Application Note
MQTT_Publish 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 MQTT Publish on WIZnet's TOE Chip.

2 Github Link

https://github.com/WIZnet-ioNIC/WIZnet-PICO-C/tree/main/examples/mqtt/publish

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 MQTT Publish Example

4.1 Step 1: Prepare software

The following serial terminal program and MQTT broker are required for MQTT Publish example test, download and install from below links.

- **Tera Term**
- **Mosquitto**

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 MQTT Publish Example

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

1. Setup SPI port and pin in 'w5x00_spi.h' in 'WIZnet-PICO-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 MQTT Publish 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 ' w5x00_mqtt_publish.c', which is the MQTT Publish example in 'WIZnet-PICO-C/examples/mqtt/publish/' directory.

- Setup IP, other network settings to suit your network environment.

```
/* 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_STATIC                  // DHCP enable/disable
};
```

3. Setup MQTT configuration in w5x00_mqtt_publish.c' in 'WIZnet-PICO-C/examples/mqtt/publish/' directory.

- In the MQTT configuration, the broker IP is the IP of your desktop or laptop where broker will be created.

```
/* Port */
#define PORT_MQTT 1883

/* MQTT */
#define MQTT_CLIENT_ID "rpi-pico"
#define MQTT_USERNAME "wiznet"
#define MQTT_PASSWORD "0123456789"
#define MQTT_PUBLISH_TOPIC "publish_topic"
#define MQTT_PUBLISH_PAYLOAD "Hello, World!"
#define MQTT_KEEP_ALIVE 60 // 60 milliseconds

static uint8_t g_mqtt_broker_ip[4] = {192, 168, 11, 3};
```

4.4 Step 4: Build

1. After completing the MQTT Publish 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, 'w5x00_mqtt_publish.uf2' is generated in 'WIZnet-PICO-C/build/examples/mqtt/publish' directory.

4.5 Step 5: 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.

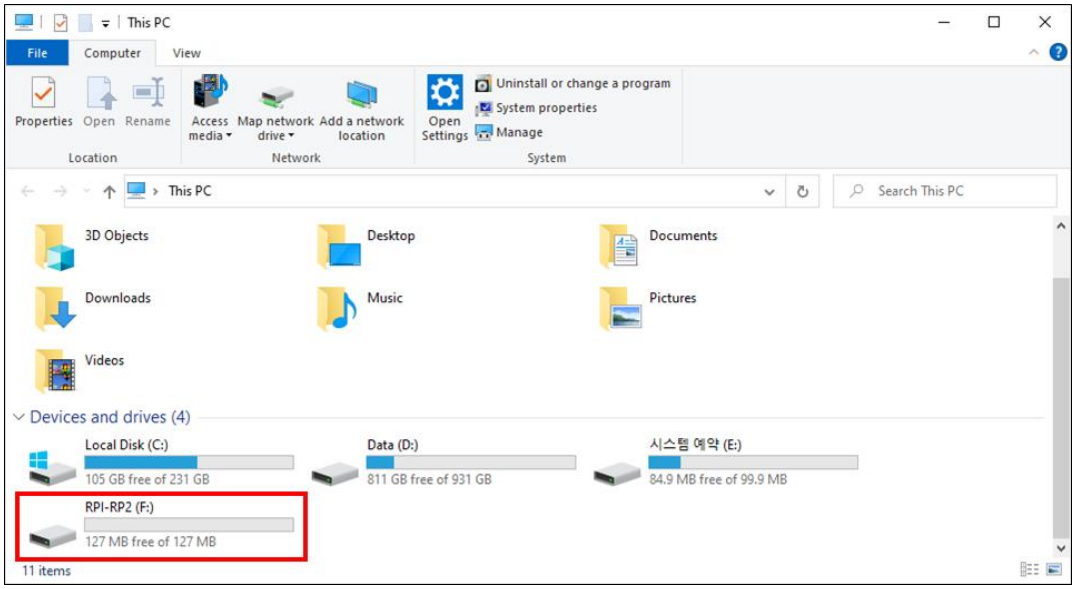

Figure 1. USB mass storage

2. Drag and drop ' w5x00_mqtt_publish.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.

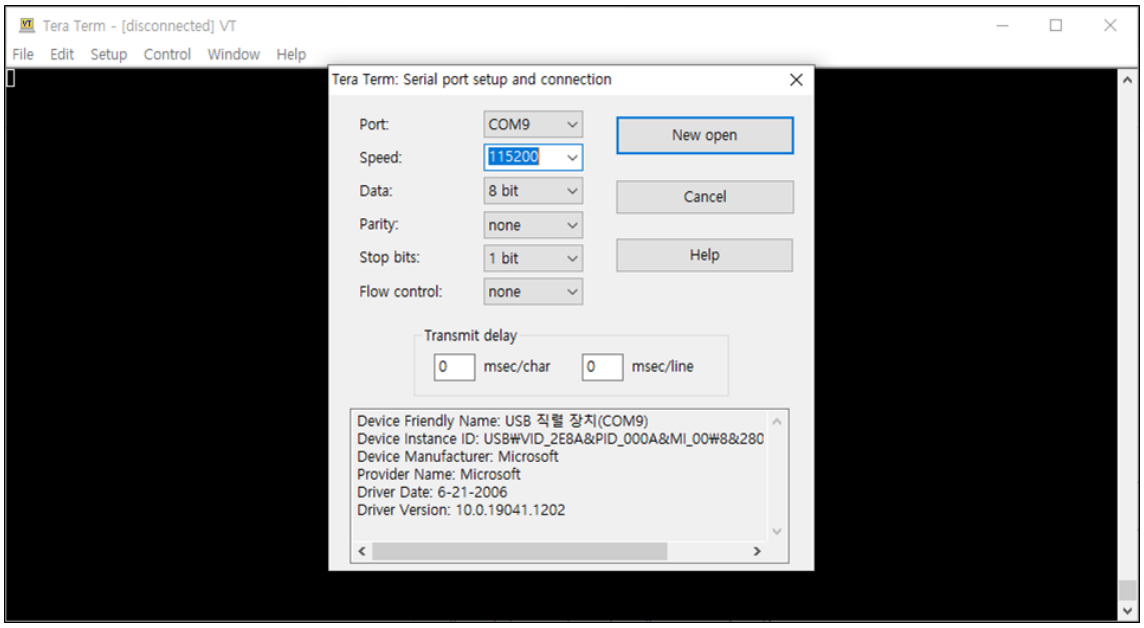

Figure 2. Tera Term

4. Run Mosquitto to be used as the broker.

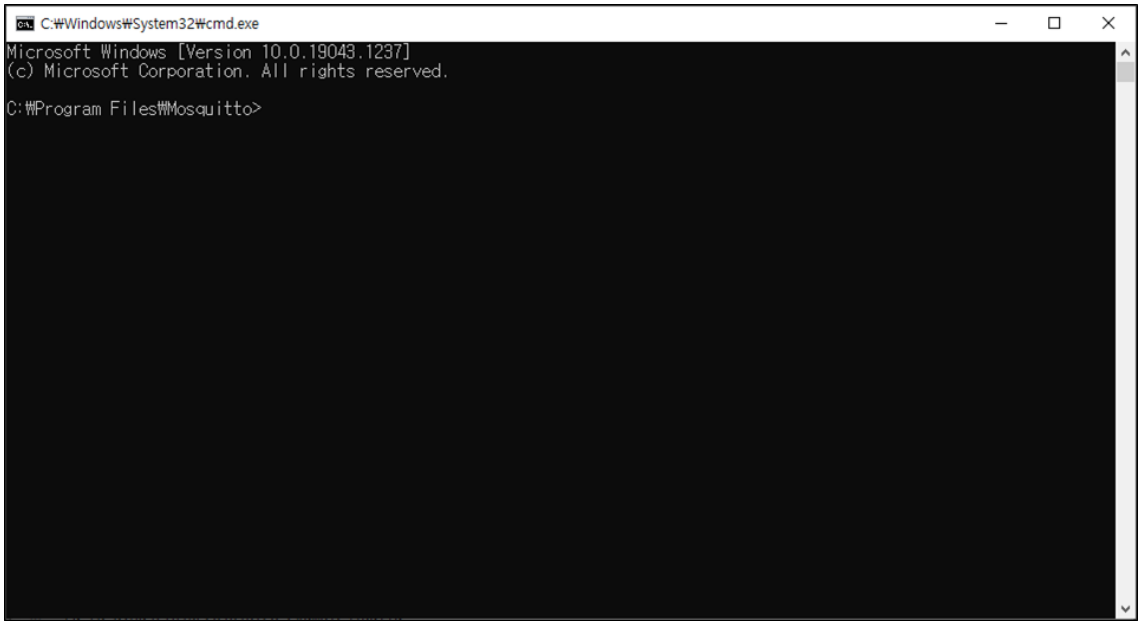

Figure 3. Run Mosquitto

5. Create broker using Mosquitto by executing the following command. If the broker is created normally, the broker's IP is the current IP of your desktop or laptop, and the port is 1883 by default.

```
mosquitto -c mosquitto.conf -v
```

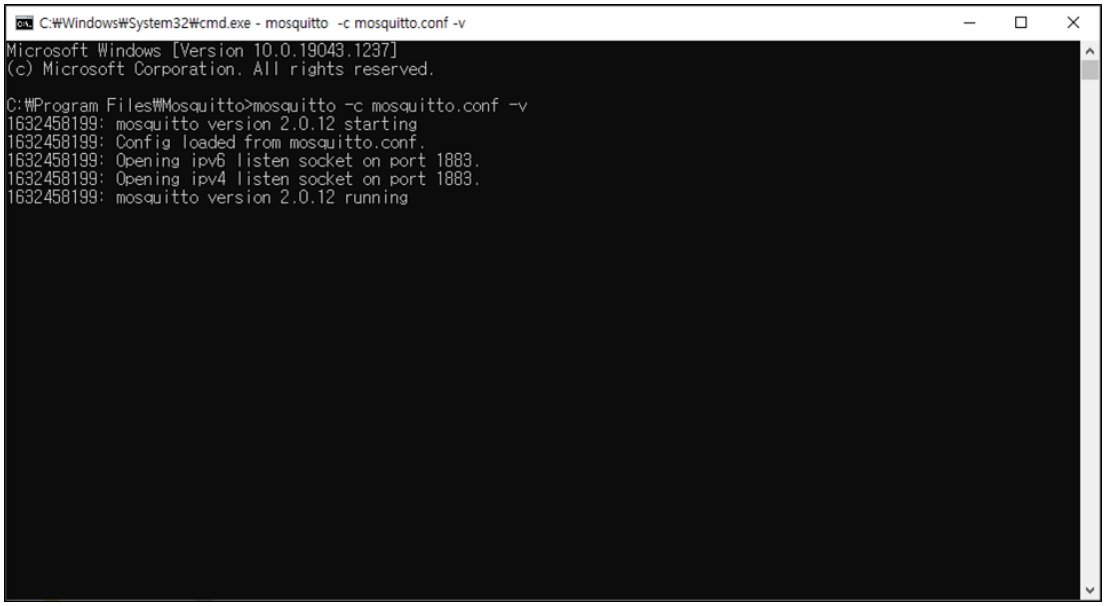

Figure 4. Create MQTT broker using mosquitto

6. Reset your board.

7. If the MQTT Publish 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 broker and publishing the message.

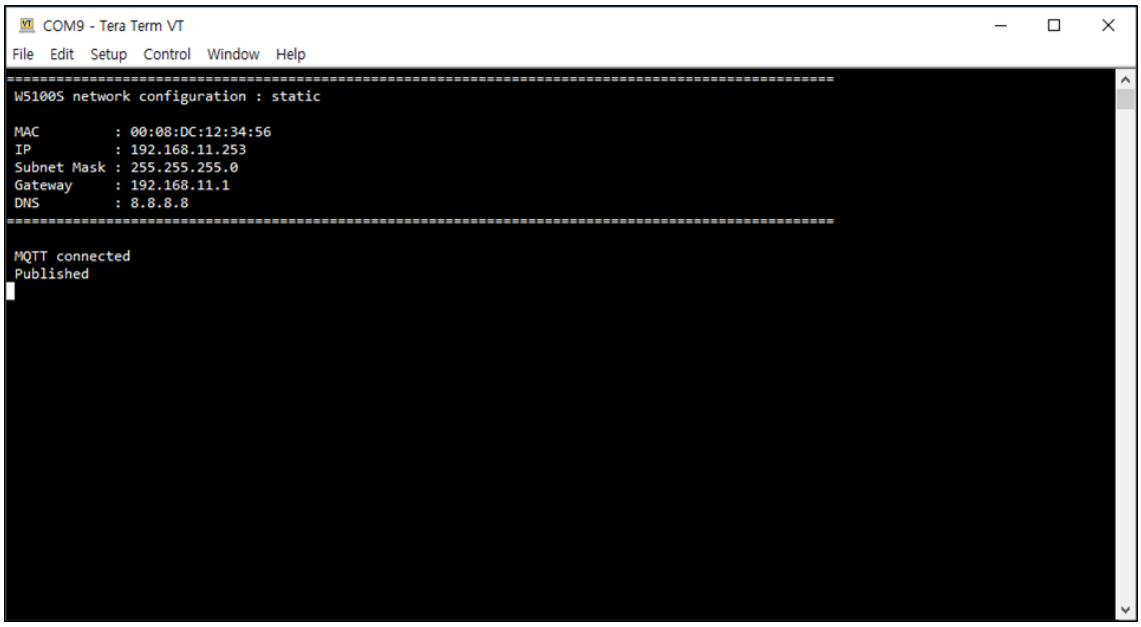

Figure 5. Connect to broker and publish message 1

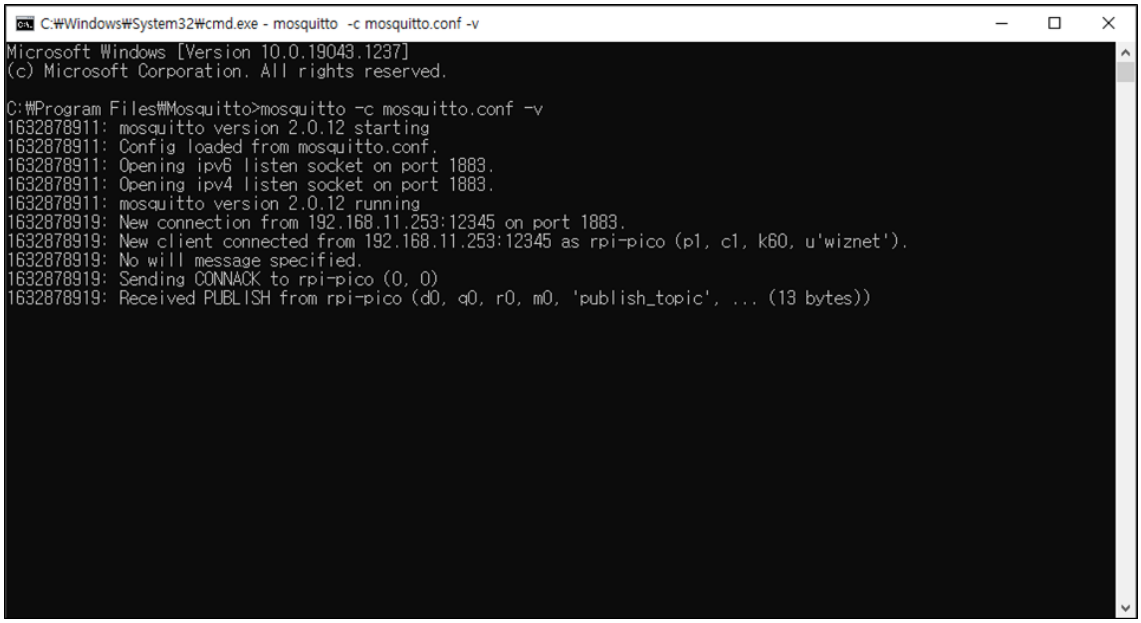

Figure 6. Connect to broker and publish message 2

4.6 Appendix

In Mosquitto versions earlier than 2.0 the default is to allow clients to connect without authentication. In 2.0 and up, you must choose your authentication options explicitly before clients can connect. Therefore, if you are using version 2.0 or later, refer to following link to setup 'mosquitto.conf' in the directory where Mosquitto is installed.

- **Authentication Methods**

Revision history

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

Table 1. Revision history