CC32xx MQTT Client

Overview

Return to CC31xx & CC32xx Home Page

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.

Application details

MQTT Network

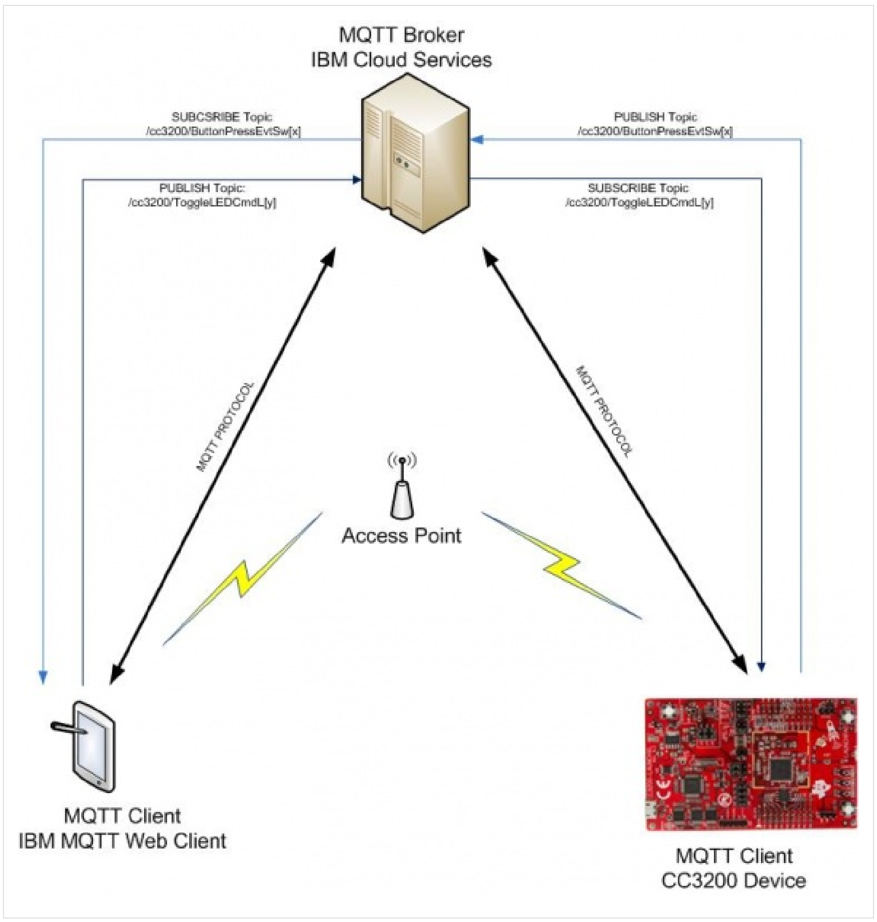
A simple MQTT network contains a server / broker (like a central hub) that can handle connections from multiple clients. Each of the connected client can publish data for any topic (token). Whereas, the server / broker is responsible for forwarding the data published for a topic by a client to all the other clients who have subscribed for that particular topic. This is a very simplistic description of a MQTT network to set the tone for the sample application provided in SDK.

MQTT Library

MQTT library abstracts out the underlying intricacies of MQTT network and provide you with an intuitive and easy to use APIs to implement the MQTT protocol on CC3200 device.

Application

This application make use of the APIs from MQTT client library to communicate with an IBM web client using the IBM broker. Three LEDs on the CC3200 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 buttons on the CC3200 device.



Configurations

Most the parameters user will need to modify are specified as MACROs

```
/*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

#define MAX_BROKER_CONN          1

#define SERVER_MODE                        MQTT_3_1
/*Specifying Receive time out for the Receive task*/
#define RCV_TIMEOUT                    30

/*Background receive task priority*/
#define TASK_PRIORITY                     3

/* Keep Alive Timer value*/
#define KEEP_ALIVE_TIMER             25
```

```
/*Clean session flag*/
#define CLEAN_SESSION                    true

/*Retain Flag. Used in publish message. */
#define RETAIN                                1

/*Defining Publish Topic*/
#define PUB_TOPIC_FOR_SW3              "/cc3200/ButtonPressEvtSw3"
#define PUB_TOPIC_FOR_SW2             "/cc3200/ButtonPressEvtSw2"

/*Defining Number of topics*/
#define TOPIC_COUNT                   3

/*Defining Subscription Topic Values*/
#define TOPIC1                               "/cc3200/ToggleLEDCmdL1"
#define TOPIC2                               "/cc3200/ToggleLEDCmdL2"
#define TOPIC3                               "/cc3200/ToggleLEDCmdL3"
```

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

Note: CC3200 MQTT library has the capability of connecting to multiple brokers simultaneously. So, if one wishes to do that, configuration for the new connection can be added as new element(on next index) in connect_config.

```
/* connection configuration */
connect_config usr_connect_config[] =
{
    {
        {
            {
                  SL_MQTT_NETCONN_URL,
                SERVER_ADDRESS,
                PORT_NUMBER,
                0,
                0,
                0,
                NULL
            },
            SERVER_MODE,
            true,
        },
        NULL,
        "user1",
        NULL,
        NULL,
        true,
        KEEP_ALIVE_TIMER,
        {Mqtt_Recv, sl_MqttEvt, sl_MqttDisconnect},
        TOPIC_COUNT,
        {TOPIC1, TOPIC2, TOPIC3},
        {QOS2, QOS2, QOS2},
        {WILL_TOPIC,WILL_MSG,WILL_QOS,WILL_RETAIN},
        false
    }
};

/* library configuration */
SlMqttClientLibCfg_t Mqtt_Client={
    1882,
    TASK_PRIORITY,
    30,
    true,
    UART_PRINT
};
```

Source Files briefly explained

- **main.c** - The main file implementing the mqtt client.

Supporting Files

- **pinmux.c** - Generated by the PinMUX utility.

Common Files

- **button_if.c - Interface file to handle button click events**
- **gpio_if.c** - Basic GPIO interface APIs. Used to control the LEDs.
- **network_if.c** - Common functions to handle connection to AP and FreeRTOS hook functions.
- **startup_ccs.c** - CCS related functions
- **startup_ewarm.c** - IAR related functions
- **timer_if.c** - Wrapper functions for timer module driver
- **uart_if.c** - To display status information over the UART

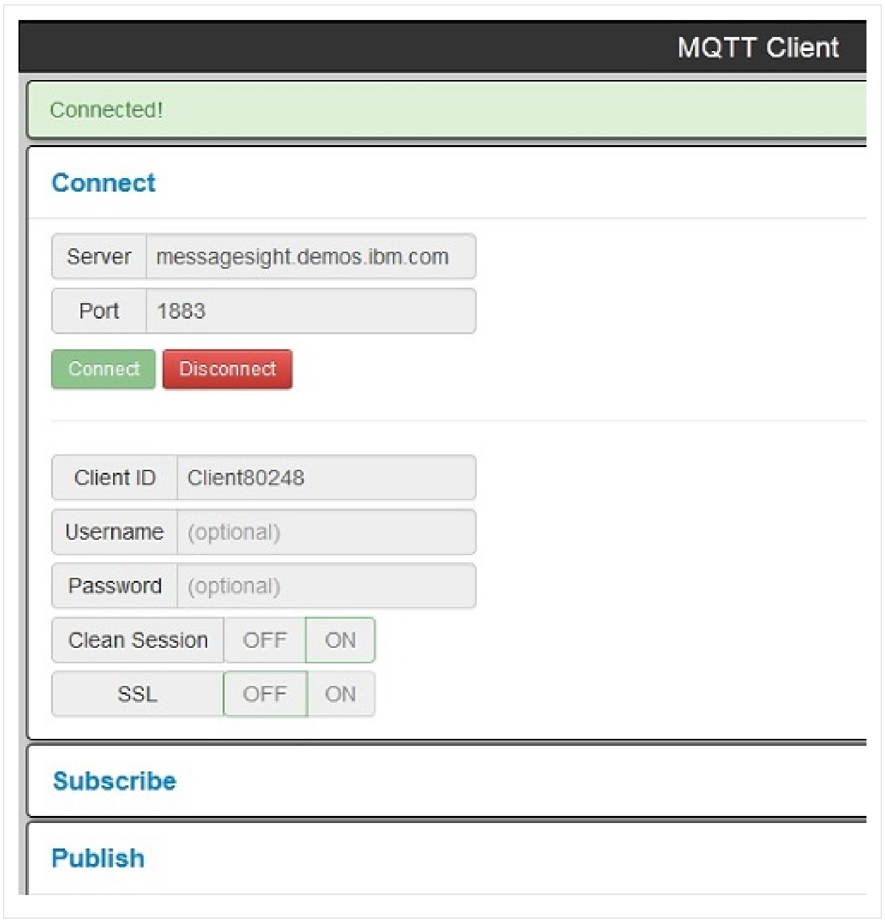
Usage

1. Subscribe to the **/cc3200/ButtonPressEvtSw2** and **/cc3200/ButtonPressEvtSw3** topics from the web client.#Setup a serial communication application (HyperTerminal/TeraTerm). For detail info visit CC31xx & CC32xx Terminal Setting
   On the host PC, open a hyperterminal, with the following settings
   - **Port:** Enumerated COM port
   - **Baud rate:** 115200
   - **Data:** 8 bit
   - **Parity:** None
   - **Stop:** 1 bit
   - **Flow control:** None
2. Run the reference application (Flashing the bin/IAR/CCS).
3. Application requires the AP to have the internet connectivity.
4. On the Hyperterminal, in case the connection to this default AP is unsuccessful, the user is prompted to enter the AP details.

5. The GREEN LED denotes that the network processor has come up successfully, RED LED continuously blinks as long as a connection with AP is not established. Once established, the RED LED stays continuously ON for one second and then all the LEDs will be turned off.
6. From another machine, open a web browser and type http://m2m.demos.ibm.com/mqttclient/in the url box. You will reach the web client from IBM.
7. Connect this web client to IBM broker by filling in the required fields(use below screenshot as reference for the field values).
8. Publish on **/cc3200/ToggleLEDCmdL1, /cc3200/ToggleLEDCmdL2** or**/cc3200/ToggleLEDCmdL3** to control LED9, LED10 and LED11 respectively.
9. Press the SW2 or SW3 switch on CC3200 device to publish message on **/cc3200/ButtonPressEvtSw2** or **/cc3200/ButtonPressEvtSw3** respectively.

Web Client Screentshot

Connection


MQTT Client

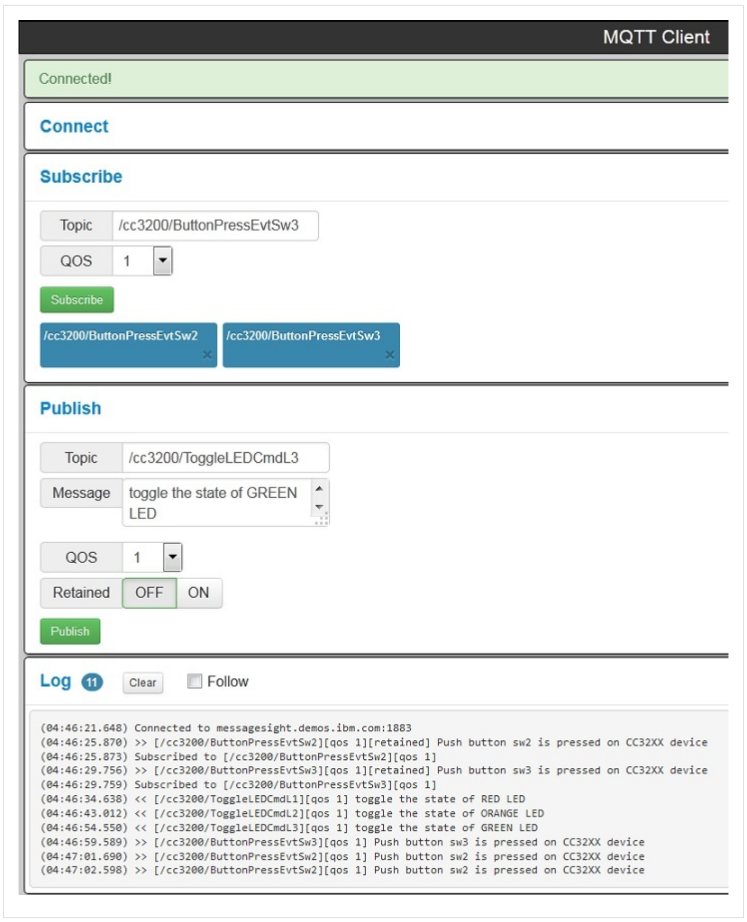
Connected!

Connect

Server	messagesight.demos.ibm.com
Port	1883

Client ID	Client80248
Username	(optional)
Password	(optional)
Clean Session	OFF [✓ ON]
SSL	[✓ OFF] ON

Subscribe and Publish

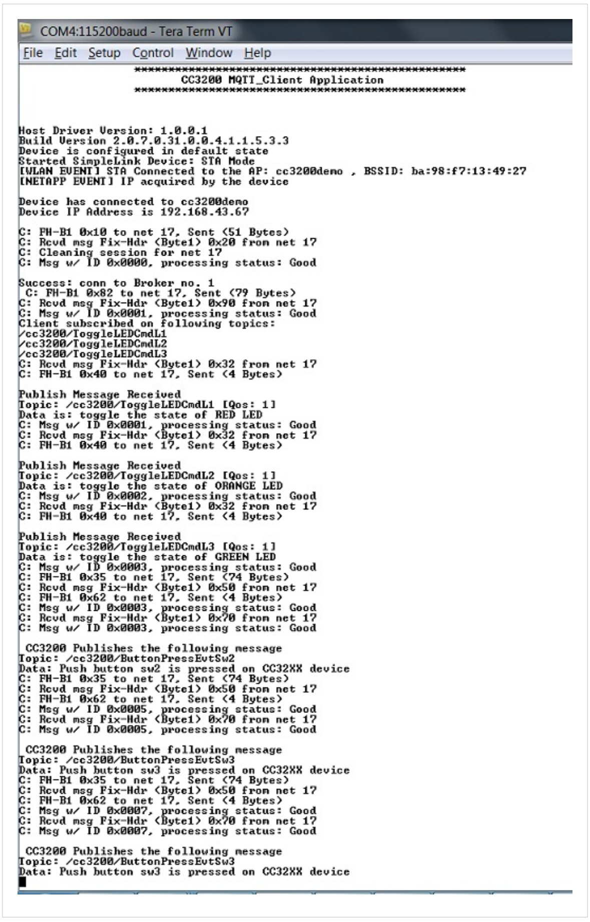


Topic /cc3200/ButtonPressEvtSw3
QOS 1

Topic /cc3200/ToggleLEDCmdL3
Message toggle the state of GREEN LED
QOS 1
Retained [✓ OFF] ON

```
(04:46:21.648) Connected to messagesight.demos.ibm.com:1883
(04:46:25.870) >> [/cc3200/ButtonPressEvtSw2][qos 1][retained] Push button sw2 is pressed on CC32XX device
(04:46:25.873) Subscribed to [/cc3200/ButtonPressEvtSw2][qos 1]
(04:46:29.756) >> [/cc3200/ButtonPressEvtSw3][qos 1][retained] Push button sw3 is pressed on CC32XX device
(04:46:29.759) Subscribed to [/cc3200/ButtonPressEvtSw3][qos 1]
(04:46:34.638) << [/cc3200/ToggleLEDCmdL1][qos 1] toggle the state of RED LED
(04:46:43.012) << [/cc3200/ToggleLEDCmdL2][qos 1] toggle the state of ORANGE LED
(04:46:54.550) << [/cc3200/ToggleLEDCmdL3][qos 1] toggle the state of GREEN LED
(04:46:59.589) >> [/cc3200/ButtonPressEvtSw3][qos 1] Push button sw3 is pressed on CC32XX device
(04:47:01.690) >> [/cc3200/ButtonPressEvtSw2][qos 1] Push button sw2 is pressed on CC32XX device
(04:47:02.598) >> [/cc3200/ButtonPressEvtSw2][qos 1] Push button sw2 is pressed on CC32XX device
```

Tera Term Screenshot


COM4:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
                  ***************************************************
                          CC3200 MQTT_Client Application
                  ***************************************************


Host Driver Version: 1.0.0.1
Build Version 2.0.7.0.31.0.0.4.1.1.5.3.3
Device is configured in default state
Started SimpleLink Device: STA Mode
[WLAN EVENT] STA Connected to the AP: cc3200demo , BSSID: ba:98:f7:13:49:27
[NETAPP EVENT] IP acquired by the device

Device has connected to cc3200demo
Device IP Address is 192.168.43.67

C: FH-B1 0x10 to net 17, Sent (51 Bytes)
C: Rcvd msg Fix-Hdr (Byte1) 0x20 from net 17
C: Cleaning session for net 17
C: Msg w/ ID 0x0000, processing status: Good

Success: conn to Broker no. 1
 C: FH-B1 0x82 to net 17, Sent (79 Bytes)
C: Rcvd msg Fix-Hdr (Byte1) 0x90 from net 17
C: Msg w/ ID 0x0001, processing status: Good
Client subscribed on following topics:
/cc3200/ToggleLEDCmdL1
/cc3200/ToggleLEDCmdL2
/cc3200/ToggleLEDCmdL3
C: Rcvd msg Fix-Hdr (Byte1) 0x32 from net 17
C: FH-B1 0x40 to net 17, Sent (4 Bytes)

Publish Message Received
Topic: /cc3200/ToggleLEDCmdL1 [Qos: 1]
Data is: toggle the state of RED LED
C: Msg w/ ID 0x0001, processing status: Good
C: Rcvd msg Fix-Hdr (Byte1) 0x32 from net 17
C: FH-B1 0x40 to net 17, Sent (4 Bytes)

Publish Message Received
Topic: /cc3200/ToggleLEDCmdL2 [Qos: 1]
Data is: toggle the state of ORANGE LED
C: Msg w/ ID 0x0002, processing status: Good
C: Rcvd msg Fix-Hdr (Byte1) 0x32 from net 17
C: FH-B1 0x40 to net 17, Sent (4 Bytes)

Publish Message Received
Topic: /cc3200/ToggleLEDCmdL3 [Qos: 1]
Data is: toggle the state of GREEN LED
C: Msg w/ ID 0x0003, processing status: Good
C: FH-B1 0x35 to net 17, Sent (74 Bytes)
C: Rcvd msg Fix-Hdr (Byte1) 0x50 from net 17
C: FH-B1 0x62 to net 17, Sent (4 Bytes)
C: Msg w/ ID 0x0003, processing status: Good
C: Rcvd msg Fix-Hdr (Byte1) 0x70 from net 17
C: Msg w/ ID 0x0003, processing status: Good

 CC3200 Publishes the following message
Topic: /cc3200/ButtonPressEvtSw2
Data: Push button sw2 is pressed on CC32XX device
C: FH-B1 0x35 to net 17, Sent (74 Bytes)
C: Rcvd msg Fix-Hdr (Byte1) 0x50 from net 17
C: FH-B1 0x62 to net 17, Sent (4 Bytes)
C: Msg w/ ID 0x0005, processing status: Good
C: Rcvd msg Fix-Hdr (Byte1) 0x70 from net 17
C: Msg w/ ID 0x0005, processing status: Good

 CC3200 Publishes the following message
Topic: /cc3200/ButtonPressEvtSw3
Data: Push button sw3 is pressed on CC32XX device
C: FH-B1 0x35 to net 17, Sent (74 Bytes)
C: Rcvd msg Fix-Hdr (Byte1) 0x50 from net 17
C: FH-B1 0x62 to net 17, Sent (4 Bytes)
C: Msg w/ ID 0x0007, processing status: Good
C: Rcvd msg Fix-Hdr (Byte1) 0x70 from net 17
C: Msg w/ ID 0x0007, processing status: Good

 CC3200 Publishes the following message
Topic: /cc3200/ButtonPressEvtSw3
Data: Push button sw3 is pressed on CC32XX device
```

Suggestion

Note: Client name is unique for every client. It is recommended to change the client ID for CC3200 device to avoid connection issues.

Additional Info

Note: If anyone wishes to have prints(on serial comm port) about the packet level details of MQTT packets, just define **DEBUG_NET_DEV** as the predefined symbol while building the mqtt library.

Limitations/Known Issues

1. Client can connect to at most 4 brokers at a time

Article Sources and Contributors

CC32xx MQTT Client *Source*: http://processors.wiki.ti.com/index.php?oldid=195086 *Contributors*: Jitgupta

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 *License*: unknown *Contributors*: A0221015
File:Mqtt_client_application_model.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:Mqtt_client_application_model.jpg *License*: unknown *Contributors*: Jitgupta
File:mqtt_connection_screenshot.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:Mqtt_connection_screenshot.jpg *License*: unknown *Contributors*: Jitgupta
File:mqtt_sub_pub_screenshot.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:Mqtt_sub_pub_screenshot.jpg *License*: unknown *Contributors*: Jitgupta
File:Mqtt client teraterm screenshot.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:Mqtt_client_teraterm_screenshot.jpg *License*: unknown *Contributors*: Jitgupta