

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>

Contents

1 Introduction.....	3
2 Github Link	3
3 Applicable products	3
4 How to Test MQTT Publish Example.....	3
4.1 Step 1: Prepare software	3
4.2 Step 2: Prepare hardware	3
4.3 Step 3: Setup MQTT Publish Example	4
4.4 Step 4: Build	6
4.5 Step 5: Upload and Run	7
4.6 Appendix	10
Revision history	11

Figures

FIGURE 1. USB MASS STORAGE.....	7
FIGURE 2. TERA TERM	7
FIGURE 3. RUN MOSQUITTO	8
FIGURE 4. CREATE MQTT BROKER USING MOSQUITTO.....	8
FIGURE 5. CONNECT TO BROKER AND PUBLISH MESSAGE 1	9
FIGURE 6. CONNECT TO BROKER AND PUBLISH MESSAGE 2	9

Tables

TABLE 1. REVISION HISTORY	11
---------------------------------	----

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](#)

[W6100-EVB-Pico2](#)

[W6300-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, 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.

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 "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_I00_PIN      18
#define PIO_SPI_DATA_I01_PIN      19
#define PIO_SPI_DATA_I02_PIN      20
#define PIO_SPI_DATA_I03_PIN      21
#define PIN_CS                    16
#define PIN_RST                   22

```

2. 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.
3. 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
};

```

3. 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};
```

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, 'wizchip_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, W5500-EVB-Pico2, W6100-EVB-Pico2 or W6300-EVB-Pico2 power on the board, the USB mass storage 'RPI-RP2' is automatically mounted.

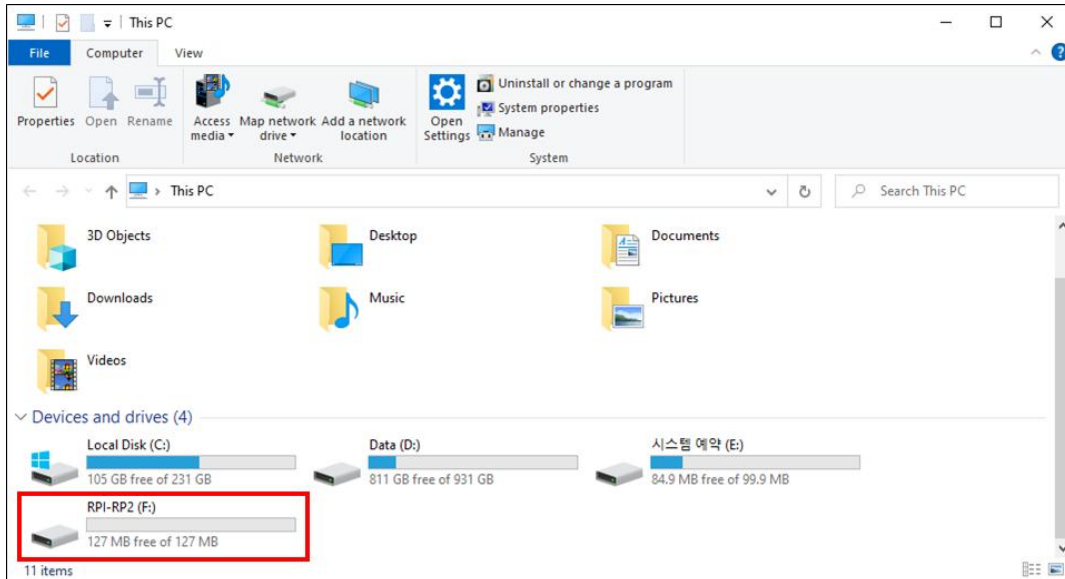


Figure 1. USB mass storage

2. Drag and drop 'wizchip_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, W5500-EVB-Pico2, W6100-EVB-Pico2 or W6300-EVB-Pico2 that is open with Tera Term

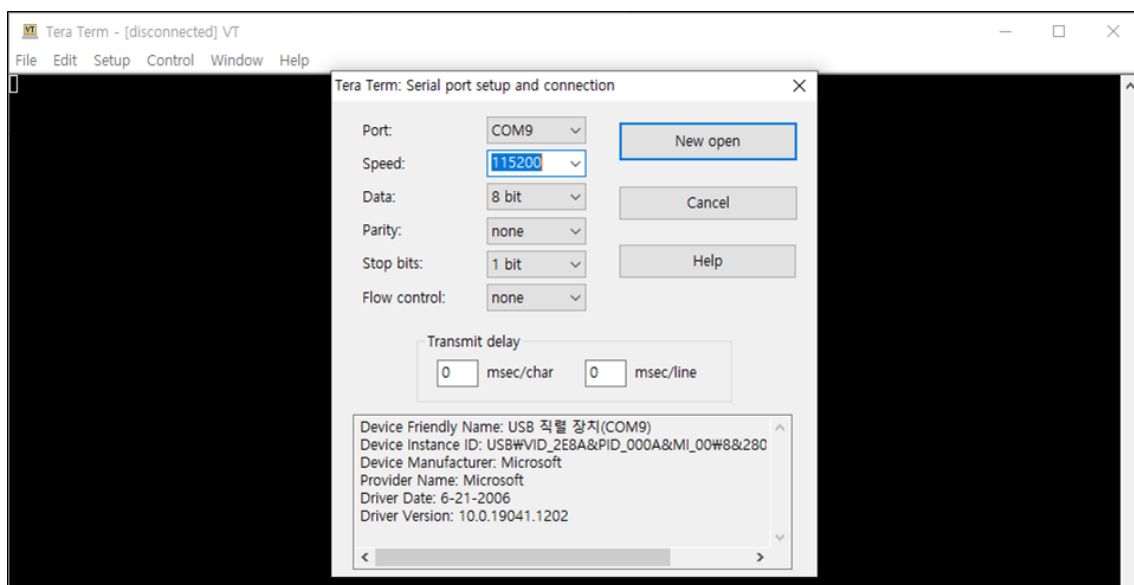


Figure 2. Tera Term

4. broker using Mosquitto by executing

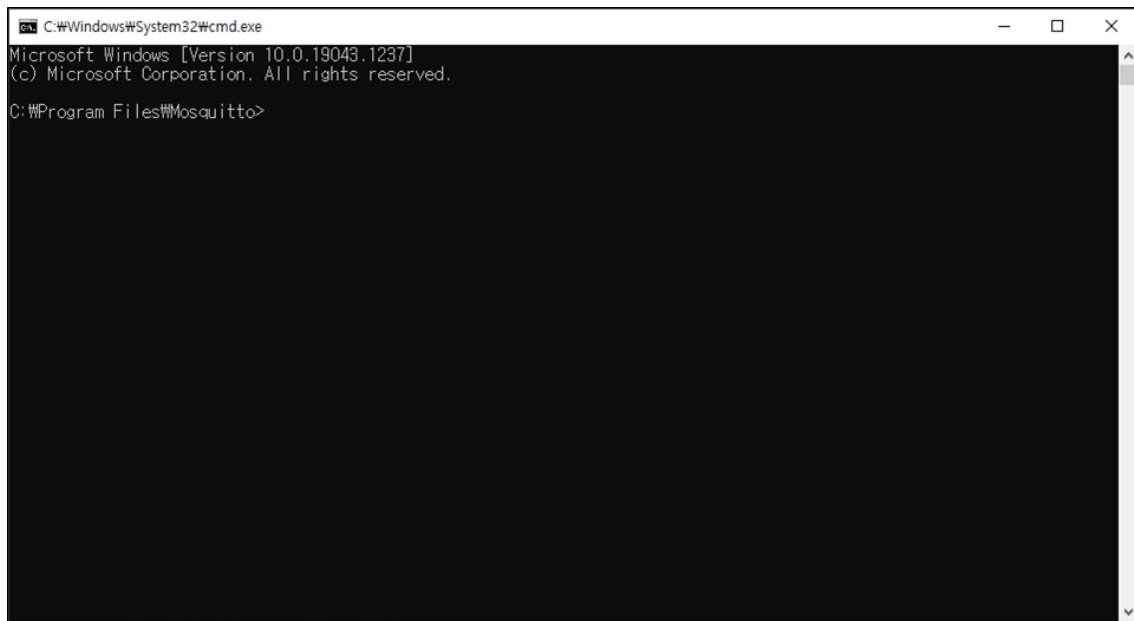


Figure 3. Run Mosquitto

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
```

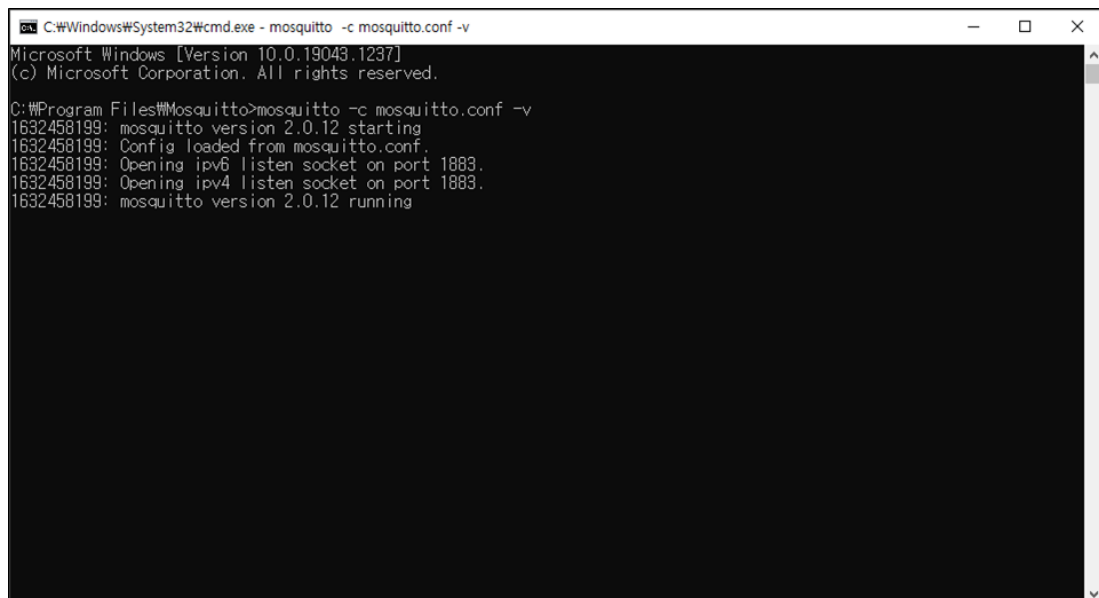
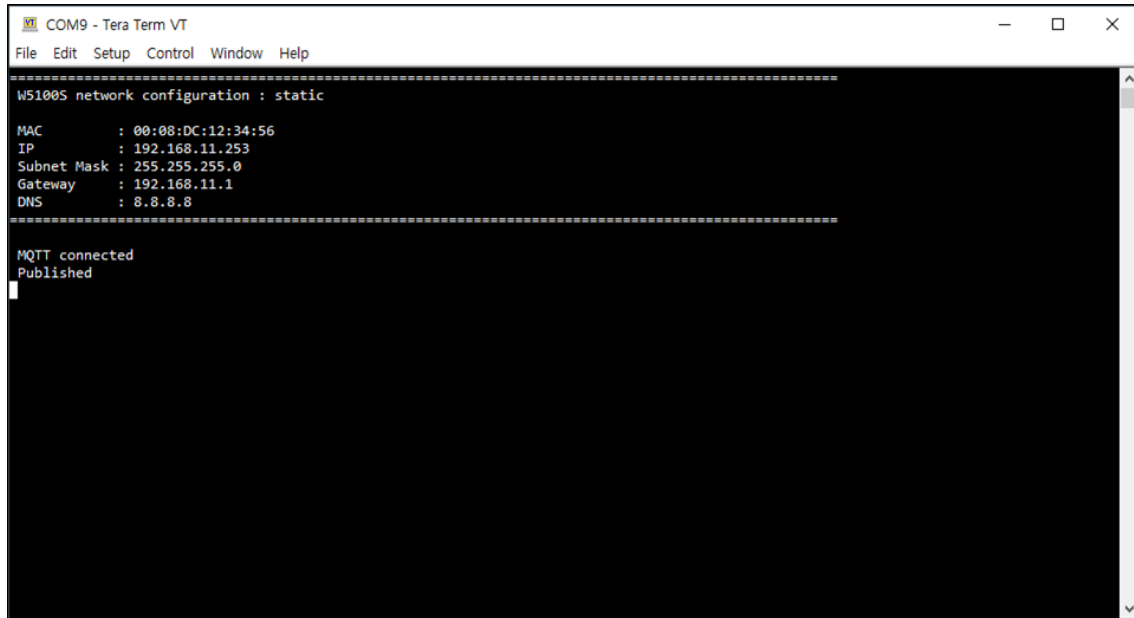


Figure 4. Create MQTT broker using mosquitto

6. Reset your board.

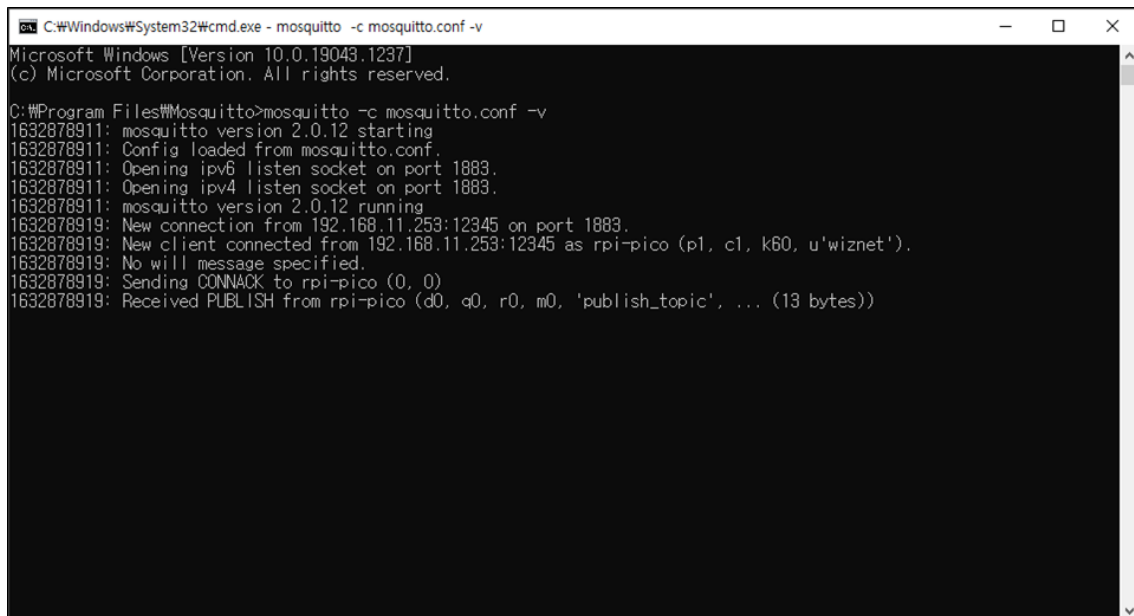
7. If the MQTT Publish example works normally on Raspberry Pi Pico, W5100S-EVB-Pico, W5500-EVB-Pico, W55RP20-EVB-Pico, W5100S-EVB-Pico2, W5500-EVB-Pico2, W6100-EVB-Pico2 or W6300-EVB-Pico2, you can see the network information of Raspberry Pi Pico, W5100S-EVB-Pico, W5500-EVB-Pico, W55RP20-EVB-Pico, W5100S-EVB-Pico2, W5500-EVB-Pico2, W6100-EVB-Pico2 or W6300-EVB-Pico2, connecting to the broker and publishing the message.



```

COM9 - Tera Term VT
File Edit Setup Control Window Help
=====
W5100S network configuration : static
MAC      : 00:08:DC:12:34:56
IP       : 192.168.11.253
Subnet Mask : 255.255.255.0
Gateway   : 192.168.11.1
DNS       : 8.8.8.8
=====
MQTT connected
Published
  
```

Figure 6. Connect to broker and publish message 1



```

C:\Windows\System32\cmd.exe - mosquitto -c mosquitto.conf -v
Microsoft Windows [Version 10.0.19043.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files\Mosquitto>mosquitto -c mosquitto.conf -v
1632878911: mosquitto version 2.0.12 starting
1632878911: Config loaded from mosquitto.conf.
1632878911: Opening ipv6 listen socket on port 1883.
1632878911: Opening ipv4 listen socket on port 1883.
1632878911: mosquitto version 2.0.12 running
1632878919: New connection from 192.168.11.253:12345 on port 1883.
1632878919: New client connected from 192.168.11.253:12345 as rpi-pico (p1, c1, k60, u'wiznet').
1632878919: No will message specified.
1632878919: Sending CONNACK to rpi-pico (0, 0)
1632878919: Received PUBLISH from rpi-pico (d0, q0, r0, m0, 'publish_topic', ... (13 bytes))
  
```

Figure 5. 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.
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/>