Using MQTT Client with RAFW Connectivity Devices

This document describes how to use the MQTT client application with Dynamic Power Management (DPM) support on the RA6W1/RA6W2/RRQ61001/RRQ61051, enabling low-power MQTT publish/subscribe operations over Wi-Fi.

Contents

1. Introduction **3**
2. System Overview **3**
2.1 Software Requirements 4
2.2 Hardware Requirements 5
3. Set Up and Configure Hardware **5**
4. MQTT Client with DPM Setup **6**
4.1 Set Up EK-RA6W1/EK-RA6W2 6
4.2 Program Firmware Image 6
4.3 Set Up Mosquitto Broker 6
4.3.1 Installation 6
4.3.2 Generate TLS Certificates 6
4.3.3 Broker Configuration (mosquitto.conf) 7
4.3.4 Start the Mosquitto Broker with TLS 7
5. FSP Stack Configuration for MQTT Client with DPM **8**
5.1 Configure MQTT Parameters 9
5.2 Flash Firmware 10
5.3 Set Up Station Mode with DPM 10
5.4 Test Case 1: Device Publishes, Host PC Subscribes 12
5.5 Test Case 2: Host PC Publishes, Device Subscribe 13
6. Revision History **15**

Figures

Figure 1. Sequence diagram 4
Figure 2. Enable MQTT in Wi-Fi stack 8
Figure 3. MQTT stack view 8
Figure 4. MQTT cert properties 9
Figure 5. Host PC IPv4 address 9
Figure 6. MQTT connection and DPM entry 11
Figure 7. Device publish log (RTC wake-up) 12
Figure 8. Host PC subscriber receiving messages 13
Figure 9. Host PC publishes messages 13
Figure 10. Device receive log 14

© 2026 Renesas Electronics

Tables

Table 1. Jumper configuration settings 5
Table 2. MQTT broker configuration in mqtt.h 10
Table 3. Tera Term Serial Port settings 11

Terms and Definitions

AP	Access Point
DPM	Dynamic Power Management
DTIM	Delivery Traffic Indication Message
EVK	Evaluation Kit
FSP	Flexible Software Package
MCU	Microcontroller unit
MQTT	Message Queuing Telemetry Transport
RTC	Real-Time Clock
SNTP	Simple Network Time Protocol
SSID	Service Set Identifier
TIM	Traffic Indication Map
TLS	Transport Layer Security
UART	Universal Asynchronous Receiver/Transmitter
Wi-Fi	Wireless Fidelity

1. Introduction

The Renesas RA6W1 module is a Wi-Fi-enabled MCU designed for wireless connectivity applications. In this example, RA6W1 operates as an MQTT client that connects to an MQTT broker over a Wi-Fi network, supporting both publishing and subscribe operations. The application integrates Dynamic Power Management (DPM), allowing the device to enter a low-power sleep state between MQTT operations. The device periodically wakes from DPM to publish messages and can also be woken by incoming subscribed messages buffered at the Access Point. This makes the application suitable for battery-powered IoT use cases where low average power consumption is critical. The MQTT connection supports both non-TLS (port 1883) and TLS-secured (port 8883) communication. When TLS is enabled, mutual authentication is performed using a Root CA certificate, a client certificate, and a client private key.

2. System Overview

The RA6W1/RA6W2 operates as an MQTT client connecting to a Mosquitto broker on a host PC over Wi-Fi, publishing and receiving messages between DPM low-power sleep cycles (Figure 1). The host PC runs the broker, test clients (`mosquitto_pub/mosquitto_sub`), and a serial terminal for console monitoring.
The system consists of three components:

- **RA6W1 Device (MQTT Client)** — connects to a Wi-Fi AP in Station mode, publishes messages on rrq61x_pub during RTC wake-ups, receives messages on `rrq61x_sub` during TIM UC wake-ups, and stores MQTT configuration in DPM retention memory across sleep cycles.
- **MQTT Broker (Mosquitto)** — runs on the host PC on the same network. Routes messages between the device and test clients. Supports TLS (port 8883) and non-TLS (port 1883).
- **Host PC (Test Client)** — uses `mosquitto_pub` and `mosquitto_sub` to exchange messages with the device. Runs Tera Term to monitor device console output over UART.

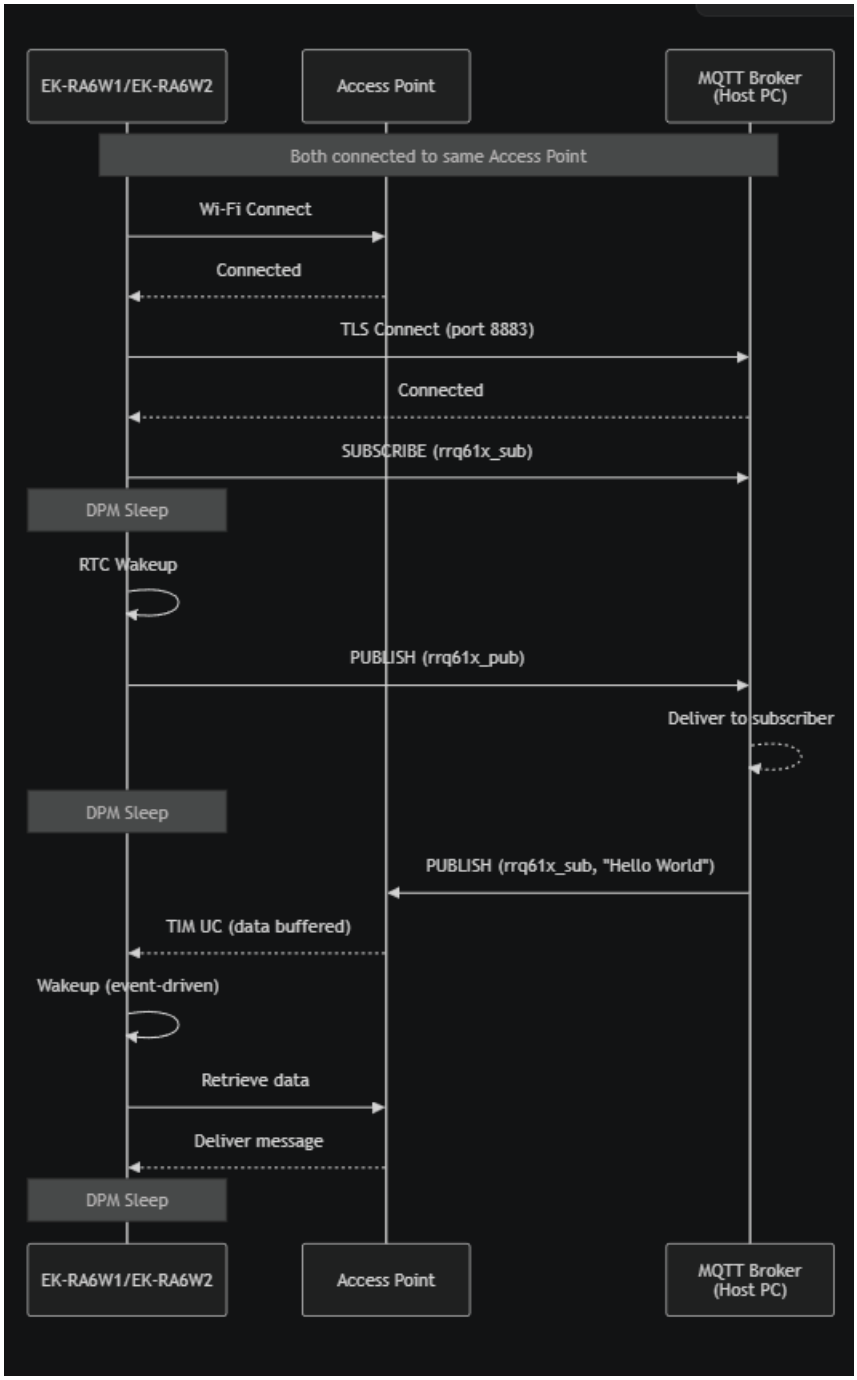


Figure 1. Sequence diagram

2.1 Software Requirements

The following software tools and packages are required to complete the project:

- Renesas Electronics Flexible Software Package (FSP): Version 2.1.0 or later.
- e^2 studio IDE: Version 2026-04.2 or later.
- GCC Arm Embedded Toolchain: Version 13.3.1.arm-13-24.
- Serial Terminal Application: Tera Term or similar tools for logging and monitoring output.

2.2 Hardware Requirements

- Renesas RA6W1/RA6W2 Evaluation Kit (EVK).
- RA6W1/RA6W2 MCUwith ultra-low power consumption.
- USB Type-C Cable: For programming and debugging.
- Jumper Wire (Female to Female): For connecting the RA6W1/RA6W2 MCU to the motherboard.
- Host PC: For running the Mosquitto MQTT broker and test clients.
- Wi-Fi Access Point (AP): With network connectivity. Both the device and the host PC must be connected to the same AP.

3. Set Up and Configure Hardware

To connect hardware:

1. Connect the RA6W1/RA6W2 MCU to the Evaluation Kit (EVK) using the extension socket.
2. Connect the USB Debug port on the EVK to the host PC for programming and debugging using a USB Type-C cable.
3. Set up jumpers as shown in Table 1.

Table 1. Jumper configuration settings

Jumper	Setting	Description
UART (RX)	P0_00	-
UART (TX)	P0_01	-
J102	{2,3}	Current Measurement
J105	{2,3}	Voltage Selector
J106	{2,3}	Voltage Selector
J107	{2,3}	Voltage Selector
J210	{1,2}	Reset
SWCLK(c)	SWCLK	-
SWDIO(c)	SWDIO	-

4. MQTT Client with DPM Setup

To set up MQTT Client with DPM you need to connect EK-RA6W1/EK-RA6W2, flash firmware image, and set up Mosquitto Broker.

4.1 Set Up EK-RA6W1/EK-RA6W2

1. Connect the EK-RA6W1/EK-RA6W2 evaluation board to the host PC using the USB debug port.
2. Ensure both the device and the host PC (running the Mosquitto MQTT broker) are connected to the same Wi-Fi Access Point.

4.2 Program Firmware Image

Use the MQTT client DPM project corresponding to the specific board:

- RA6W1: `wifi_mqtt_client_dpm_ek_ra6w1`
- RA6W2: `wifi_msqtt_client_dpm_ek_ra6w2`.

Build and flash the firmware image using e² studio with J-Link. No external flash programming scripts are required (see Section 5.2 Flash Firmware).

4.3 Set Up Mosquitto Broker

Install the Mosquitto MQTT broker on the host PC. The broker and command-line tools (`mosquitto_pub`, `mosquitto_sub`) are included in the installation. The broker must be configured and running before starting the device.

4.3.1 Installation

- Windows — Download and run the installer from https://mosquitto.org/download/.
- Linux (Ubuntu/Debian).

```
$ sudo apt update
$ sudo apt install mosquito
$ sudo apt install mosquitto mosquitto-clients
```

4.3.2 Generate TLS Certificates

Run the following OpenSSL commands on the host PC to generate the Root CA and client certificates:

1. Generate Root CA Key + Certificate.

```
openssl req -x509 -new -key rootCA.key -sha256 -out rootCA.crt
openssl req -x509 -new -key rootCA.key -sha256 -out rootCA.crt
```

2. Generate Client Private Key + CSR.

```
openssl ecparam -name prime256v1 -genkey -noout -out client.key && \
openssl req -new -key client.key -out client.csr
```

3. Sign Client Certificate Using Root CA.

```
 openssl x509 -req -in client.csr \
-CA rootCA.crt -CAkey rootCA.key -CAcreateserial \
-out client.crt -sha256
```

4.3.3 Broker Configuration (mosquitto.conf)

Configure the Mosquitto broker by updating mosquitto.conf with TLS settings as shown below.

```
allow_anonymous true
protocol mqtt
listener 8883

cafile /path/to/cas.crt
certfile /path/to/server.crt
keyfile /path/to/server.key
tls_version tlsv1.2
```

NOTE
Optionally configure logging and persistence settings as required for user's environment.

4.3.4 Start the Mosquitto Broker with TLS

Run the following command to start the Mosquitto broker with TLS enabled.

```
mosquitto -c mosquitto.conf -p 8883 -v
```

5. FSP Stack Configuration for MQTT Client with DPM

Figure 2 shows the FSP Stack Configuration with the `g_wifi0 (rm_wifi)` module configured to enable MQTT functionality.

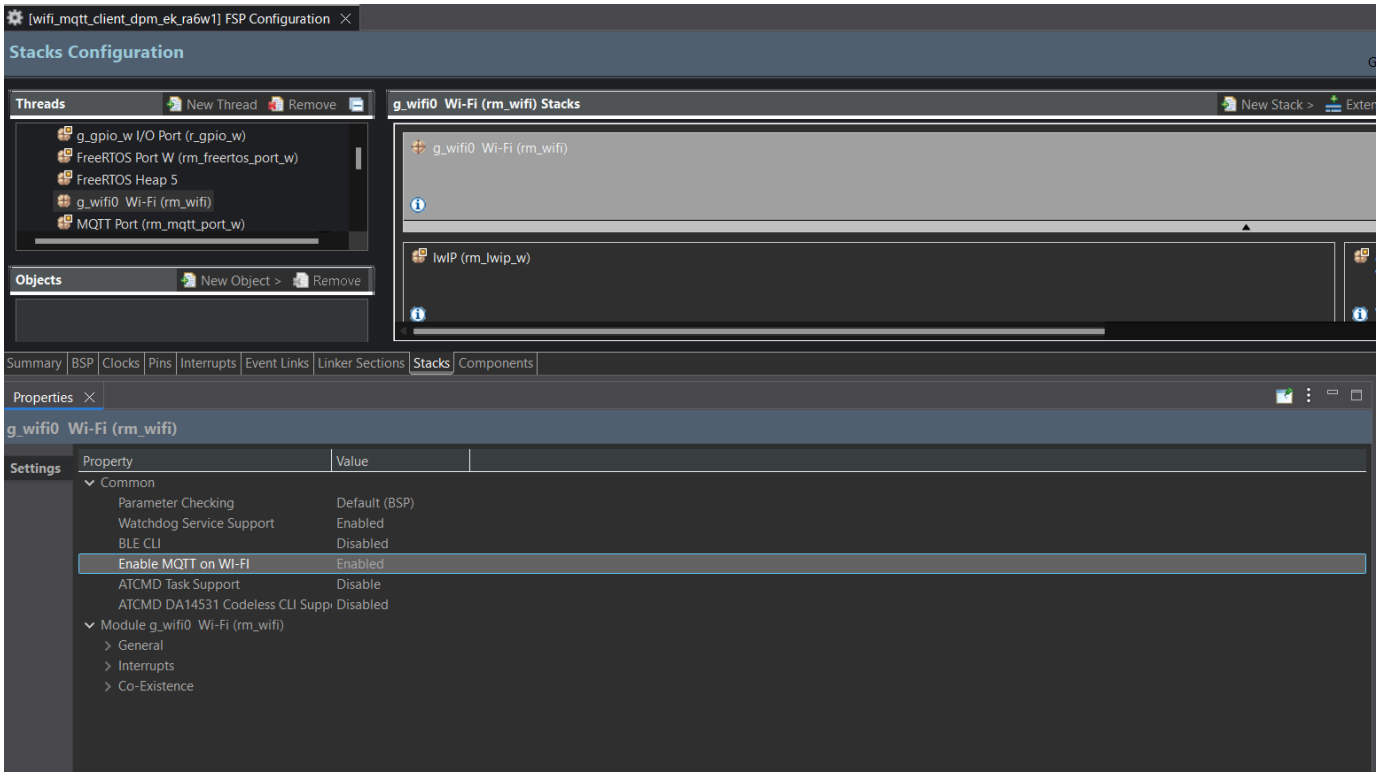


Figure 2. Enable MQTT in Wi-Fi stack

The FSP Stacks tab showing the MQTT Port (`rm_mqtt_port_w`) module and its dependencies.

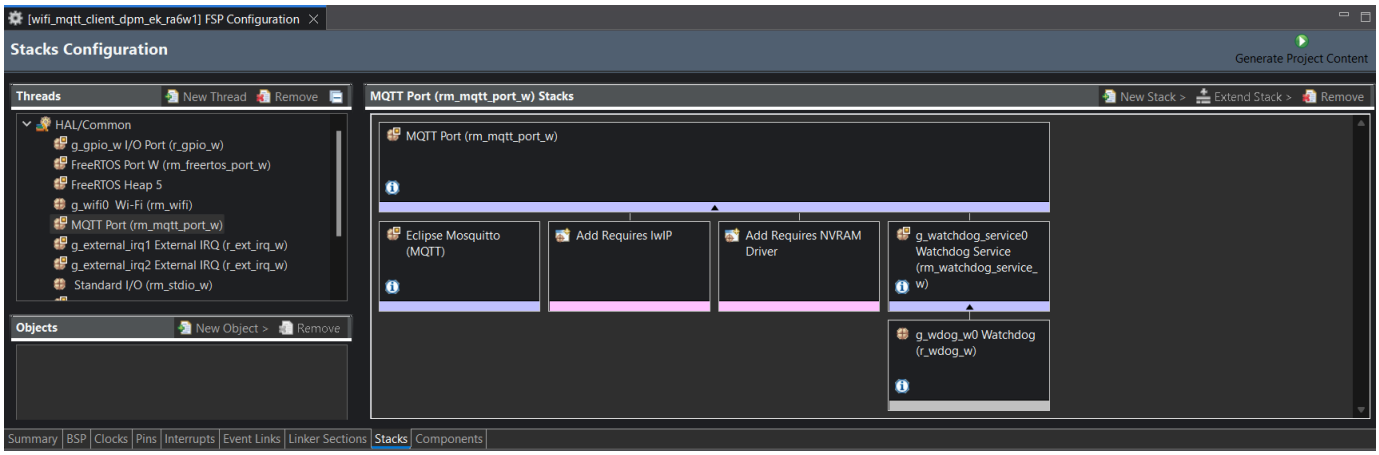


Figure 3. MQTT stack view

The MQTT Port (`rm_mqtt_port_w`) module properties showing the TLS certificate settings: Root CA, Client Certificate, and Private Key used for mutual authentication on port 8883.

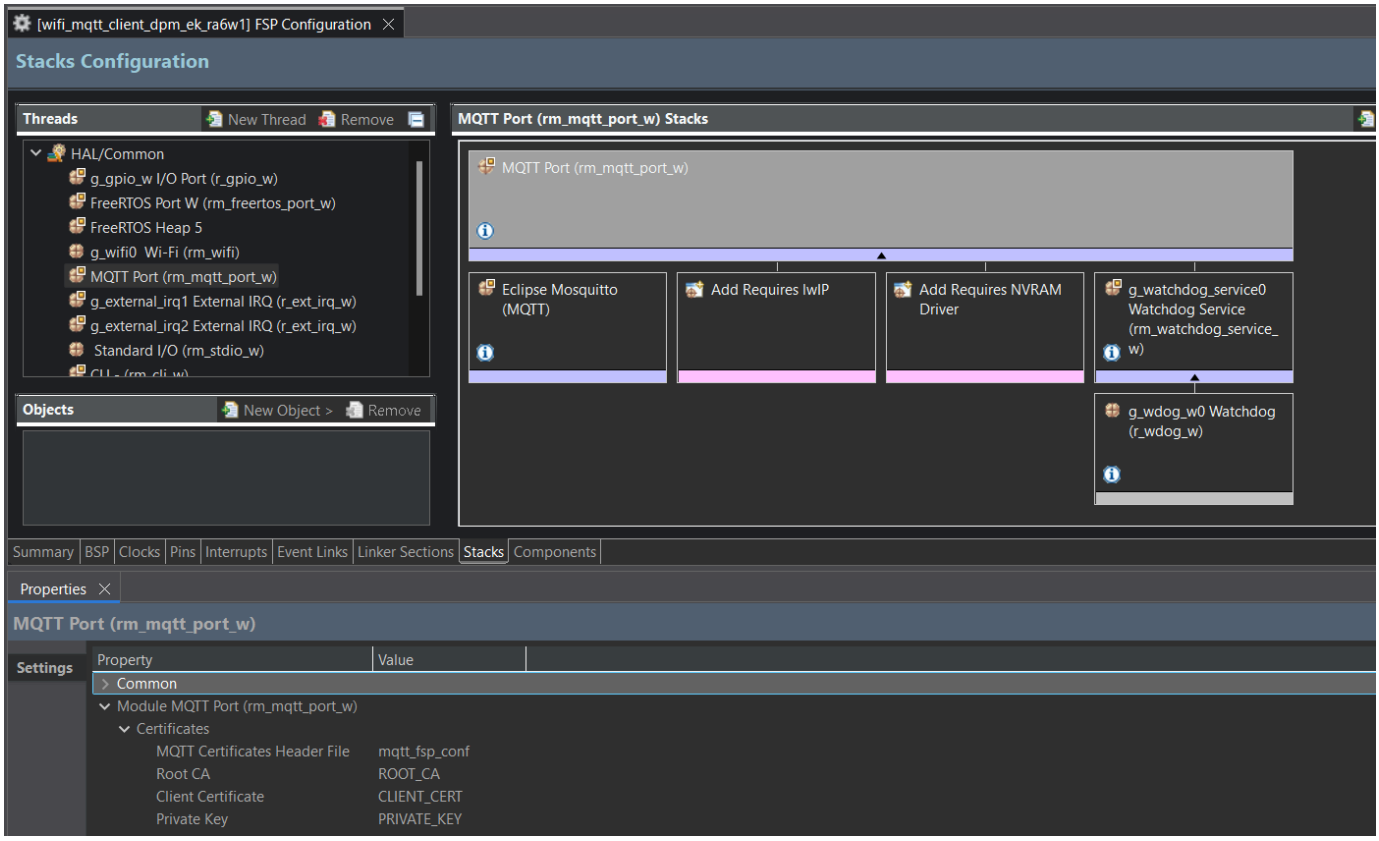


Figure 4. MQTT cert properties

To run the MQTT client with DPM example, complete the following procedures:

- Configure MQTT Parameters (see Section 5.1 Configure MQTT Parameters)
- Flash the firmware (see Section 5.2 Flash Firmware)
- Set up Station mode with DPM (see Section 5.3 Set Up Station Mode with DPM).

5.1 Configure MQTT Parameters

Before building and flashing, update the MQTT parameters in `mqtt.h` to match the user's network setup:

1. Find the host PC's IP address (the PC running the Mosquitto broker).

NOTE
The IPv4 address of the network adapter is connected to the same Access Point.

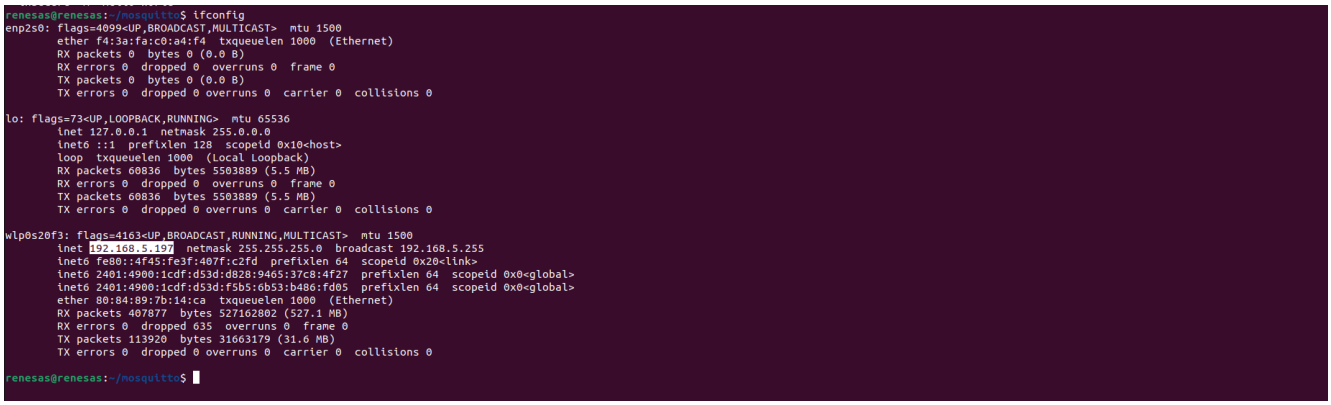


Figure 5. Host PC IPv4 address

2. Update `mqtt.h` file with the broker details.

3. Ensure the certificates in `mqtt_fsp_conf.h` match the ones used by the broker (generated in Section Set Up Mosquitto Broker).
4. Copy the generated `rootCA.crt`, `client.crt`, and `client.key` contents into the ROOT_CA, CLIENT_CERT, and PRIVATE_KEY macros respectively. Both RA6W1 and RA6W2 use the same certificates for testing.

Table 2. MQTT broker configuration in mqtt.h

Macro	Default value	Description
MQTT_BROKER_IP	ip address	IP address of the MQTT broker, for example 192.168.10.2
MQTT_BROKER_PORT	1883	Broker port. Use 1883 for non-TLS o
MQTT_BROKER_PORT_TLS	8883	Broker port. Use 8883 for TLS
MQTT_SUB_TOPIC	rrq61x_sub	Topic the client subscribes to for incoming messages
MQTT_PUB_TOPIC	rrq61x_pub	Topic the client publishes messages to
MQTT_PUBLISH_MSG	Hello from RA6W1	Default message payload for publish operations
MQTT_AUTO_MODE	MQTT_INIT_MAGIC	Enables automatic MQTT connect/publish/subscribe on startup
DPM_WAKEUP_INTERVAL_US	20000	DPM wake-up interval in microseconds between MQTT operations

5.2 Flash Firmware

Generate project content, build, and flash the firmware to the board using e² studio.

1. Open e^2 studio and connect the EK-RA6W1/EK-RA6W2 board to the PC using the USB (J-Link).
2. Import the project into e^2 studio.
3. Open `configuration.xml` and click **Generate Project Content**.
4. Build the project (**Project** > **Build Project**).
5. Go to **Debug Configurations** > **Startup** tab and select the firmware image file.
6. Click **Apply**, then **Debug**.

The firmware is automatically flashed to the device. After completion, the board reboots and begins execution.

5.3 Set Up Station Mode with DPM

Before connecting to Wi-Fi, ensure the Mosquitto broker is running successfully on the host PC (see Section Section 4.3.4 Start the Mosquitto Broker with TLS).

1. Open a serial terminal (Tera Term) with the settings shown in Table 3 and press the reset button on the board. After reset, the Easy Setup wizard starts on the serial console.
2. Configure the Wi-Fi connection by selecting the country code, Station mode, frequency band, SSID, and passphrase through the Easy Setup prompts.
3. Enable DPM when prompted.

```
Enable DPM (Dynamic Power Management)? [Yes/No/Manual/Quit]: y

DPM factors: Defaults? [Yes/No/Quit]: y

==========================================
DPM MODE          : Enable
Keep Alive Time   : 30000ms
User Wakeup Time   : 0ms
TIM Wakeup Count   : 10 dtim
==========================================
DPM CONFIGURATION CONFIRM? [Yes/No/Quit]: y
```

4. Enable SNTP time synchronization.

```
SNTP Client enable? [Yes/No/Quit]: Y
```

```
 SNTP Period time (1 ~ 36 hours)? (default: 36 hours) [Quit] 24
 GMT Timezone +xx:xx|-xx:xx (-12:00 ~ +12:00)? (default: 00:00) [Quit]

 SNTP Server 0 addr ? (default: pool.ntp.org) [Quit]
Input:
 SNTP Server 1 addr ? (default: 1.pool.ntp.org) [Quit]
Input:
 SNTP Server 2 addr ? (default: 2.pool.ntp.org) [Quit]
Input:
==========================================
SNTP Client     : Enable
SNTP Period time: 24 hours
SNTP GMT Timezone: 00:00
SNTP Server addr: pool.ntp.org
SNTP Server addr1: 1.pool.ntp.org
SNTP Server addr2: 2.pool.ntp.org
==========================================
 SNTP Client CONFIRM? [Yes/No/Quit]: Y
```

The device connects to the AP, obtains an IP address, starts the MQTT client, and enters DPM sleep as shown in Figure 6.

```
Wakeup source is 0x4

        ******************************************************
        *              RA6W1-RRQ61001 SDK Information
        * ----------------------------------------------------
        *
        * - CHIP Name       : RA6W1-RRQ61001 (D3095B)
        * - SKU Type        : 0x20, Dual Band (2.4 + 5 GHz) WI-FI 6
        * - CPU Type        : Cortex-M33 (160MHz)
        * - Kernel Version  : FreeRTOS V11.1.0
        * - SFLASH Type     : 8 MB (Renesas AT25SL)
        * - SDK Version     : V8.0.3.0.2
        * - F/W Version     : RA6W1-RRQ61001-a9eef27c7c-1
        * - Boot Index      : 0
        * - F/W Build Time  : May  5 2026 15:27:54
        *
        ******************************************************


System Mode : Station Only (0)
Network init...OK
>>> Renesas RA6W1-RRQ61001 wpa_supplicant 2.10 - Sep/2023
>>> MAC address (sta0) : ec:9f:0d:9f:ff:a1
>>> Start STA mode...
wait

Connection COMPLETE to 60:45:cb:b0:82:c0
         Assigned addr   : 192.168.5.64
                netmask  : 255.255.255.0
                gateway  : 192.168.5.1
         DHCP Server IP  : 192.168.5.1
         Lease Time      : 24h 00m 00s
         Renewal Time    : 20h 00m 00s

>>> SNTP Server: pool.ntp.org (95.216.192.15)

>>> SNTP Time sync : 2026-05-05 14:29:35
>>> MQTT Client connection OK (ra6w1_FFA1)
>>> Start DPM Power-Down !!!
```

Figure 6. MQTT connection and DPM entry

The device is now connected to the broker and operating in DPM mode.

Table 3. Tera Term Serial Port settings

Parameter	Value
Baud rate	115200 bps
Data length	8-bit
Parity	None
Stop bit	1-bit
Flow control	None

5.4 Test Case 1: Device Publishes, Host PC Subscribes

In this test case, the RA6W1/RA6W2 device publishes messages on each RTC DPM wake-up. The host PC subscribes to receive these messages.

1. Ensure the Mosquitto broker is running in a separate console on the host PC (see Section 4.3.4 Start the Mosquitto Broker with TLS).
2. Open a new console on the host PC and subscribe to the device's publish topic.
   a. For non-TLS:

```
mosquitto_sub -h <BROKER_IP> -p 1883 -t rrq61x_pub
```

   b. For TLS:

```
mosquitto_sub -h <BROKER_IP>   -p 8883 --cafile cas.crt --cert
/home/renesas/mosquitto/server.crt --key /home/renesas/mosquitto/server.key --tls-version
tlsv1.2 --insecure -t rrq61x_pub
```

3. Observe the device serial terminal. On each RTC wake-up, the device logs are similar to Figure 7.

```
>>> Start DPM Power-Down !!!

Wakeup source is 0x82
>>> TIM STATUS: 0x00001000
>>> TIM : FAST
 Waking from RTC DPM
(Tx: Len=16,Topic=rrq61x_pub,Msg_ID=1
< Mqtt Pub EnQ : SUCCESS >>

[1;36m
         < USER WAKE UP : USR_TMR >

>>> Start DPM Power-Down !!!

Wakeup source is 0x82
>>> TIM STATUS: 0x00001000
>>> TIM : FAST
 Waking from RTC DPM
(Tx: Len=16,Topic=rrq61x_pub,Msg_ID=1
< Mqtt Pub EnQ : SUCCESS >>

[1;36m
         < USER WAKE UP : USR_TMR >

>>> Start DPM Power-Down !!!
```

Figure 7. Device publish log (RTC wake-up)

4. Observe the host PC subscriber console. Published messages appear as displayed in Figure 8.

```
mosquitto_sub -h <BROKER_IP> -p 8883 --cafile cas.crt --cert /home/renesas/mosquitto/server.crt
--key /home/renesas/mosquitto/server.key --tls-version tlsv1.2 --insecure -t rrq61x_pub
```

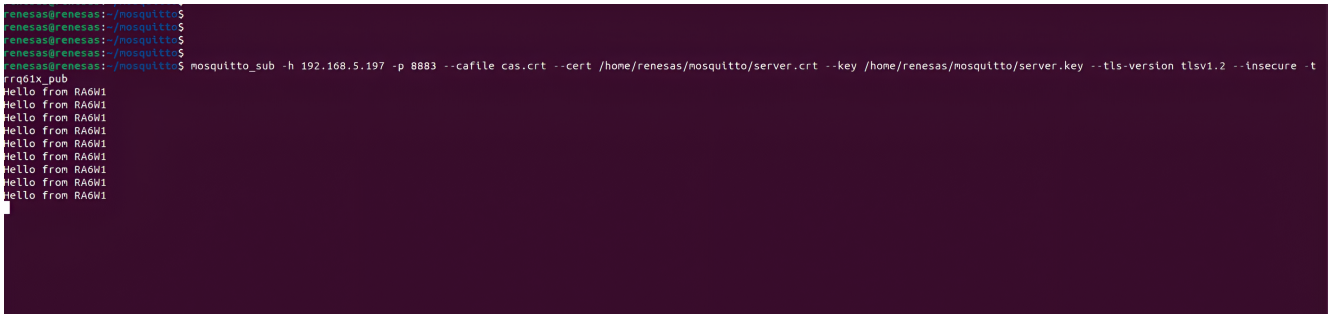


Figure 8. Host PC subscriber receiving messages

The device continues publishing on each periodic RTC wake-up with incrementing Msg_ID values. After each publishing, the device returns to DPM sleep.

5.5 Test Case 2: Host PC Publishes, Device Subscribe

In this test case, the host PC publishes a message to the device's subscribe topic. The device wakes from DPM when the AP buffers the incoming data and signals.

1. Ensure the Mosquitto broker is running, and the device is in DPM mode (see Section 4.3.4 Start the Mosquitto Broker with TLS).
2. Open a new console on the host PC and publish a message to the device.
   a. For non-TLS

```
mosquitto_pub -h <BROKER_IP> -p 1883 -t rrq61x_sub -m "Hello World"
```

   b. For TLS:

```
mosquitto_pub -h <BROKER_IP> -p 8883 --cafile cas.crt --cert
/home/renesas/mosquitto/server.crt --key /home/renesas/mosquitto/server.key --tls-version
tlsv1.2 -t rrq61x_sub --insecure -m "Hello World"
```



Figure 9. Host PC publishes messages

3. Observe the device serial terminal. When a message is published, the AP signals the TIM UC and the device wakes immediately to receive it, as shown in Figure 10.

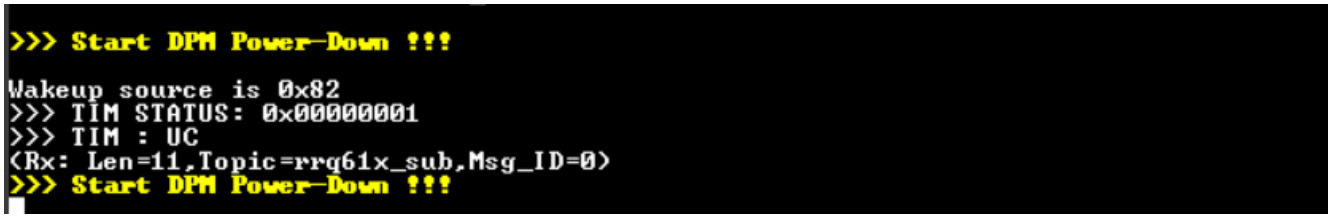


Figure 10. Device receive log

The Len = 11 corresponds to the 11-byte payload Hello World. After receiving it, the device returns to DPM sleep.

6. Revision History

Revision	Date	Description
1.01	July 21, 2026	Updated the Software requirements section.
1.00	May 20, 2026	Initial release.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01)

CorporateHeadquarters

TOYOSU FORESIA, 3-2-24 Toyosu
Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/

© 2026 Renesas Electronics Corporation. All rights reserved.