**Application Note**

**MQTT\_Publish Example**

**Version 1.1.0**



© 2024 WIZnet Co., Ltd. All Rights Reserved.

For more information, visit our website at [http://www.wiznet.io](http://www.wiznet.io/)

Contents

[1 Introduction 3](#_Toc195083780)

[2 Github Link 3](#_Toc195083781)

[3 Applicable products 3](#_Toc195083782)

[4 How to Test MQTT Publish Example 3](#_Toc195083783)

[4.1 Step 1: Prepare software 3](#_Toc195083784)

[4.2 Step 2: Prepare hardware 3](#_Toc195083785)

[4.3 Step 3: Setup MQTT Publish Example 4](#_Toc195083786)

[4.4 Step 4: Build 6](#_Toc195083787)

[4.5 Step 5: Upload and Run 7](#_Toc195083788)

[4.6 Appendix 10](#_Toc195083789)

[Revision history 11](#_Toc195083790)

**Figures**

[Figure 1. USB mass storage 6](#_Toc184045396)

[Figure 2. Tera Term 6](#_Toc184045397)

[Figure 3. Run Mosquitto 7](#_Toc184045398)

[Figure 4. Create MQTT broker using mosquitto 7](#_Toc184045399)

[Figure 5. Connect to broker and publish message 1 8](#_Toc184045400)

[Figure 6. Connect to broker and publish message 2 8](#_Toc184045401)

**Tables**

Table 1. Revision history 10

1. Introduction

This Application Note covers the implementation of MQTT Publish on WIZnet's TOE Chip.

1. Github Link

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

1. Applicable products

[Raspberry Pi Pico & WIZnet Ethernet HAT](https://docs.wiznet.io/Product/Open-Source-Hardware/wiznet_ethernet_hat)

[W5100S-EVB-Pico](https://docs.wiznet.io/Product/iEthernet/W5100S/w5100s-evb-pico)

[W5500-EVB-Pico](https://docs.wiznet.io/Product/iEthernet/W5500/w5500-evb-pico)

[W55RP20-EVB-Pico](https://docs.wiznet.io/Product/ioNIC/W55RP20/w55rp20-evb-pico)

[W5100S-EVB-Pico2](https://docs.wiznet.io/Product/iEthernet/W5100S/w5100s-evb-pico2)

[W5500-EVB-Pico2](https://docs.wiznet.io/Product/iEthernet/W5500/w5500-evb-pico2)

[W6100-EVB-Pico2](https://docs.wiznet.io/Product/iEthernet/W6100/w6100-evb-pico2)[W6300-EVB-Pico2](https://docs.wiznet.io/Product/iEthernet/W6300/w6300-evb-pico2)

1. How to Test MQTT Publish Example
   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](https://osdn.net/projects/ttssh2/releases/)**
* **[Mosquitto](https://mosquitto.org/download/)**
  1. Step 2: Prepare hardware

If you are using W5100S-EVB-Pico, W5500-EVB-Pico, W55RP20-EVB-Pico, W5100S-EVB-Pico2 W5500-EVB-Pico2, W6100-EVB-Pico2 or W6300-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, W5500-EVB-Pico2, W6100-EVB-Pico2 or W6300-EVB-Pico2 ethernet port.
3. Connect Raspberry Pi Pico, W5100S-EVB-Pico , W5500-EVB-Pico or W6100-EVB-Pico2 to desktop or laptop using 5 pin micro USB cable. W55RP20-EVB-Pico, W5100S-EVB-Pico2 , W5500-EVB-Pico2 or W6300-EVB-Pico2 require a USB Type-C cable.  
   1. 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 ''wizchip\_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 |

* For W6300-EVB-PICO or W6300-EVB-PICO2:
* If you are using the W6300-EVB-PICO or W6300-EVB-PICO2, use the following pinout and SPI clock divider configuration:

|  |
| --- |
| /\* SPI \*/  #elif (DEVICE\_BOARD\_NAME == W6300\_EVB\_PICO2)  // SPI SCLK SPEED = 66.5MHz / (PIO\_CLOCK\_DIV\_MAJOR + (PIO\_CLOCK\_DIV\_MINOR / 256))  #define PIO\_CLOCK\_DIV\_MAJOR 1  #define PIO\_CLOCK\_DIV\_MINOR 130  #define USE\_PIO  #define PIO\_IRQ\_PIN 15  #define PIO\_SPI\_SCK\_PIN 17  #define PIO\_SPI\_DATA\_IO0\_PIN 18  #define PIO\_SPI\_DATA\_IO1\_PIN 19  #define PIO\_SPI\_DATA\_IO2\_PIN 20  #define PIO\_SPI\_DATA\_IO3\_PIN 21  #define PIN\_CS 16  #define PIN\_RST 22 |

1. Setup network configuration such as IP in ' wizchip\_mqtt\_publish.c', which is the MQTT Publish example in 'WIZnet-PICO-C/examples/mqtt/publish/’ directory.
2. 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  #if \_WIZCHIP\_ > W5500  .lla = {0xfe, 0x80, 0x00, 0x00,  0x00, 0x00, 0x00, 0x00,  0x02, 0x08, 0xdc, 0xff,  0xfe, 0x57, 0x57, 0x25}, // Link Local Address  .gua = {0x00, 0x00, 0x00, 0x00,  0x00, 0x00, 0x00, 0x00,  0x00, 0x00, 0x00, 0x00,  0x00, 0x00, 0x00, 0x00}, // Global Unicast Address  .sn6 = {0xff, 0xff, 0xff, 0xff,  0xff, 0xff, 0xff, 0xff,  0x00, 0x00, 0x00, 0x00,  0x00, 0x00, 0x00, 0x00}, // IPv6 Prefix  .gw6 = {0x00, 0x00, 0x00, 0x00,  0x00, 0x00, 0x00, 0x00,  0x00, 0x00, 0x00, 0x00,  0x00, 0x00, 0x00, 0x00}, // Gateway IPv6 Address  .dns6 = {0x20, 0x01, 0x48, 0x60,  0x48, 0x60, 0x00, 0x00,  0x00, 0x00, 0x00, 0x00,  0x00, 0x00, 0x88, 0x88}, // DNS6 server  .ipmode = NETINFO\_STATIC\_ALL  #else  .dhcp = NETINFO\_STATIC  #endif  }; |

1. Setup MQTT configuration in ‘wizchip\_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}; |

* 1. 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, ‘wizchip\_mqtt\_publish.uf2' is generated in 'WIZnet-PICO-C/build/examples/mqtt/publish' directory.
   1. Step 5: Upload and Run
3. [텍스트, 소프트웨어, 컴퓨터 아이콘, 웹 페이지이(가) 표시된 사진

   자동 생성된 설명](https://github.com/WIZnet-ioNIC/WIZnet-PICO-C/blob/main/static/images/mqtt/publish/raspberry_pi_pico_usb_mass_storage.png) While pressing the BOOTSEL button of Raspberry Pi Pico, W5100S-EVB-Pico, W5500-EVB-Pico, W55RP20-EVB-Pico, W5100S-EVB-Pico2, W5500-EVB-Pico2, W6100-EVB-Pico2 or W6300-EVB-Pico2 power on the board, the USB mass storage 'RPI-RP2' is automatically mounted.

Figure 1. USB mass storage

1. Drag and drop ' wizchip\_mqtt\_publish.uf2' onto the USB mass storage device 'RPI-RP2'.
2. Connect to the serial COM port of Raspberry Pi Pico, W5100S-EVB-Pico, W5500-EVB-Pico, W55RP20-EVB-Pico, W5100S-EVB-Pico2, W5500-EVB-Pico2, W6100-EVB-Pico2 or W6300-EVB-Pico2 that is open with Tera Term[텍스트, 스크린샷, 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

   자동 생성된 설명](https://github.com/WIZnet-ioNIC/WIZnet-PICO-C/blob/main/static/images/mqtt/publish/connect_to_serial_com_port.png)

Figure 2. Tera Term

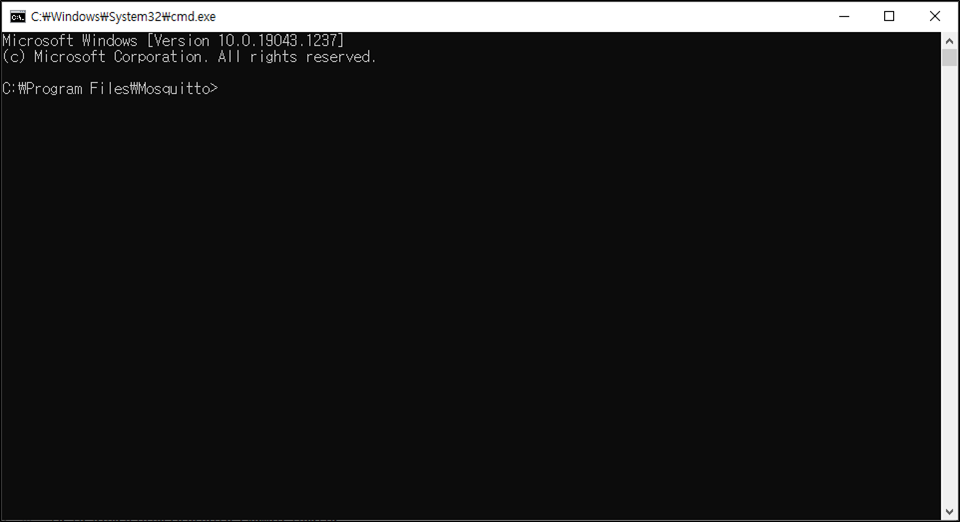
1. [](https://github.com/WIZnet-ioNIC/WIZnet-PICO-C/blob/main/static/images/mqtt/publish/run_mosquitto.png) broker using Mosquitto by executing

Figure 3. Run Mosquitto

1. 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 |

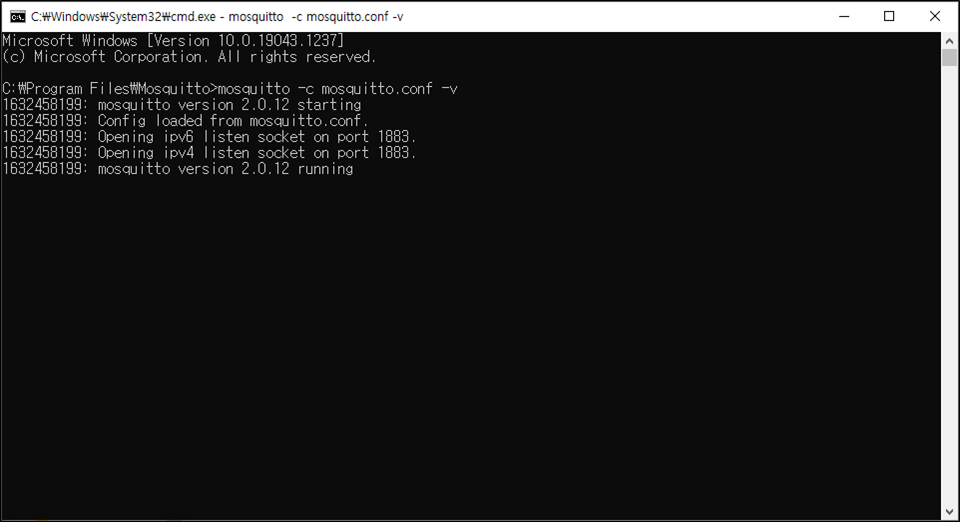
[](https://github.com/WIZnet-ioNIC/WIZnet-PICO-C/blob/main/static/images/mqtt/publish/create_mqtt_broker_using_mosquitto.png)

Figure 4. Create MQTT broker using mosquitto

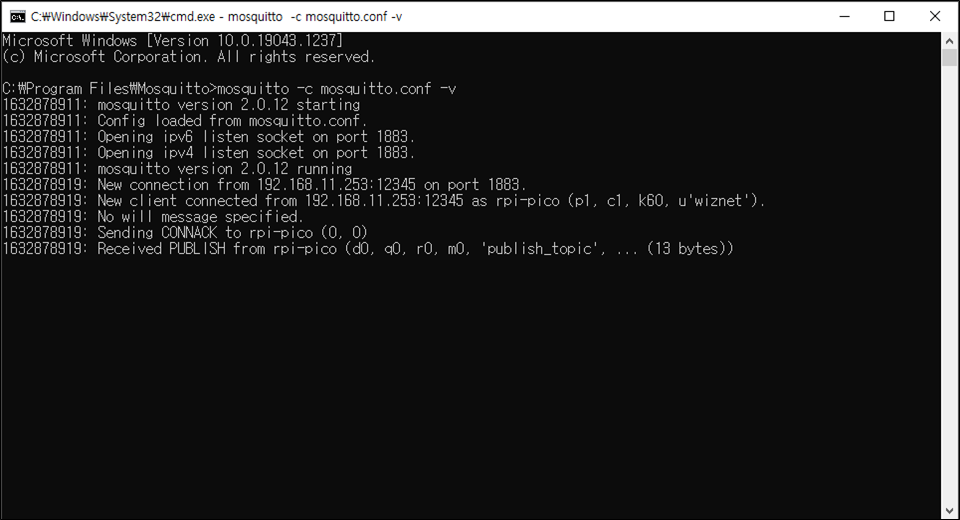
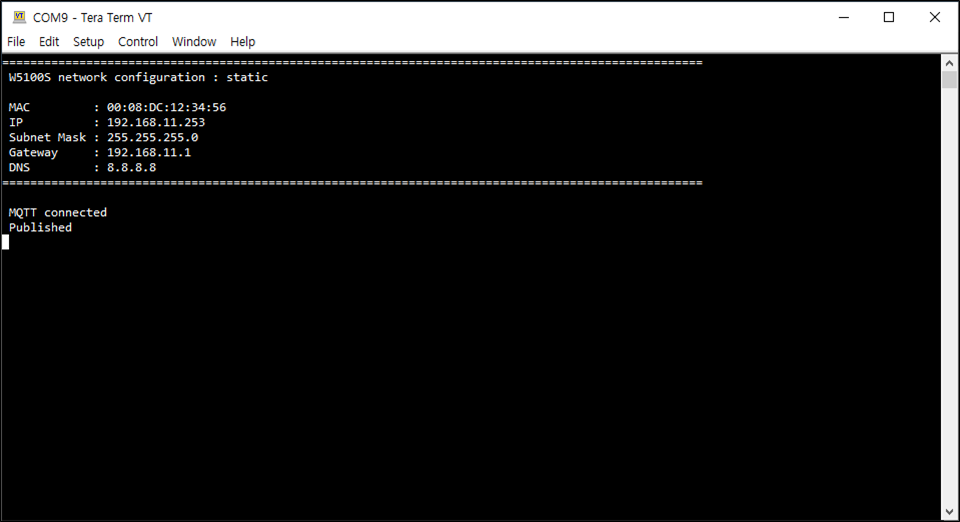
1. Reset your board.
2. [](https://github.com/WIZnet-ioNIC/WIZnet-PICO-C/blob/main/static/images/mqtt/publish/see_network_information_of_raspberry_pi_pico_connecting_to_broker_and_publishing_message_2.png)[](ttps://github.com/WIZnet-ioNIC/WIZnet-PICO-C/blob/main/static/images/mqtt/publish/see_network_information_of_raspberry_pi_pico_connecting_to_broker_and_publishing_message_1.png) If the MQTT Publish example works normally on Raspberry Pi Pico, W5100S-EVB-Pico, W5500-EVB-Pico, W55RP20-EVB-Pico, W5100S-EVB-Pico2, W5500-EVB-Pico2, W6100-EVB-Pico2 or W6300-EVB-Pico2, you can see the network information of Raspberry Pi Pico, W5100S-EVB-Pico, W5500-EVB-Pico, W55RP20-EVB-Pico, W5100S-EVB-Pico2, W5500-EVB-Pico2, W6100-EVB-Pico2 or W6300-EVB-Pico2, connecting to the broker and publishing the message.

Figure 6. Connect to broker and publish message 2

Figure 5. Connect to broker and publish message 1

* 1. 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**](https://mosquitto.org/documentation/authentication-methods/)

Revision history

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Descriptions** |
| Ver. 1.0.0 | Nov, 2024 | Initial release. |
| Ver. 1.1.0 | Jun, 2025 | Modified ioLibrary (w5x00, w6x00) integration for WIZnet chips. |

Table 1. Revision history

**Copyright Notice**

Copyright 2024 WIZnet Co., Ltd. All Rights Reserved.

Technical Support: <https://forum.wiznet.io/>

Sales & Distribution: sales@wiznet.io

For more information, visit our website at <https://www.wiznet.io/>