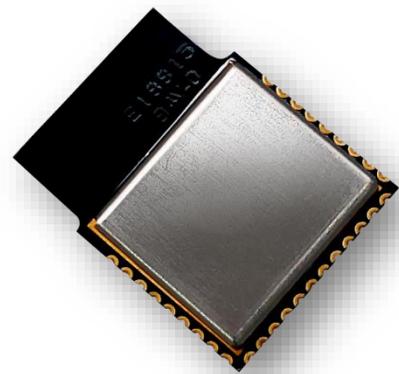


RYWB116

MQTT Client Application User Guide



PROTOCOL OVERVIEW

MQTT is a publish-subscribe based "light weight" messaging protocol for use on top of the TCP/IP protocol.

The MQTT connection itself is always between one client and the broker, no client is connected to another client directly.

MQTT CLIENT

A MQTT client is any device from a micro controller up to a full fledged server, that has a MQTT library running and is connecting to an MQTT broker over any kind of network.

MQTT Clients can share the information on a particular topic using MQTT protocol.

MQTT clients connect to the MQTT broker using TCP connection and can subscribe and publish on any desired topic.

The other clients which are subscribed for that topic will receive the published messages.

MQTT BROKER

The publish-subscribe messaging pattern requires a message broker.

The broker is primarily responsible for receiving all messages, filtering them, decide who is interested in it and then sending the message to all subscribed clients.

It also holds the session of all persisted clients including subscriptions and missed messages.

Another responsibility of the broker is the authentication and authorization of clients.

A simple demonstration of subscribing and publishing of temperature is shown below

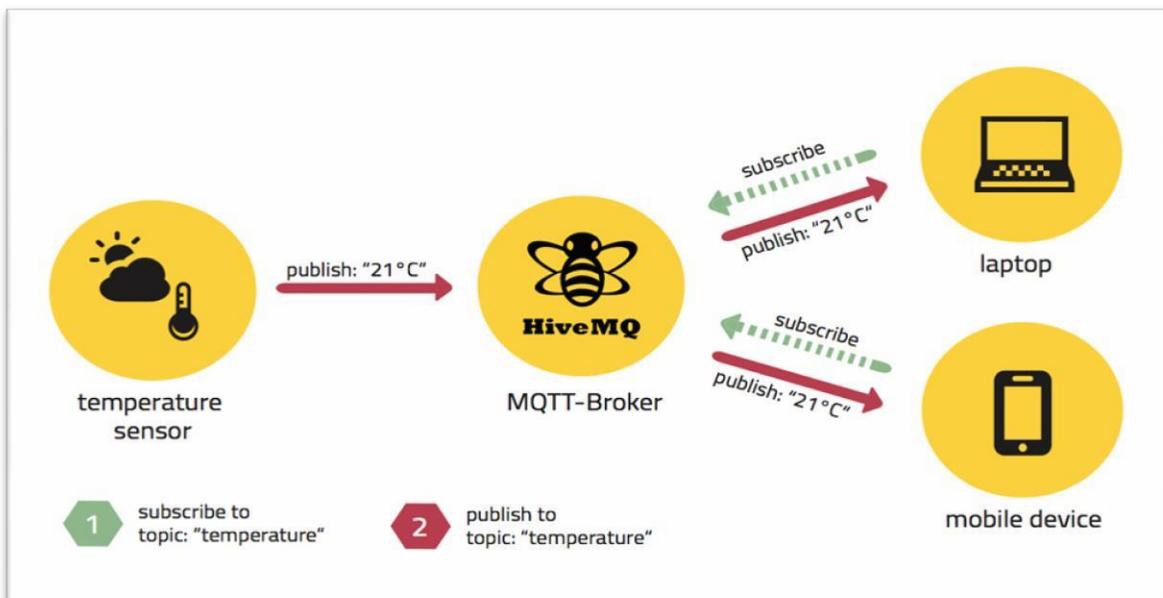


Figure 1: Demonstration of MQTT protocol

APPLICATION OVERVIEW

This is a sample application which demonstrates the following MQTT functionality.

1. MQTT client (RYWB116) connects to the MQTT broker which is running on Windows PC 2
2. Subscribes to the given topic
3. Publishes MQTT message on that topic and receives the messages which are published on that topic

SETUP REQUIRED

1. Windows PC 1 with Coocox IDE
2. RYWB116 device
3. WLAN Access point
4. Windows PC 2 with MQTT broker installed in it
5. Windows PC 3 with MQTT utility which acts as MQTT client

NOTE:

1. MQTT broker for different OS platforms can be downloaded from the link_
<http://mosquitto.org/download/>
2. MQTT Utility which has to be installed in Windows PC 3 can be downloaded from the below given link
<https://www.eclipse.org/downloads/download.php?file=/paho/1.0/org.eclipse.paho.mqtt.utility-1.0.0.jar>

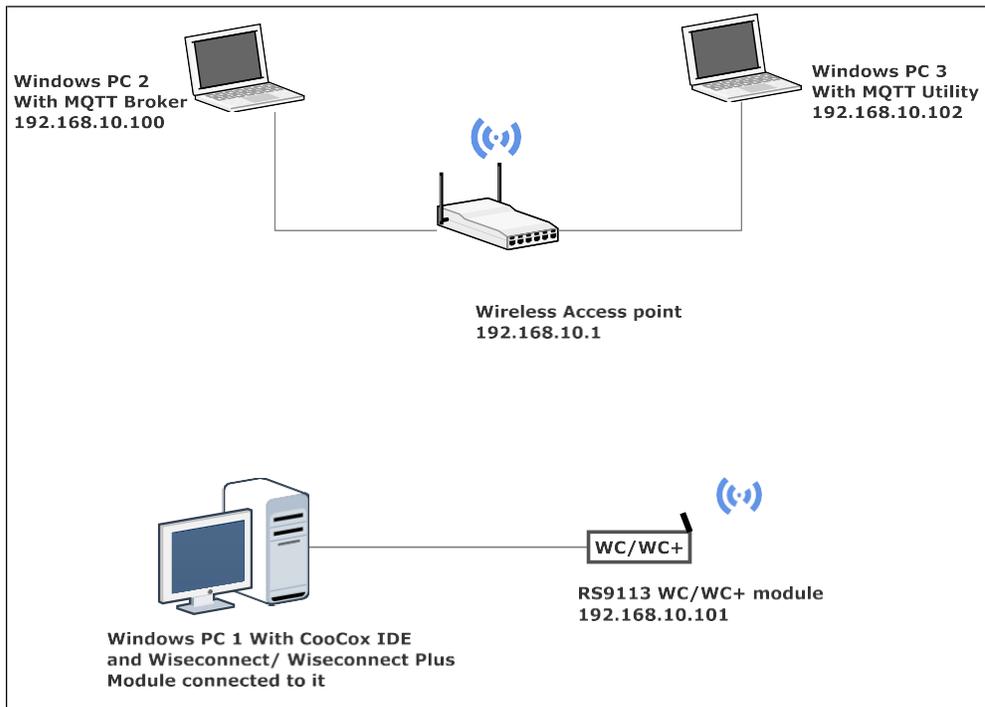


Figure 2: MQTT Client demo set up

DESCRIPTION

This application is used to configure RYWB116 connected to a WLAN Access point.

This example witness MQTT Client functionality of the RYWB116. This example demonstrates how message exchanges happens between two MQTT clients on the subscribed topic.

CONFIGURING THE APPLICATION

Edit the `rsi_mqtt.c` file in the following path.

`sapis/examples/wlan/mqtt_client` From given configuration,

1. Configure the module to join to the specific Access point by configuring following parameters

`SSID` refers to the name of the Access point to connect.

`PSK` refers to the secret key if the Access point was configured in WPA/WPA2 security modes.

```
#define SSID                "<ap_name>"
#define CHANNEL_NO         0
#define SECURITY_TYPE      <security-type>
#define PSK                 "<psk>"
```

2. To configure the IP of the RYWB116 module in DHCP mode

```
#define DHCP_MODE          <dhcp mode>
```

To configure IP through dhcp set `DHCP_MODE` to 1, else configure to 0

3. To configure static IP address

IP address to be configured to the device should be in long format and in little endian byte order. The following parameters are valid only if `DHCP_MODE` is 1.

Example: To configure "192.168.10.1" as IP address, update the macro `DEVICE_IP` as `0x010AA8C0`.

```
#define DEVICE_IP          0X010AA8C0
```

IP address of the gateway should also be in long format and in little endian byte order

Example: To configure "192.168.10.1" as Gateway, update the macro `GATEWAY` as `0x010AA8C0`

```
#define GATEWAY            0x010AA8C0
```

IP address of the network mask should also be in long format and in little endian byte order

Example: To configure "255.255.255.0" as network mask, update the macro `NETMASK` as `0x00FFFFFF`

```
#define NETMASK            0x00FFFFFF
```

4. Connect to the MQTT broker/server by configuring the following parameters

MQTT server IP address should be in long format and in little endian byte order

```
#define SERVER_IP_ADDRESS 0x640AA8C0
```

Server port number of MQTT broker/server

```
#define SERVER_PORT 1883
```

Client port number of the MQTT client

```
#define CLIENT_PORT 1883
```

5. The following parameters are configured if OS is used.

WLAN task priority is given and this should be of low priority

```
#define RSI_WLAN_TASK_PRIORITY 1
```

Driver task priority is given and this should be of highest priority

```
#define RSI_DRIVER_TASK_PRIORITY 1
```

WLAN Task stack size is configured by this macro

```
#define RSI_WLAN_TASK_STACK_SIZE 500
```

Driver Task stack size is configured by this macro

```
#define RSI_DRIVER_TASK_STACK_SIZE 500
```

6. The following parameters are used for the MQTT client specific information configuration

MQTT client Keep alive period

```
#define RSI_KEEP_ALIVE_PERIOD 100
```

Memory to initialize MQTT client Info structure

```
#define MQTT_CLIENT_INIT_BUFF_LEN 3500
```

Global buffer or memory which is used for MQTT client initialization.

This buffer is used for the MQTT client information storage.

```
uint8_t mqtt_client_buffer[MQTT_CLIENT_INIT_BUFF_LEN]
```

QOS of the message. This QOS is MQTT protocol specific and the valid values are 0,1 and 2. This field indicates the level of assurance for delivery of an Application Message. The QoS levels are

0 - At most once delivery

1 - At least once delivery

2 - Exactly once delivery

```
#define QOS 0
```

MQTT topic to which client is supposed to subscribe

```
#define RSI_MQTT_TOPIC "REYAX"
```

MQTT Message to publish on the topic subscribed

```
uint8_t publish_message[] ="THIS IS MQTT CLIENT DEMO  
FROM REYAX"
```

MQTT Client ID with which MQTT client connects to MQTT broker/server

```
uint8_t clientID[] = "MQTTCLIENT"
```

user name for login credentials

```
int8_t username[] = "username"
```

password for login credentials

```
int8_t password[] = "password"
```

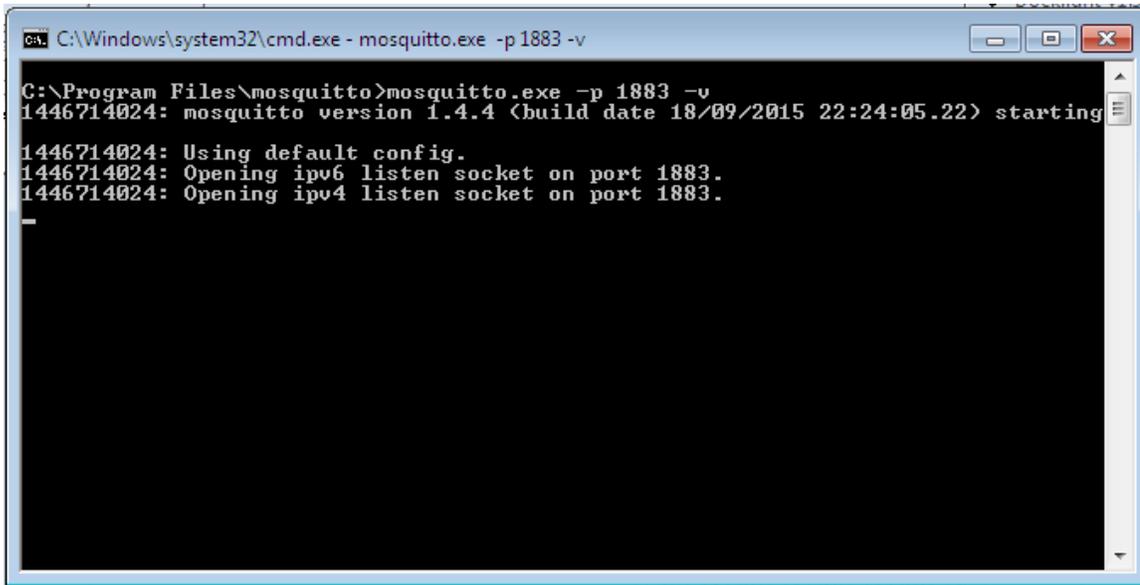
Edit the Wlan configuration file:

sapis/include/rsi_wlan_config.h

CONCURRENT_MODE	DISABLE
RSI_FEATURE_BIT_MAP	FEAT_SECURITY_OPEN
RSI_TCP_IP_BYPASS	DISABLE
RSI_TCP_IP_FEATURE_BIT_MAP	TCP_IP_FEAT_DHCPV4_CLIENT
RSI_CUSTOM_FEATURE_BIT_MAP	0
RSI_BAND	RSI_BAND_2P4GHZ

EXECUTING THE APPLICATION

1. Connect Windows PC 3 and Windows PC 2 to the access point
2. Run MQTT broker application with the server port configured `SERVER_PORT` on Windows PC 2 as shown in the screen shot below



```

C:\Windows\system32\cmd.exe - mosquitto.exe -p 1883 -v

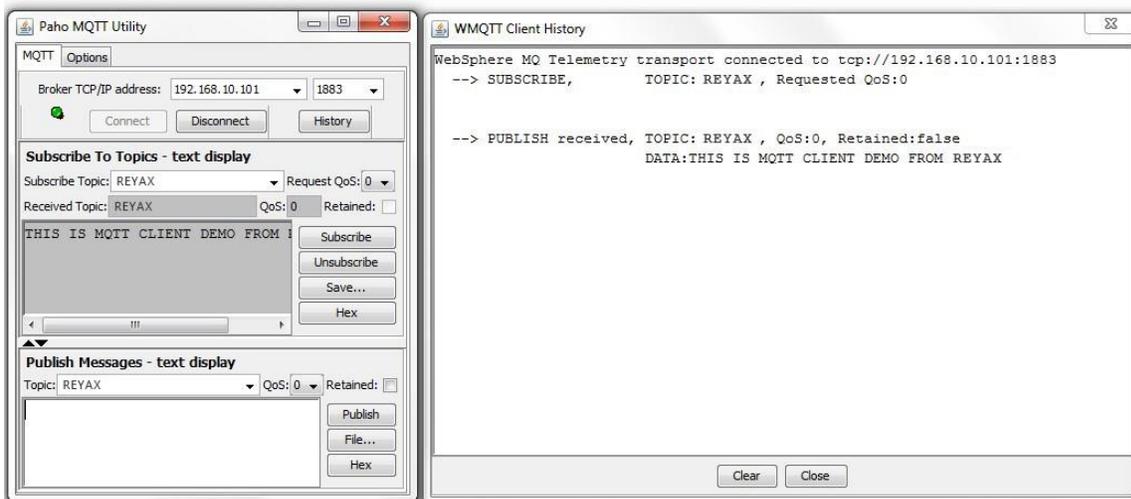
C:\Program Files\mosquitto>mosquitto.exe -p 1883 -v
1446714024: mosquitto version 1.4.4 (build date 18/09/2015 22:24:05.22) starting
1446714024: Using default config.
1446714024: Opening ipv6 listen socket on port 1883.
1446714024: Opening ipv4 listen socket on port 1883.
  
```

Figure 3: MQTT Broker application listening state

3. Open the MQTT client Utility in Windows PC 3 and connect to the MQTT broker which is running in the Windows PC 2 and subscribe to the topic given in the application file `RSI_MQTT_TOPIC`. The screen shot is given below for the MQTT utility

Give the port number of the MQTT broker in the port field as 1883

Enter the topic to subscribe as "REYAX"



4. Connect RYWB116 to the Windows PC running Cocoox IDE.
5. Configure the macros in the files located at
`sapis/examples/wlan/rsi_mqtt.c`
`sapis/include/rsi_wlan_config.h`
6. Build and launch the application.
7. After the program gets executed, RYWB116 would be connected to same Access point having the configuration same that of in the application and get IP.
8. Once the RYWB116 is connected to the MQTT broker, connected MQTT clients information message is displayed in the MQTT broker application.
9. After the subscription to the topic "**REYAX**", publish a message ["**THIS IS MQTT CLIENT DEMO FROM REYAX**"] on the subscribed topic.
10. Now this message is received by the MQTT utility which is running on Windows PC 3.
11. Now publish a message using MQTT Utility on the same topic. Now this message is message received by RYWB116.

NOTE:1. Multiple MQTT client instances can be created

LIMITATIONS

MQTT client application keeps on polling for the data to receive on the subscribed topic irrespective of receive timeout mentioned in the `rsi_mqtt_poll_for_rcv_data` API.



Taiwan: sales@reyax.com
China: sales@reyax.com.cn
http://reyax.com