Application Note AWS_IoT_MQTT Example

Version 1.0.0

For more information, visit our website at http://www.wiznet.io

Contents

Figures

Tables

1 Introduction

This Application Note covers the implementation of subscribing and publishing via MQTT using AWS IoT on WIZnet's TOE Chip.

2 Github Link

https://github.com/WIZnet-ioNIC/WIZnet-PICO-AWS-C/tree/main/examples/aws_iot_mqtt

3 Applicable products

Raspberry Pi Pico & WIZnet Ethernet HAT

W5100S-EVB-Pico

W5500-EVB-Pico

W55RP20-EVB-Pico

W5100S-EVB-Pico2

W5500-EVB-Pico2

4 How to Test AWS IoT MQTT Example

4.1 Step 1: Prepare software

The following serial terminal programs are required for AWS IoT MQTT example test, download and install from below links.

- **Tera Term**

4.2 Step 2: Prepare hardware

If you are using W5100S-EVB-Pico, W5500-EVB-Pico, W55RP20-EVB-Pico, W5100S-EVB-Pico2 or W5500-EVB-Pico2, you can skip '1. Combine...'

1. Combine WIZnet Ethernet HAT with Raspberry Pi Pico.
2. Connect ethernet cable to WIZnet Ethernet HAT, W5100S-EVB-Pico, W5500-EVB-Pico, W55RP20-EVB-Pico, W5100S-EVB-Pico2 or W5500-EVB-Pico2 ethernet port.
3. Connect Raspberry Pi Pico, W5100S-EVB-Pico or W5500-EVB-Pico to desktop or laptop using 5 pin micro USB cable. W55RP20-EVB-Pico, W5100S-EVB-Pico2 or W5500-EVB-Pico2 require a USB Type-C cable.

4.3 Step 3: Setup AWS IoT

In order to connect to AWS IoT through MQTT, the development environment must be configured to use AWS IoT.

This example was tested by configuring AWS IoT Core. Please refer to the 'Create AWS IoT resources' section of document below and configure accordingly.

- **Create AWS IoT resources**
- **What is AWS IoT?**

4.4 Step 4: Setup AWS IoT MQTT Example

To test the AWS IoT MQTT example, minor settings shall be done in code.

1. Setup SPI port and pin in 'w5x00_spi.h' in 'WIZnet-PICO-AWS-C/port/ioLibrary_Driver/' directory.

Setup the SPI interface you use.

- If you use the W5100S-EVB-Pico, W5500-EVB-Pico, W5100S-EVB-Pico2 or W5500-EVB-Pico2,

```
/* SPI */
#define SPI_PORT spi0

#define PIN_SCK 18
#define PIN_MOSI 19
#define PIN_MISO 16
#define PIN_CS 17
#define PIN_RST 20
```

- If you want to test with the AWS IoT MQTT example using SPI DMA, uncomment USE_SPI_DMA.

```
/* Use SPI DMA */
//#define USE_SPI_DMA // if you want to use SPI DMA, uncomment.
```

- If you use the W55RP20-EVB-Pico,

```
/* SPI */
#define USE_SPI_PIO

#define PIN_SCK 21
#define PIN_MOSI 23
#define PIN_MISO 22
#define PIN_CS 20
#define PIN_RST 25
```

2. Setup network configuration such as IP in ' aws_iot_mqtt.c', which is the AWS IoT MQTT example in 'WIZnet-PICO-AWS-C/examples/aws_iot_mqtt/' directory.

- Setup IP, other network settings to suit your network environment and whether to use DHCP.

```
/* Network */
static wiz_NetInfo g_net_info =
    {
        .mac = {0x00, 0x08, 0xDC, 0x12, 0x34, 0x56}, // MAC address
        .ip = {192, 168, 11, 2},                     // IP address
        .sn = {255, 255, 255, 0},                    // Subnet Mask
        .gw = {192, 168, 11, 1},                     // Gateway
        .dns = {8, 8, 8, 8},                         // DNS server
        .dhcp = NETINFO_DHCP                // DHCP enable/disable
};
```

3. Setup AWS IoT configuration.

- MQTT_DOMAIN should be setup as AWS IoT data endpoint, and MQTT_USERNAME and MQTT_PASSWORD do not need to be setup. Setup MQTT_CLIENT_ID same as thing created during AWS IoT Core setup.

```
#define MQTT_DOMAIN "account-specific-prefix-ats.iot.ap-northeast-
                     2.amazonaws.com"
#define MQTT_PUB_TOPIC "$aws/things/my_rp2040_thing/shadow/update"
#define MQTT_SUB_TOPIC "$aws/things/my_rp2040_thing
                        /shadow/update/accepted"
#define MQTT_USERNAME NULL
#define MQTT_PASSWORD NULL
#define MQTT_CLIENT_ID "my_rp2040_thing"
```

4. Setup device certificate and key.

- You must enter the root certificate, client certificate and private key that were downloaded in Step 3.
- Root certificate uses the RSA 2048 bit key, Amazon Root CA 1, and does not use the public key.
- Device certificate and key can be set in 'mqtt_certificate.h' in 'WIZnet-PICO-AWS-C/examples/aws_iot_mqtt/' directory.

```
uint8_t mqtt_root_ca[] =
"-----BEGIN CERTIFICATE-----\r\n"
"...\r\n"
"-----END CERTIFICATE-----\r\n";

uint8_t mqtt_client_cert[] =
"-----BEGIN CERTIFICATE-----\r\n"
"...\r\n"
```

```
"-----END CERTIFICATE-----\r\n";

uint8_t mqtt_private_key[] =
"-----BEGIN RSA PRIVATE KEY-----\r\n"
"...\r\n"
"-----END RSA PRIVATE KEY-----\r\n";
```

4.5 Step 5: Build

1. After completing the AWS IoT MQTT example configuration, click 'build' in the status bar at the bottom of Visual Studio Code or press the 'F7' button on the keyboard to build.
2. When the build is completed, 'aws_iot_mqtt.uf2' is generated in 'WIZnet-PICO-AWS-C/build/examples/aws_iot_mqtt/' directory.

4.6 Step 6: Upload and Run

1. While pressing the BOOTSEL button of Raspberry Pi Pico, W5100S-EVB-Pico, W5500-EVB-Pico, W55RP20-EVB-Pico, W5100S-EVB-Pico2 or W5500-EVB-Pico2 power on the board, the USB mass storage 'RPI-RP2' is automatically mounted.

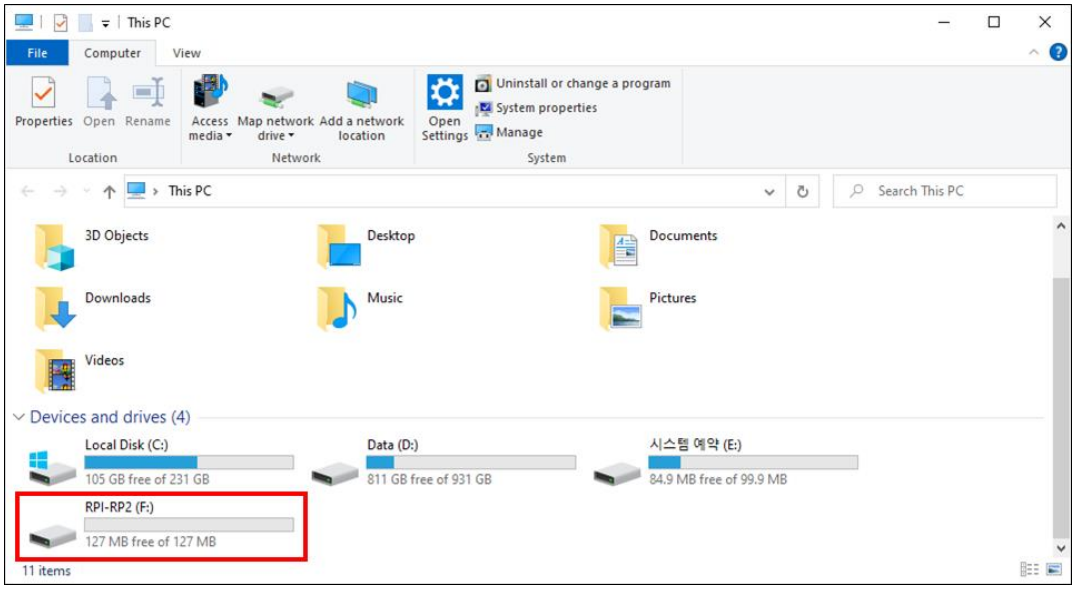

Figure 1. USB mass storage

2. Drag and drop ' aws_iot_mqtt.uf2' onto the USB mass storage device 'RPI-RP2'.

3. Connect to the serial COM port of Raspberry Pi Pico, W5100S-EVB-Pico, W5500-EVB-Pico, W55RP20-EVB-Pico, W5100S-EVB-Pico2 or W5500-EVB-Pico2 with Tera Term.

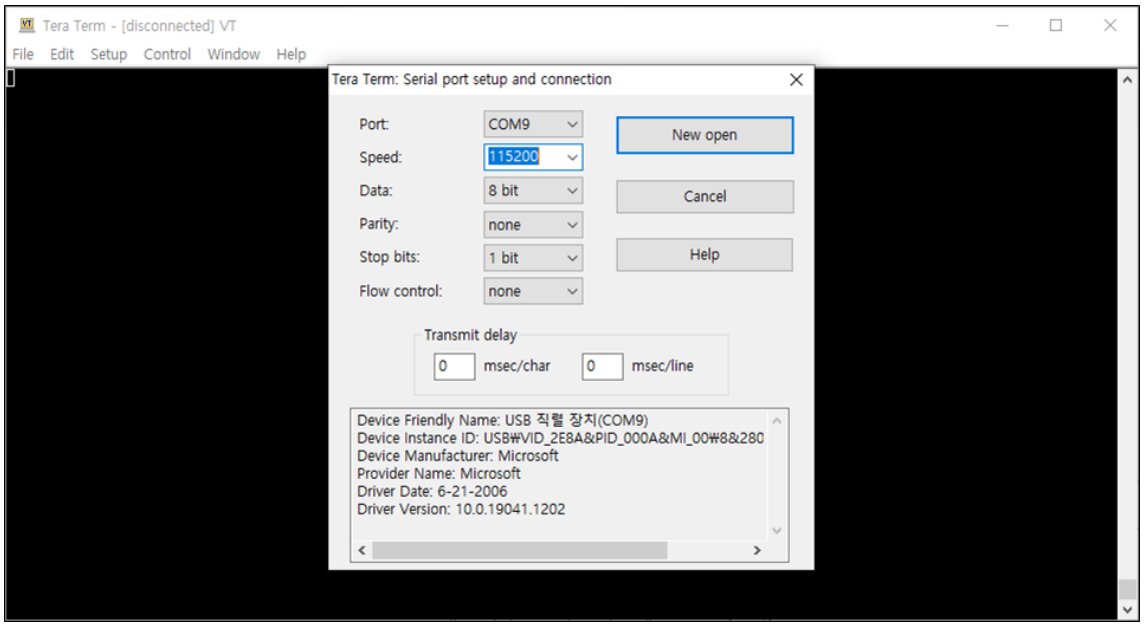

Figure 2. Tera Term

4. Reset your board.
5. If the AWS IoT MQTT example works normally on Raspberry Pi Pico, W5100S-EVB-Pico, W5500-EVB-Pico, W55RP20-EVB-Pico, W5100S-EVB-Pico2 or W5500-EVB-Pico2, you can see the network information of Raspberry Pi Pico, W5100S-EVB-Pico, W5500-EVB-Pico, W55RP20-EVB-Pico, W5100S-EVB-Pico2 or W5500-EVB-Pico2, connecting to the AWS IoT and publishing the message.

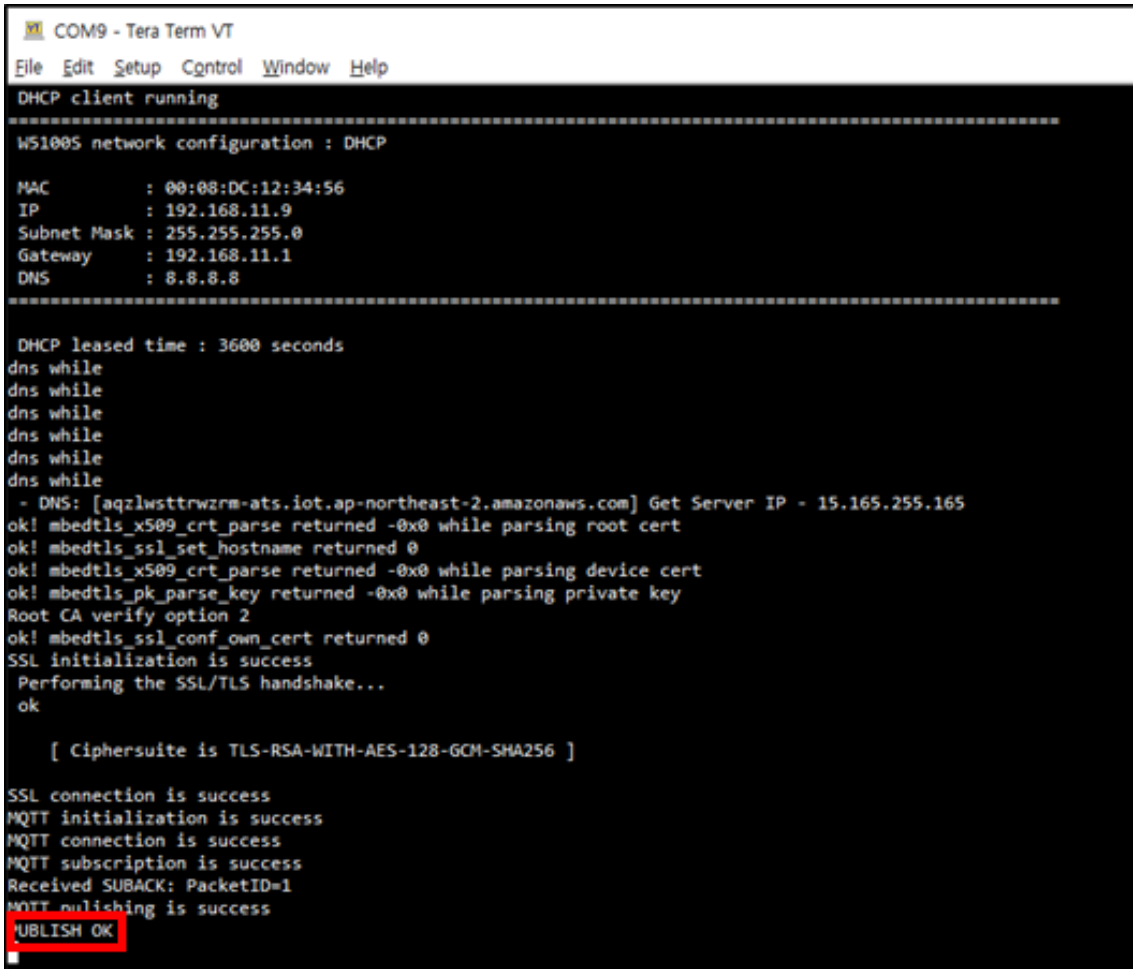

Figure 4. Network Info and connect to AWS IoT and publishing message

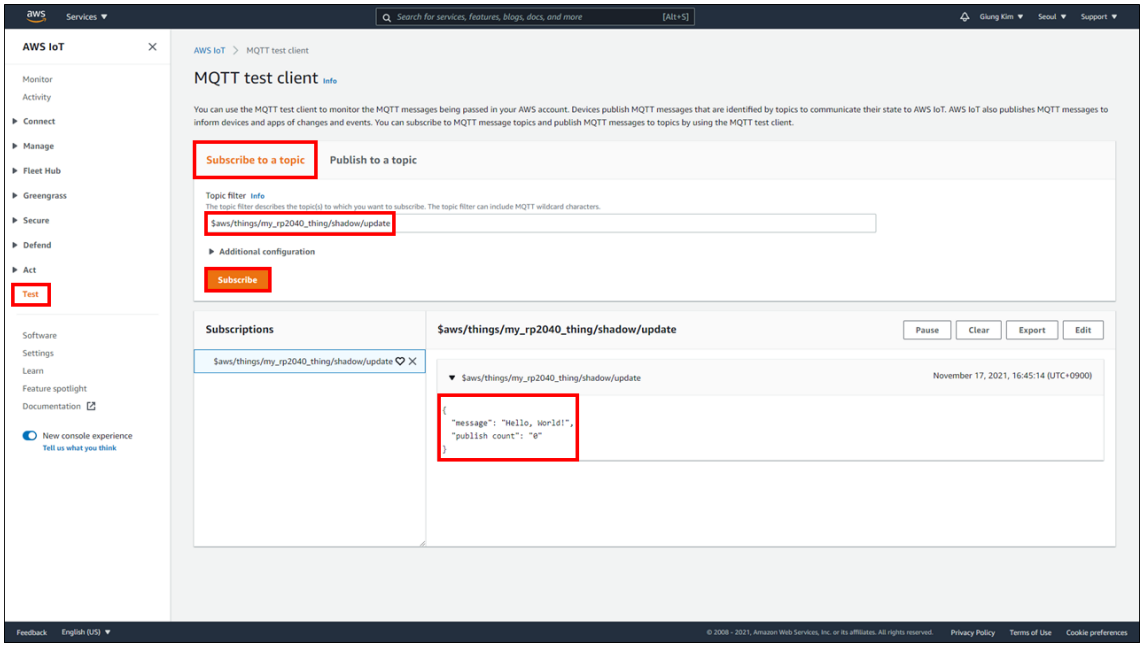

Figure 3. Subscribe to publish topic and receive publish message through test function

6. If you publish the message through the test function in AWS IoT Core to the subscribe topic was configured in Step 4, you can see that the Raspberry Pi Pico, W5100S-EVB-Pico, W5500-EVB-Pico, W55RP20-EVB-Pico, W5100S-EVB-Pico2 or W5500-EVB-Pico2 receive the message about the subscribe topic.

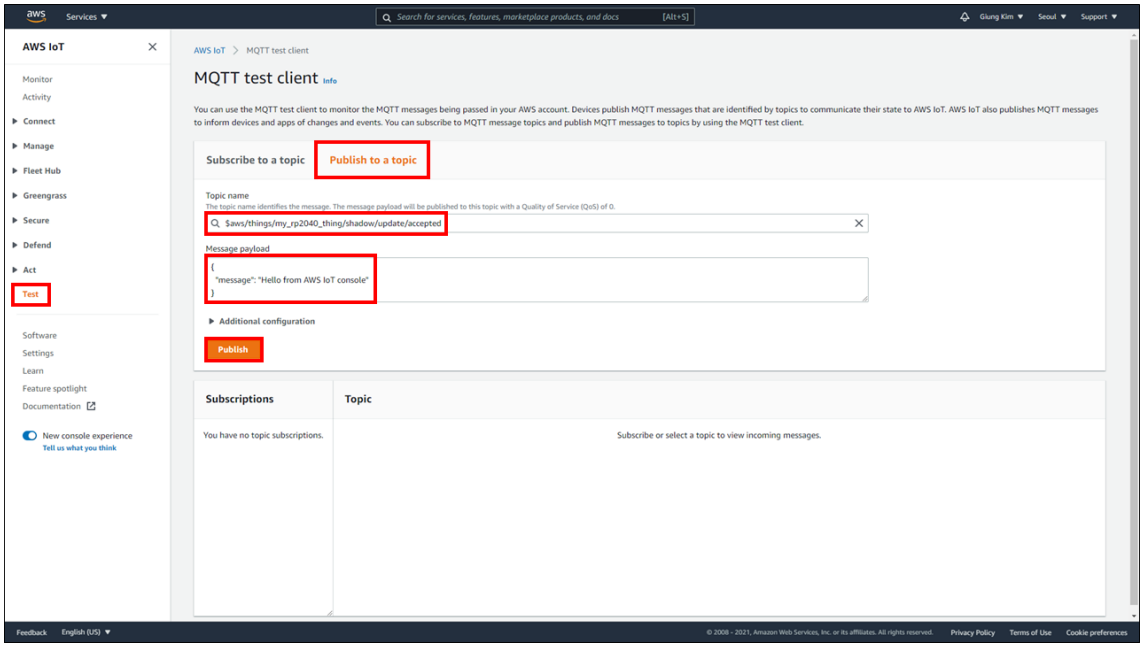

Figure 5. Publish message through test function

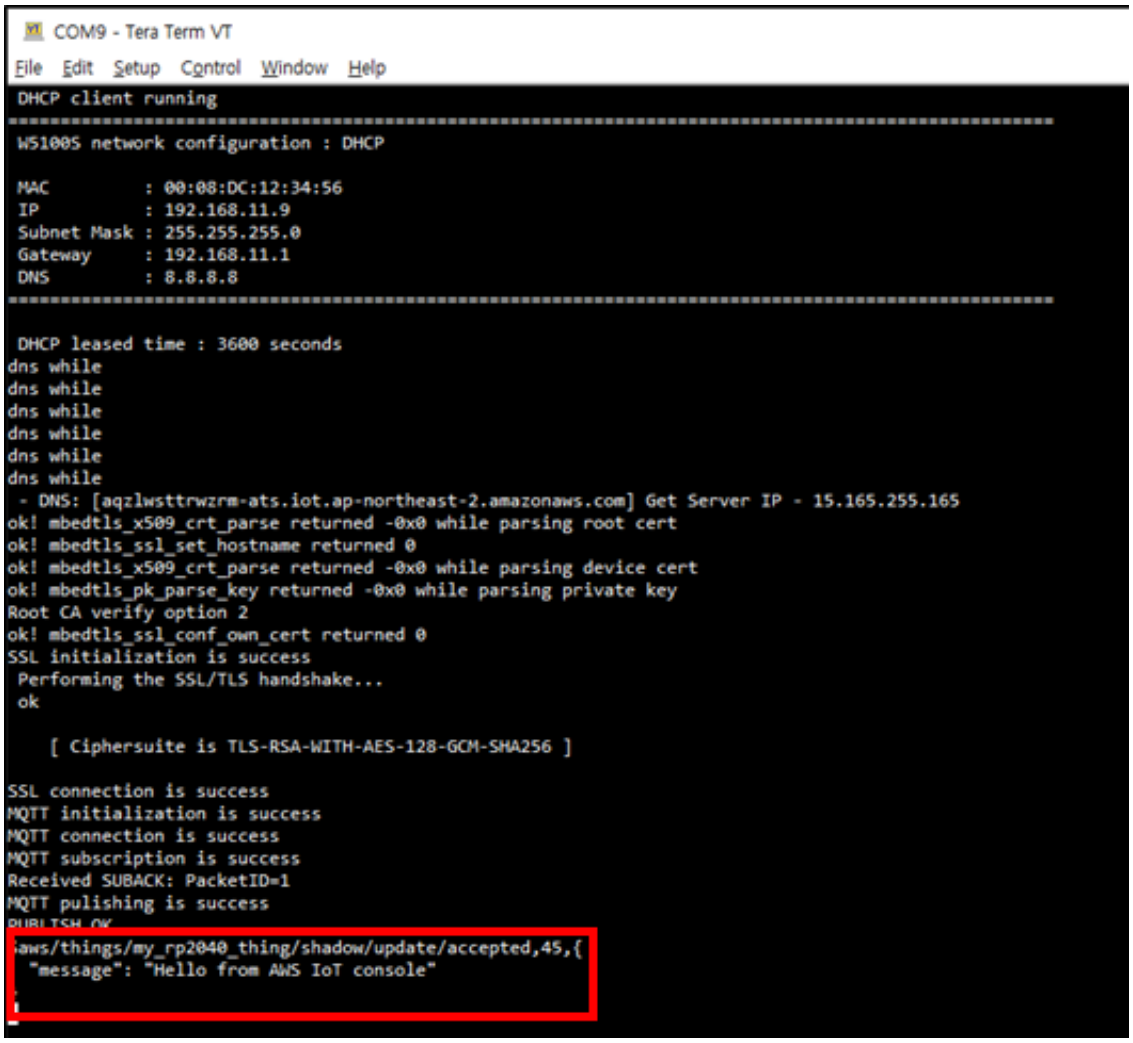

Figure 6. Receive message about subscribe topic

Revision history

Version	Date	Descriptions
Ver. 1.0.0	Dec, 2024	Initial release.

Table 1. Revision history