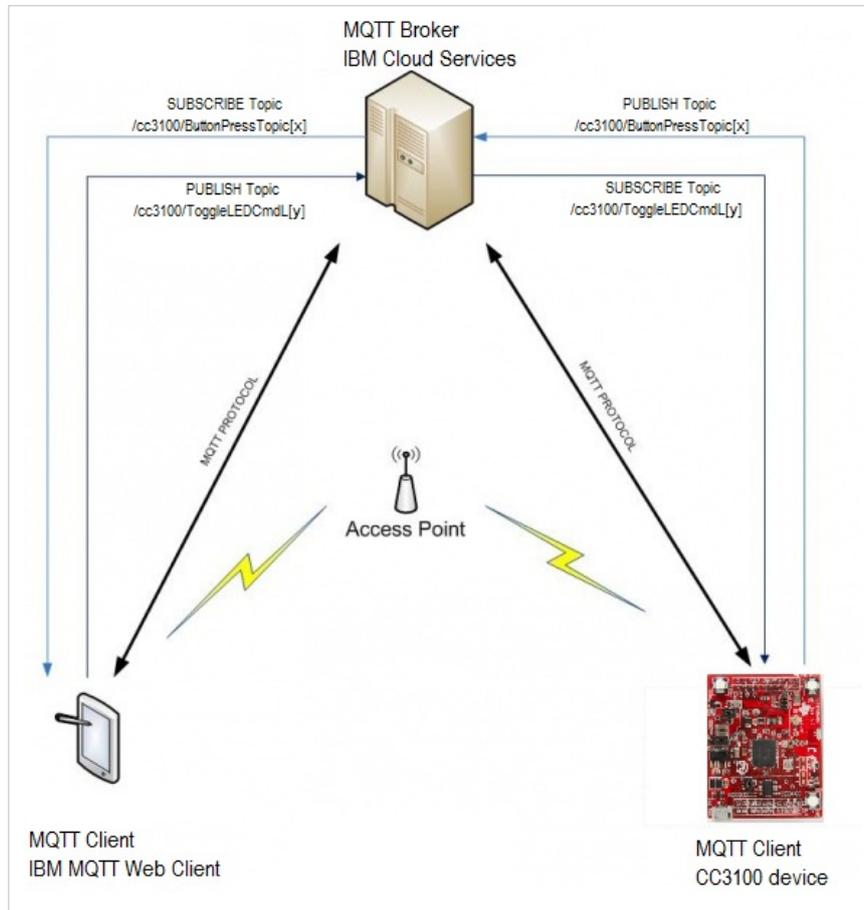
To work with CC3100, the MQTT library is provided on the MCU MSP430F5529-launchpad.

The MQTT Library API documentation can be found at: **cc3100-sdk/docs**

- **mqtt\_client.chm** - API document for the MQTT Client library.
  - **mqtt\_server.chm** - API document for the MQTT Server library.
  - **sl\_mqtt\_client.chm** - API document for the simple link wrapper around MQTT Client library.
  - **sl\_mqtt\_server.chm** - API document for the simple link wrapper around MQTT Server library.
-

## Application

This application makes use of the APIs from MQTT client library to communicate with an IBM web client using the IBM broker. Two LEDs on the MSP430F5529-launchpad device can be controlled from a web client by publishing msg on appropriate topics. Similarly, message can be published on pre-configured topics(in the code) by pressing the switch button on the MSP430F5529-launchpad device.



## Configurations

Most the parameters that user would need to modify are specified as MACROS. This can be found in `mqtt_config.h`.

```
/*Operate Lib in MQTT 3.1 mode.*/
#define MQTT_3_1_1          false /*MQTT 3.1.1 */
#define MQTT_3_1           true  /*MQTT 3.1 */

/*Defining Broker IP address and port Number*/
#define SERVER_ADDRESS      "messagesight.demos.ibm.com"
#define PORT_NUMBER        1883

/* Client ID must be unique*/
#define CLIENT_ID           "garfield007"
#define MAX_BROKER_CONN    1
#define SERVER_MODE        MQTT_3_1

/*Specifying Receive time out for the Receive task*/
```

```

#define RCV_TIMEOUT          30

/* Keep Alive Timer value in seconds.
 * Maximum time interval between messages received from a client.
 */
#define KEEP_ALIVE_TIMER    25

/* Clean session flag. The server must discard any previously
 * maintained information about the client.
 */
#define CLEAN_SESSION      true

/* Retain Flag. Used in publish message.
 * The server should hold on to the publish message after it has
 * been delivered to the current subscribers.
 */
#define RETAIN              1

/* Defining Publish Topics */
#define PUB_TOPIC_1         "/cc3100/ButtonPressTopic1"
#define PUB_TOPIC_2         "/cc3100/ButtonPressTopic2"

/* Defining number of subscribe topics */
#define SUB_TOPIC_COUNT     2

/* Defining Subscription Topics */
#define SUB_TOPIC1          "/cc3100/ToggleLEDCmdL1"
#define SUB_TOPIC2          "/cc3100/ToggleLEDCmdL2"

```

which will eventually get populated in following structures containing connection configuration and library configuration respectively.

**Note:** While using the MQTT library with CC3100/MSP430F5520-launchpad, it has the capability of connecting to a maximum of 2 brokers simultaneously. Therefore, if one wishes to do that, configuration for the new connection can be added as new element(on next index) in `connect_config`.

Also, since the MQTT client example connects only to one broker, the macro **MAX\_BROKER\_CONN** is set to 1.

```

/* Connection configuration */
connect_config usr_connect_config[] =
{
    {
        {
            SL_MQTT_NETCONN_URL,
            SERVER_ADDRESS,
            PORT_NUMBER,
            0,
            0,
            0,

```

```
        NULL
    },
    SERVER_MODE,
    true,
},
NULL,
CLIENT_ID, /* Must be unique */
NULL,
NULL,
true,
KEEP_ALIVE_TIMER,
{Mqtt_Recv, sl_MqttEvt, sl_MqttDisconnect},
SUB_TOPIC_COUNT,
{SUB_TOPIC1, SUB_TOPIC2},
{QOS2, QOS2},
{WILL_TOPIC, WILL_MSG, WILL_QOS, WILL_RETAIN},
false
}
};
```

## Source files and build environment briefly explained

### Source files

- **main.c** - The main file implementing the MQTT client functionality.
- **board.c** - The platform file for MSP430F5529-launchpad, which defines the platform related functionality.

### Build setup

- The **mqtt\_client** example has dependency on the freertos library **osi\_lib.lib** and the MQTT library **mqtt.a**. Therefore, before building the **mqtt\_client** example, one should build the freertos and the MQTT library projects within **cc3100-sdk\platform\msp430f5529lp\library\_project\_ccs**.
- While importing the library projects such as the freertos and the MQTT library projects on CCS, one should uncheck the **Copy projects into workspace** checkbox.
- To build the MQTT library on the MSP430F5529-launchpad, below are the key configuration definitions for MQTT client

The maximum number of server connections **CFG\_MQTT\_CL\_CTXS** set to **2**.

The maximum number of simultaneous server connections **CFG\_CL\_MQTT\_CTXS** set to **2**.

The buffer length **CFG\_SL\_CL\_BUF\_LEN** set to **256**.

Number of buffers **CFG\_SL\_CL\_MAX\_MQP** set to **2**.

Stack size for the MQTT receive task **CFG\_SL\_CL\_STACK** set to **1024**.

## Porting MQTT library to a specific MCU

- To use the MQTT library with a specific MCU, one should populate the below functions in `cc3100-sdk\netapps\mqtt\platform\cc31xx_sl_net.c`
  - `platform_timer_init()` - Set the timer for host processor.
  - `platform_get_time_in_secs()` - Get the time in seconds.
- The platform related data types should be defined in `cc3100-sdk\netapps\mqtt\platform\platform.h`

## Memory and code footprint of MQTT client example on MSP430F5529-launchpad

- Memory footprint: Uses 7.054 KB of RAM.
- Code footprint: Uses 44.4 KB of Flash.

## Usage

**Prerequisite:** This application requires an access-point with internet connectivity

1. Connect the board to a Windows-PC and configure the terminal-program for seeing the logs - CC31xx\_&\_CC32xx\_Terminal\_Setting\_Wiki <sup>[1]</sup> has detailed instructions for configuring the terminal-program.
2. Open `sl_common.h` and change `SSID_NAME`, `SEC_TYPE` and `PASSKEY` as per your access-point's properties. SimpleLink device will connect to this AP when the application is executed.
3. Build and run the project using CCS.
4. The message **Connection established with AP** indicates that the CC3100 is connected to the AP.
5. The message **Success: conn to Broker no. 1** indicates that the CC3100 MQTT client is now connected to the broker `messagesight.demos.ibm.com`.
6. The CC3100 MQTT client will subscribe to the topics `/cc3100/ToggleLEDCmdL1` and `/cc3100/ToggleLEDCmdL2`.
7. From another machine, open a web browser and type `http://m2m.demos.ibm.com/mqttclient/in` in the url box. You will reach the web client from IBM.
  - Note:** The web client may not connect to the broker for internet connections connected over a proxy server.
8. Connect this web client to IBM broker by filling in the required fields(use below screenshot as reference for the field values).
9. From the web client subscribe to topics `/cc3100/ButonPressTopic1` and `/cc3100/ButonPressTopic2` (use below screenshot as reference for the field values).
10. Press button **S1** on MSP430F5529-launchpad. Each button press would publish a message on topic `/cc3100/ButonPressTopic1` and `/cc3100/ButonPressTopic2` alternately. These published messages are received by the web client as shown in the below screenshot.
11. On the webclient, publish message on topics `/cc3100/ToggleLEDCmdL1` and `/cc3100/ToggleLEDCmdL2`. The CC3100 receives these messages (messages are displayed on the terminal) and you can see the corresponding LED toggling on the MSP430F5529-launchpad.

## Web client Screen shot

### Connection

### MQTT Client

Connected!

#### Connect

Server	messagesight.demos.ibm.com
Port	1883
<input type="button" value="Connect"/>	<input type="button" value="Disconnect"/>

---

Client ID	Client83897
Username	(optional)
Password	(optional)
Clean Session	<input type="button" value="OFF"/> <input checked="" type="button" value="ON"/>
SSL	<input type="button" value="OFF"/> <input checked="" type="button" value="ON"/>

MQTT Web client connected to the broker

## Subscribe and Publish

### Subscribe

Topic

QOS

### Publish

Topic

Message

QOS

Retained

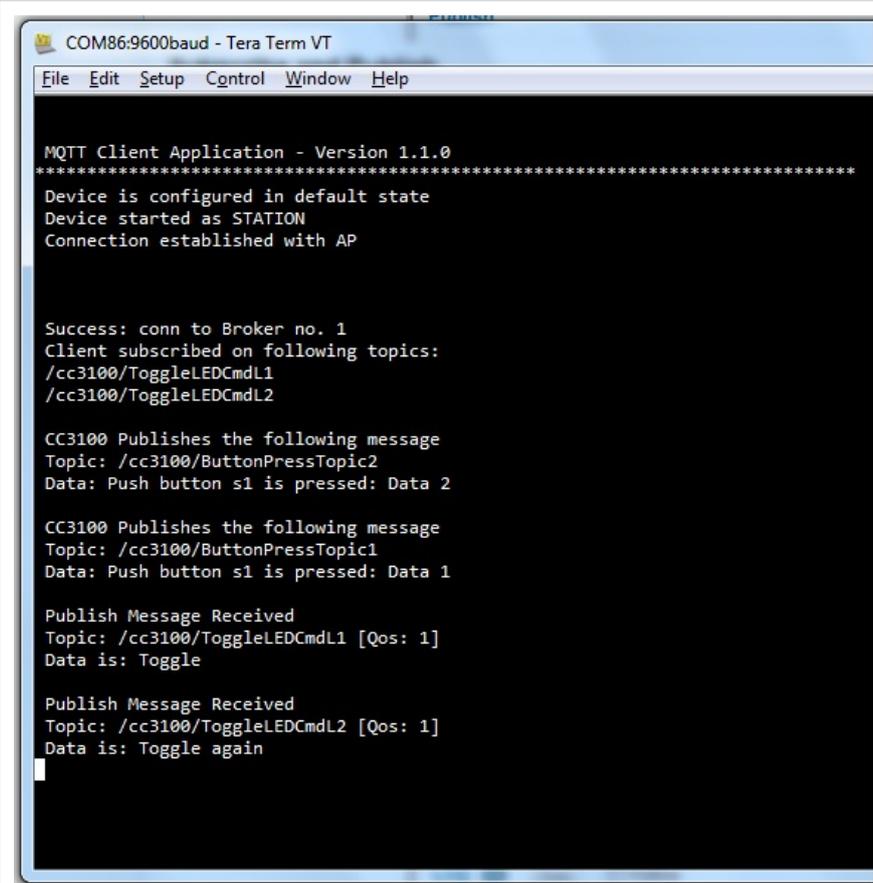
### Log

Follow

```
(13:40:33.197) Connected to messagesight.demos.ibm.com:1883
(13:41:21.152) >> [/cc3100/ButtonPressTopic1][qos 1]
[retained] Push button s1 is pressed: Data 1
(13:41:21.157) Subscribed to [/cc3100/ButtonPressTopic1][qos 1]
(13:41:34.289) >> [/cc3100/ButtonPressTopic2][qos 1]
[retained] Push button s1 is pressed: Data 2
(13:41:34.293) Subscribed to [/cc3100/ButtonPressTopic2][qos 1]
(13:41:55.504) >> [/cc3100/ButtonPressTopic2][qos 1]
Push button s1 is pressed: Data 2
(13:42:04.167) >> [/cc3100/ButtonPressTopic1][qos 1]
Push button s1 is pressed: Data 1
(13:42:25.256) << [/cc3100/ToggleLEDCmdL1][qos 1] Toggle
(13:42:37.464) << [/cc3100/ToggleLEDCmdL2][qos 1] Toggle again
```

MQTT Web client subscribes to topics. Receives the messages from CC3100. And finally publishes messages on the topics subscribed by CC3100

## Tera Term Screenshot



The screenshot shows a Tera Term terminal window titled "COM86:9600baud - Tera Term VT". The terminal displays the following text:

```
MQTT Client Application - Version 1.1.0
*****
Device is configured in default state
Device started as STATION
Connection established with AP

Success: conn to Broker no. 1
Client subscribed on following topics:
/cc3100/ToggleLEDCmdL1
/cc3100/ToggleLEDCmdL2

CC3100 Publishes the following message
Topic: /cc3100/ButtonPressTopic2
Data: Push button s1 is pressed: Data 2

CC3100 Publishes the following message
Topic: /cc3100/ButtonPressTopic1
Data: Push button s1 is pressed: Data 1

Publish Message Received
Topic: /cc3100/ToggleLEDCmdL1 [Qos: 1]
Data is: Toggle

Publish Message Received
Topic: /cc3100/ToggleLEDCmdL2 [Qos: 1]
Data is: Toggle again
```

Tera term logs for MQTT client running on CC3100/MSP430F5529-launchpad

### Suggestion

- Client ID is unique for every client. It is recommended to change the client ID for CC3100 device to avoid connection issues as shown in the configurations <sup>[2]</sup> section.

### Limitations / Known Issues

- Client can connect to at most 2 brokers at a time.
- The buffer size for the messages is limited to 256 bytes.

### References

[1] [http://processors.wiki.ti.com/index.php/CC31xx\\_&\\_CC32xx\\_Terminal\\_Setting](http://processors.wiki.ti.com/index.php/CC31xx_&_CC32xx_Terminal_Setting)

[2] [http://processors.wiki.ti.com/index.php/Staging:CC31xx\\_MQTT\\_Client#Configurations](http://processors.wiki.ti.com/index.php/Staging:CC31xx_MQTT_Client#Configurations)

# Article Sources and Contributors

**CC31xx MQTT Client** *Source:* <http://processors.wiki.ti.com/index.php?oldid=194882> *Contributors:* A0132173, Raghshenoy

## Image Sources, Licenses and Contributors

**File:Cc31xx cc32xx return home.png** *Source:* [http://processors.wiki.ti.com/index.php?title=File:Cc31xx\\_cc32xx\\_return\\_home.png](http://processors.wiki.ti.com/index.php?title=File:Cc31xx_cc32xx_return_home.png) *License:* unknown *Contributors:* A0221015

**File:Cc31xx return sample apps.png** *Source:* [http://processors.wiki.ti.com/index.php?title=File:Cc31xx\\_return\\_sample\\_apps.png](http://processors.wiki.ti.com/index.php?title=File:Cc31xx_return_sample_apps.png) *License:* unknown *Contributors:* A0221015

**Image:CC3100\_Mqtt\_client.jpg** *Source:* [http://processors.wiki.ti.com/index.php?title=File:CC3100\\_Mqtt\\_client.jpg](http://processors.wiki.ti.com/index.php?title=File:CC3100_Mqtt_client.jpg) *License:* unknown *Contributors:* Raghshenoy

**Image:MQTTConnect.png** *Source:* <http://processors.wiki.ti.com/index.php?title=File:MQTTConnect.png> *License:* unknown *Contributors:* Raghshenoy

**Image:SubscribePublish.png** *Source:* <http://processors.wiki.ti.com/index.php?title=File:SubscribePublish.png> *License:* unknown *Contributors:* Raghshenoy

**Image:MQTTScreenShot.jpg** *Source:* <http://processors.wiki.ti.com/index.php?title=File:MQTTScreenShot.jpg> *License:* unknown *Contributors:* Raghshenoy