

# Application Note

# AZURE\_2CD Example

**Version 1.0.0**



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

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

## Contents

<b>1 Introduction.....</b>	<b>4</b>
<b>2 Github Link .....</b>	<b>4</b>
<b>3 Applicable products .....</b>	<b>4</b>
<b>4 How to Test AZURE 2CD Example .....</b>	<b>4</b>
4.1 Step 1: Prepare software .....	4
4.2 Step 2: Prepare hardware .....	4
4.3 Step 3: Setup AZURE 2CD Example .....	5
4.4 Step 4: Setup Azure IoT Explorer .....	7
4.5 Step 5: Build .....	13
4.6 Step 6: Upload and Run .....	14
<b>Revision history .....</b>	<b>21</b>

## Figures

FIGURE 1. ADD IOT DEVICES.....	7
FIGURE 2. CREATE A DEVICE .....	8
FIGURE 3. DEVICE SUCCESSFULLY CREATED .....	8
FIGURE 4. CHECK THE DEVICE.....	9
FIGURE 5. COPY THE KEY STRING.....	9
FIGURE 6. SET UP AZURE IOT EXPLORER .....	10
FIGURE 7. GETTING CONNECTION STRING .....	10
FIGURE 8. ADD CONNECTION STRING .....	11
FIGURE 9. SELECT THE DEVICE .....	12
FIGURE 10. START TELEMETRY .....	12
FIGURE 11. RECEIVING EVENTS.....	13
FIGURE 12. USB MASS STORAGE.....	14
FIGURE 13. TERA TERM .....	14
FIGURE 14. NETWORK INFO AND CONNECT TO AZURE IOT HUB.....	15
FIGURE 15. GETTING DEVICE MESSAGES FROM AZURE IOT HUB.....	16
FIGURE 16. SEND CLOUD-TO-DEVICE MESSAGE .....	17
FIGURE 17. CHECK THE SEND MESSAGE .....	17

FIGURE 18. RECEIVED THE C2D MESSAGE .....	18
FIGURE 19. SEND CLOUD-TO-DEVICE MESSAGE 2 .....	18
FIGURE 20. RECEIVED THE C2D MESSAGE 2 .....	19
FIGURE 21. SEND CLOUD-TO-DEVICE MESSAGE 3 .....	19
FIGURE 22. RECEIVED THE C2D MESSAGE 3 .....	20

## Tables

TABLE 1. REVISION HISTORY .....	21
---------------------------------	----

## 1 Introduction

This Application Note covers the implementation of AZURE 2CD on WIZnet's TOE Chip.

## 2 Github Link

<https://github.com/WIZnet-ioNIC/WIZnet-PICO-AZURE-C.git>

## 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 AZURE 2CD Example

### 4.1 Step 1: Prepare software

The following serial terminal program is required for AZURE 2CD 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 AZURE 2CD Example

To test the AZURE 2CD example, minor settings shall be done in code.

1. Setup SPI port and pin in 'w5x00\_spi.h' in 'WIZnet-PICO-AZURE-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 AZURE 2CD 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. In 'WIZnet-PICO-AZURE-C/examples/main.c', uncomment APP\_2CD to choose the sample application.

```
(...)
```

```
// The application you wish to use should be uncommented
//
//#define APP_TELEMETRY
#define APP_C2D
//#define APP_CLI_X509
//#define APP_PROV_X509
```

3. Setup network configuration such as IP in 'main.c', which is the AZURE 2CD example in 'WIZnet-PICO-AZURE-C/examples/' directory.
  - Setup IP, other network settings to suit your network environment.

```
// The application you wish to use DHCP mode should be uncommented
#define _DHCP
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
#ifdef _DHCP
    .dhcp = NETINFO_DHCP // DHCP enable/disable
#else
    // this example uses static IP
    .dhcp = NETINFO_STATIC
#endif
};
```

4. Edit the 'WIZnet-PICO-AZURE-C/exmaples/sample\_certs.c' entering the proper connection string and key value from the Azure Portal:

```
/* Paste in the your iotHub connection string */
const char pico_az_connectionString[] = "[device connection string]";
```

## 4.4 Step 4: Setup Azure IoT Explorer

In Azure portal, you need to create a device and get the connection string informations as below:

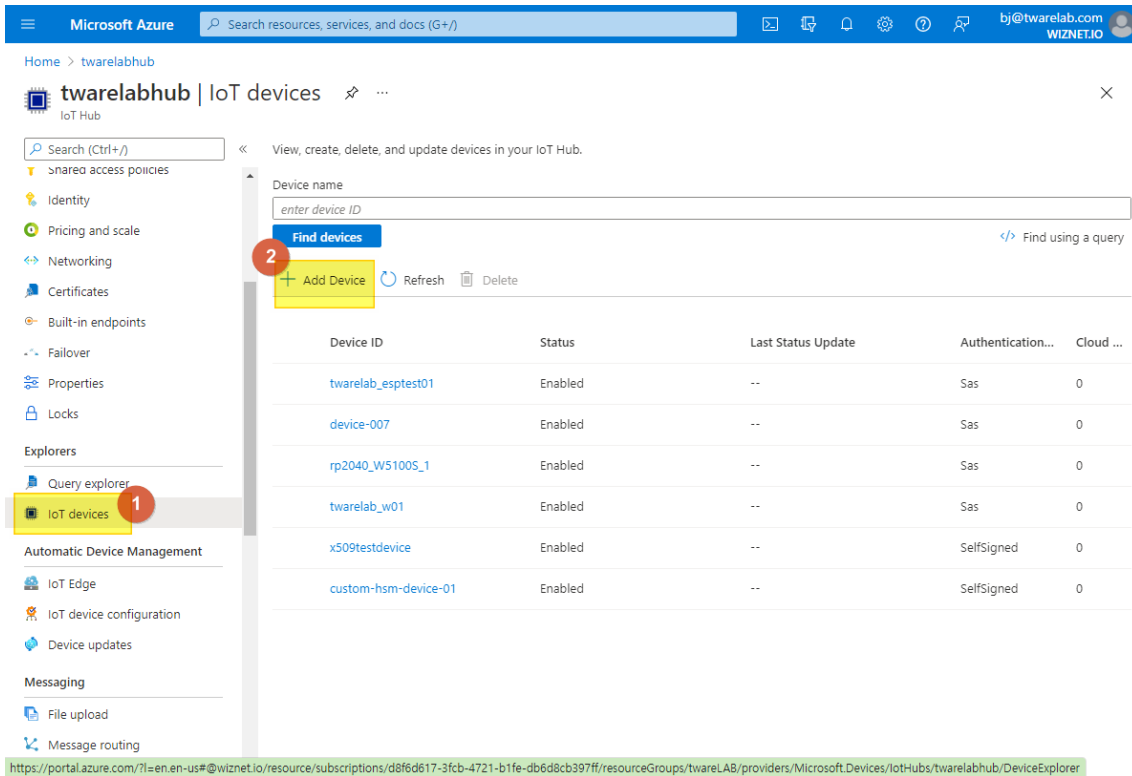


Figure 1. Add IoT devices

This example uses symmetric key

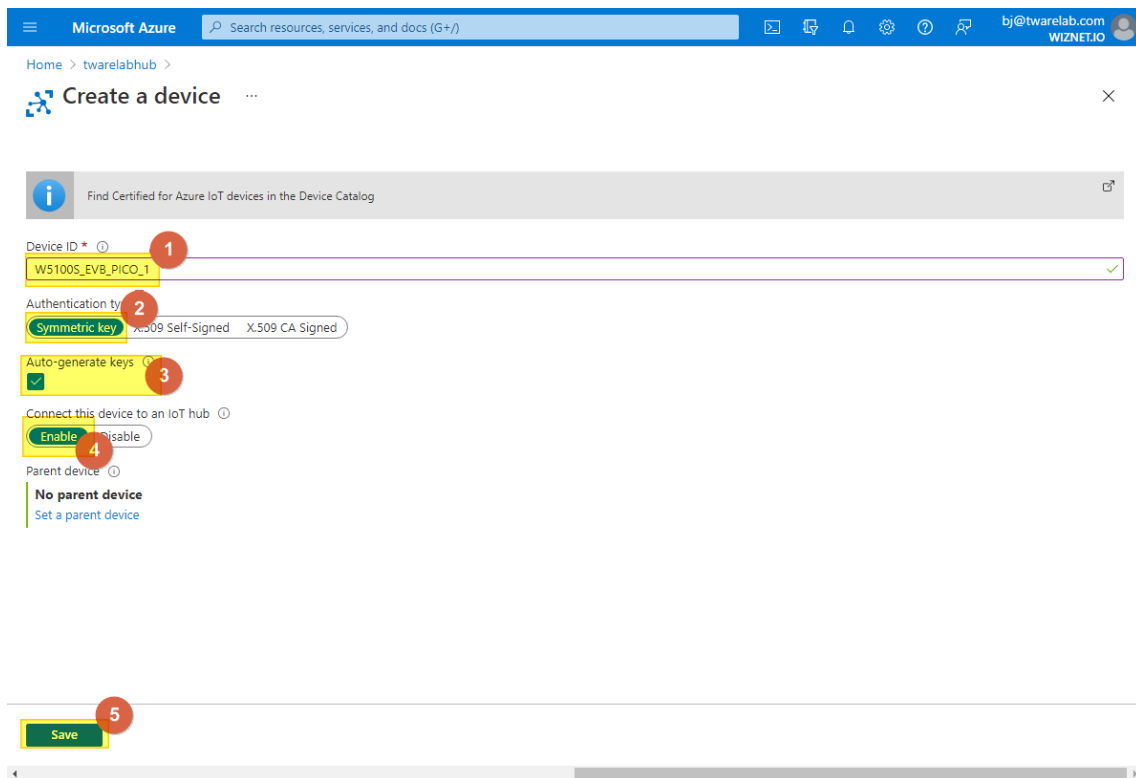


Figure 2. Create a device

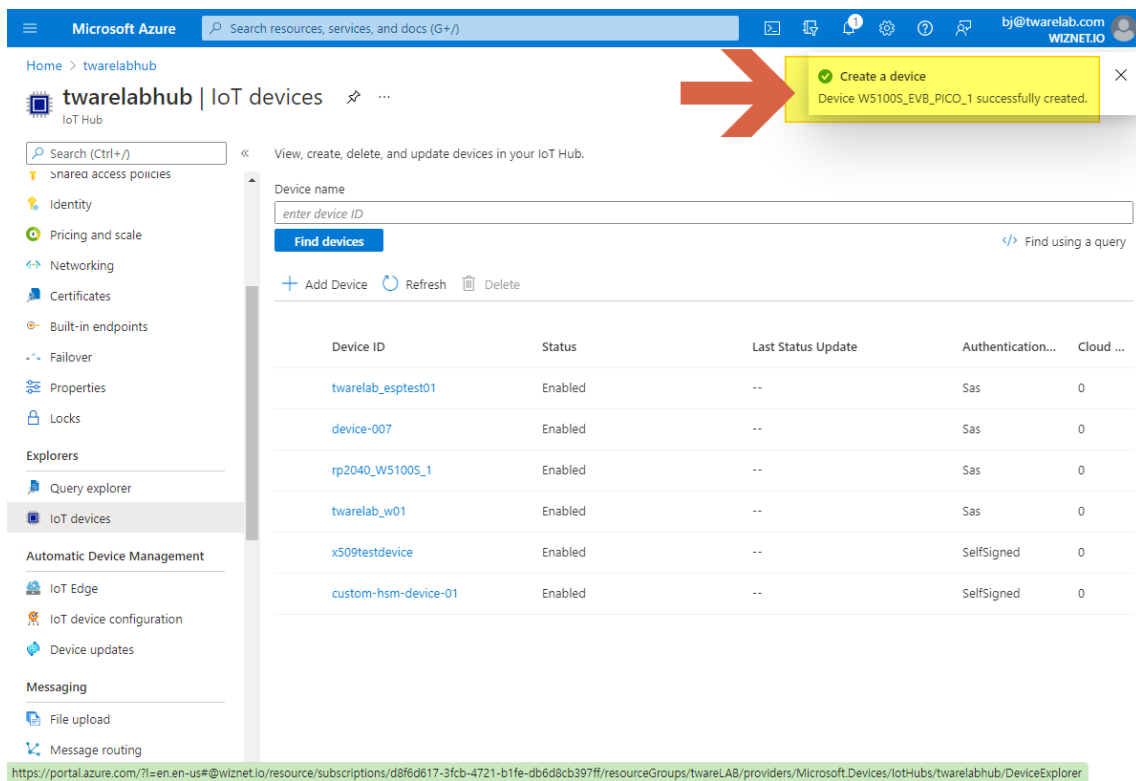


Figure 3. Device successfully created



Microsoft Azure | Search resources, services, and docs (G+)

Home > twarelabhub

twarelabhub | IoT devices

Search (Ctrl+/) << View, create, delete, and update devices in your IoT Hub.

Device name: enter device ID

Find devices Find using a query

+ Add Device Refresh Delete

Device ID	Status	Last Status Update	Authentication...	Cloud ...
twarelab_esptest01	Enabled	--	Sas	0
device-007	Enabled	--	Sas	0
rp2040_W5100S_1	Enabled	--	Sas	0
W5100S_EVB_PICO_1	Enabled	--	Sas	0
twarelab_w01	Enabled	--	Sas	0
x509testdevice	Enabled	--	SelfSigned	0
custom-hsm-device-01	Enabled	--	SelfSigned	0

Figure 4. Check the device

You copy the key string, "Primary Connection String" and paste the string into your code as described in next section.

Microsoft Azure | Search resources, services, and docs (G+)

Home > twarelabhub > W5100S\_EVB\_PICO\_1

Save Message to Device Direct Method Add Module Identity Device twin Manage keys Refresh

Device ID: W5100S\_EVB\_PICO\_1

Primary Key: [Redacted]

Secondary Key: [Redacted]

Primary Connection String: HostName=twarelabhub.azure-devices.net,DeviceId=W5100S\_EVB\_PICO\_1,SharedAccessKey=11Yabvml[Redacted].tls= [Redacted]

Secondary Connection String: [Redacted]

Enable connection to IoT Hub:  Enable  Disable

Parent device: No parent device

Module Identities Configurations

Figure 5. Copy the key string

To see the message from your IoT Device, you need to make a "Azure IoT Explorer" setting as below:

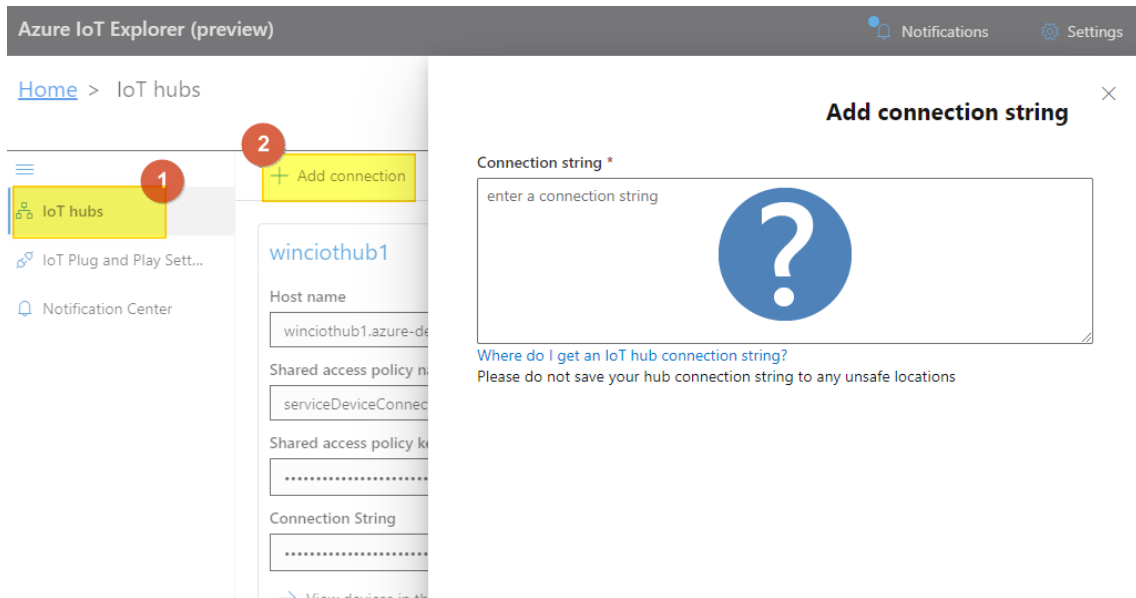


Figure 6. Set up Azure IoT Explorer

1. In Azure portal, you can get the "Connection String" as follows:

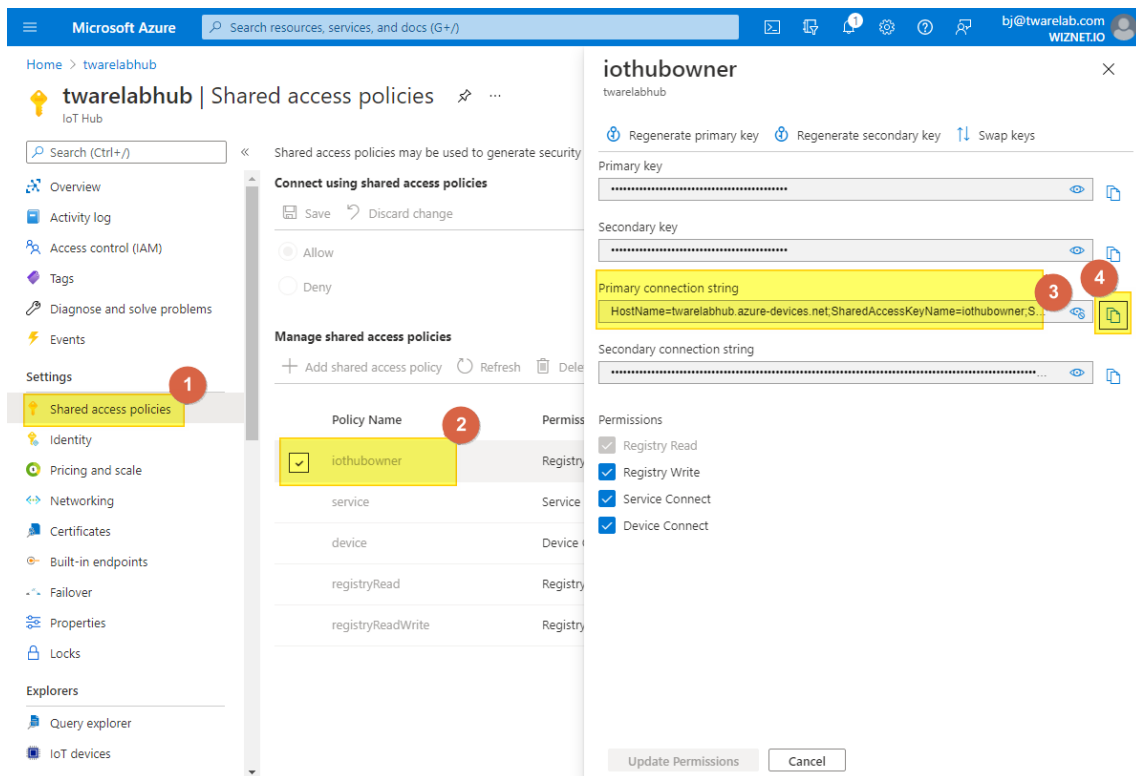


Figure 7. Getting connection string

2. Copy & paste the connection string, and click "Save".

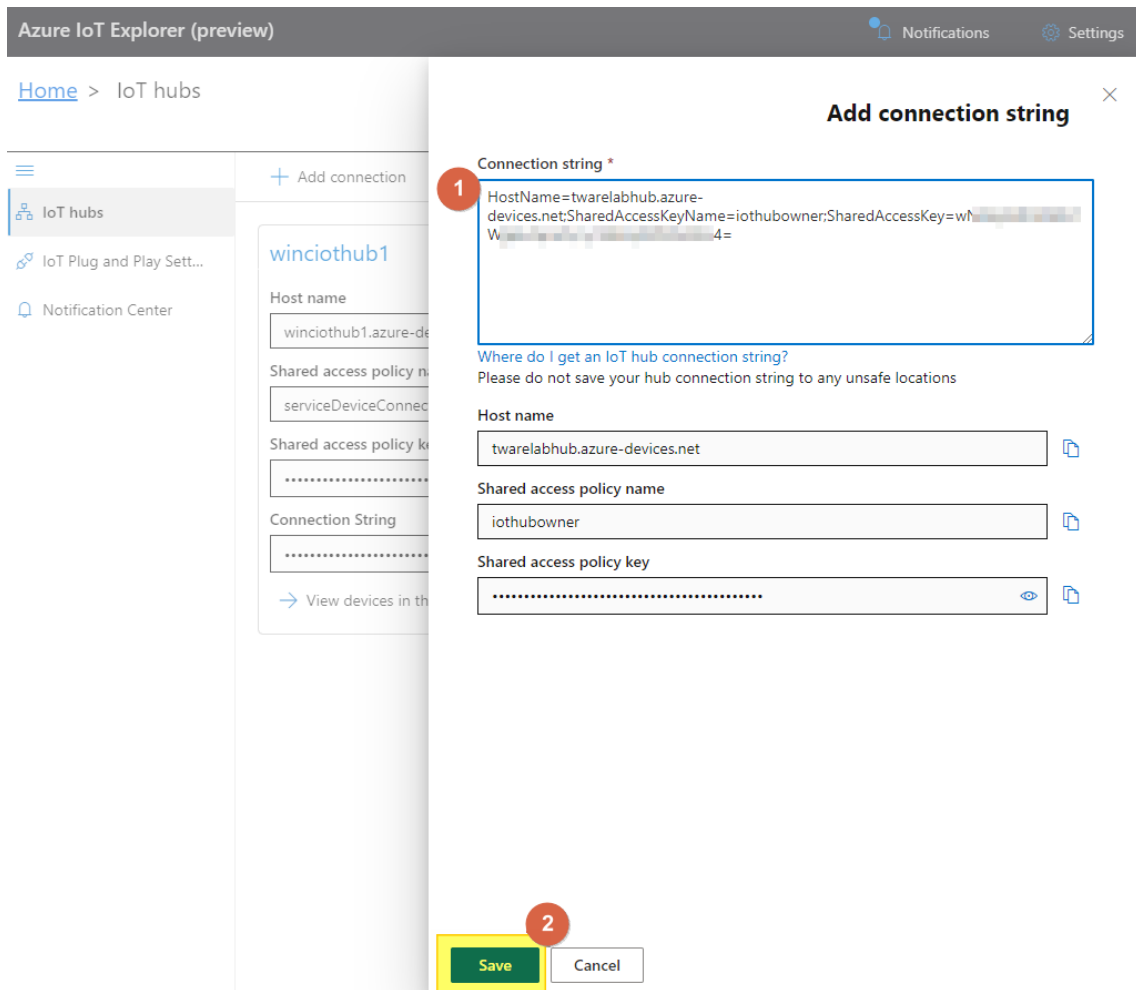


Figure 8. Add connection string

3. Find the device and click name.

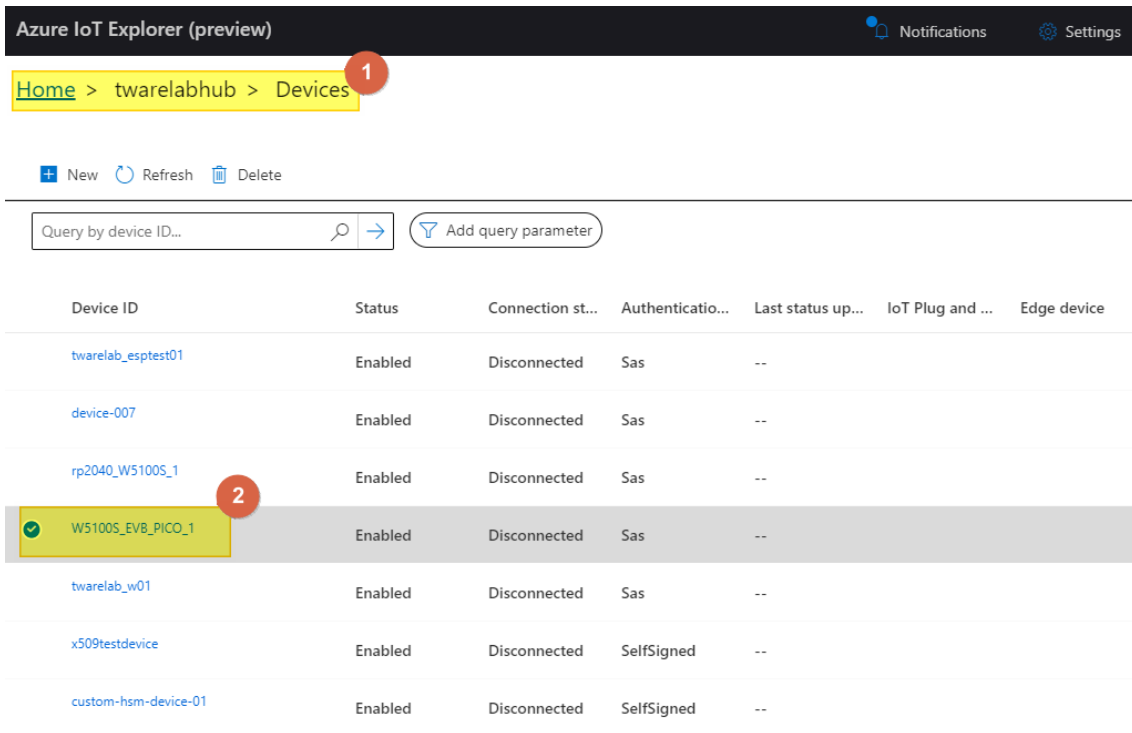


Figure 9. Select the device

4. Go to "Telemetry" menu, and click "Start".

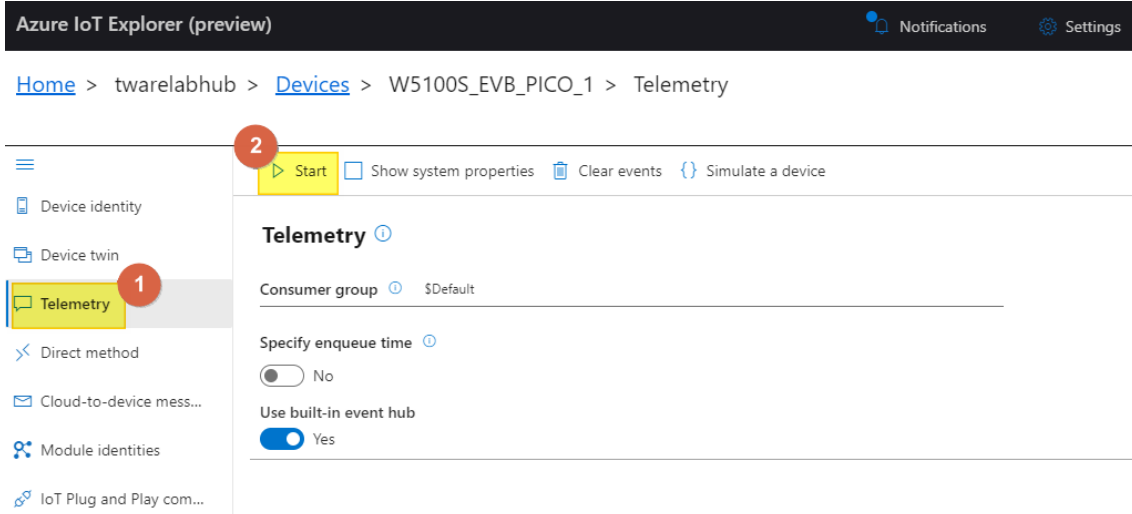


Figure 10. Start Telemetry

5. Wait for incoming message from your IoT device.

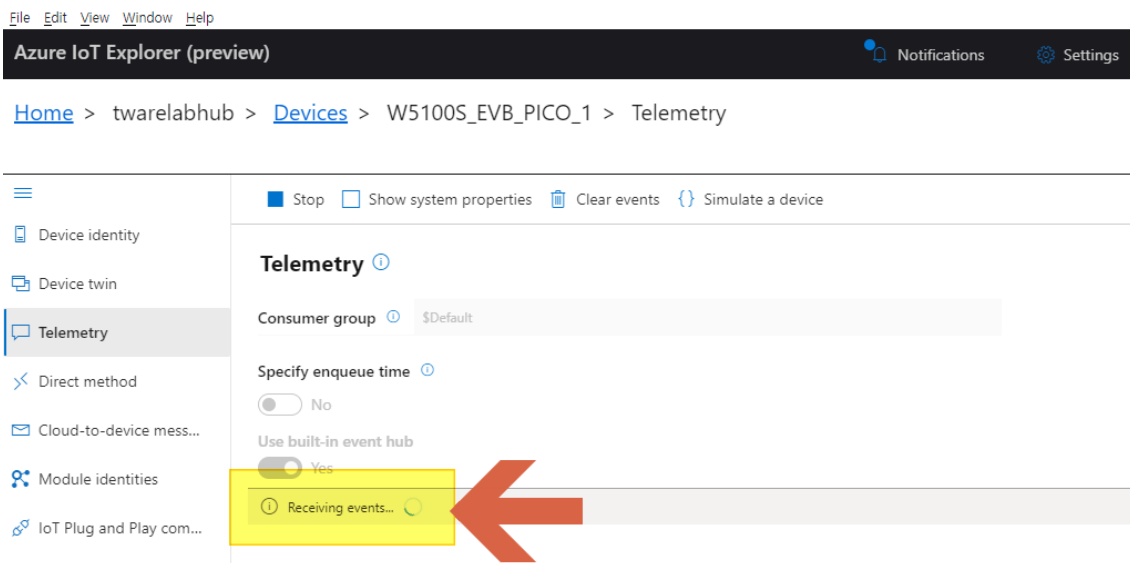


Figure 11. Receiving events

## 4.5 Step 5: Build

1. After completing the AZURE 2CD 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, 'main.uf2' is generated in 'WIZnet-PICO-AZURE-C/build/examples/' 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.

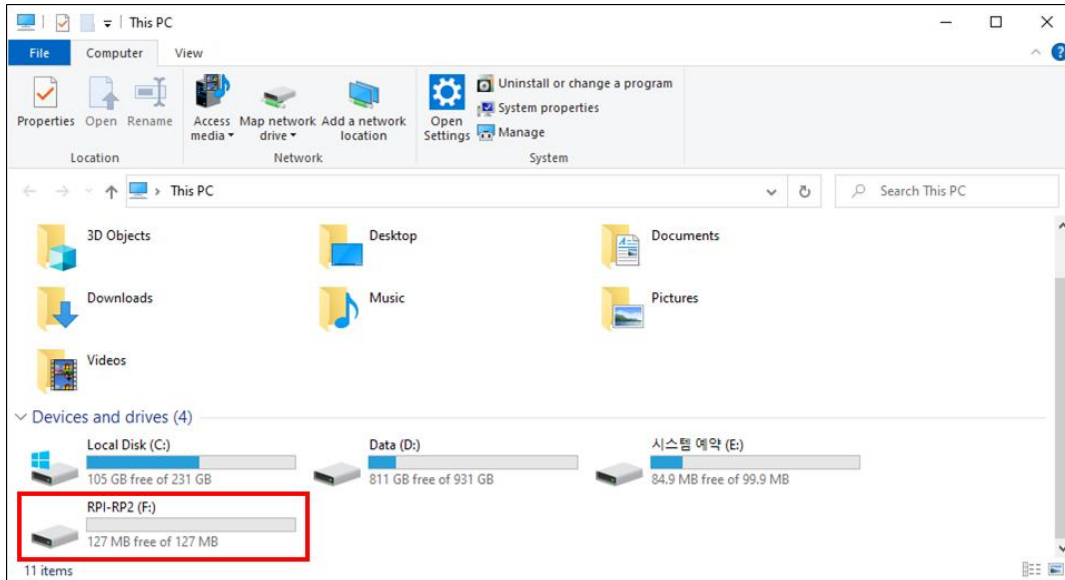


Figure 12. USB mass storage

2. Drag and drop 'main.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.

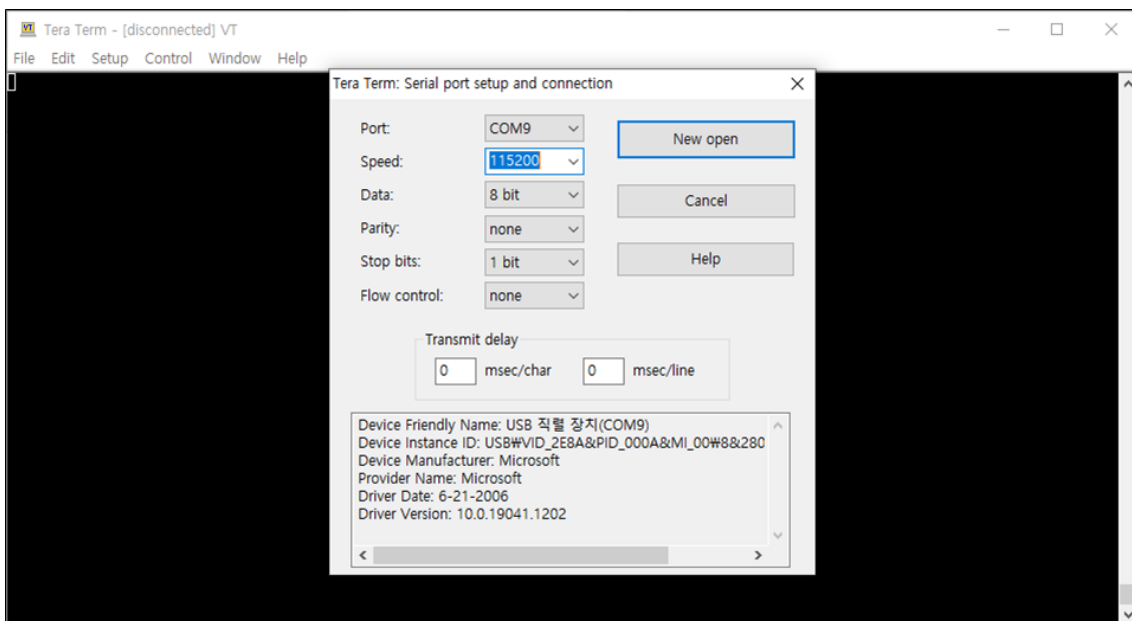


Figure 13. Tera Term

4. Reset your board.

5. If the Azure 2CD 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 Azure IoT Hub and sending the messages.

```
COM10 - Tera Term VT
File Edit Setup Control Window Help
=== socketio_dowork data recved 0 ===
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:1929: tn_left: 0, nb_want: 5
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:1932: ssl->f_recv(_timeout()) returned 5 (-0xffffffffb)
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:1952: <= fetch input
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:3452: input record: msgtype = 23, version = [3:3], msglen = 29
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:1749: => fetch input
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:1904: tn_left: 5, nb_want: 34
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:1929: tn_left: 5, nb_want: 34
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:1932: ssl->f_recv(_timeout()) returned 29 (-0xffffffff3)
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:1952: <= fetch input
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:1211: => decrypt buf
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:1715: <= decrypt buf
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:3774: <= read record
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:5399: <= read
<- 23:37:26 SUBACK | PACKET_ID: 2 | RETURN_CODE: 1
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:5205: => read
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:3700: => read record
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:1749: => fetch input
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:1904: tn_left: 0, nb_want: 5
=== socketio_dowork data recved 0 ===
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:1929: tn_left: 0, nb_want: 5
=== socketio_dowork data recved 0 ===
sending message 3 to IoTHub
Message: {"temperature":28.863,"humidity":74.205,"scale":"Celsius"}
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:5205: => read
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:3700: => read record
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:1749: => fetch input
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:1904: tn_left: 0, nb_want: 5
=== socketio_dowork data recved 0 ===
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:1929: tn_left: 0, nb_want: 5
=== socketio_dowork data recved 0 ===
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:5486: => write
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:2543: => write record
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:0529: => encrypt buf
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:0766: before encrypt: msglen = 212, including 0 bytes of padding
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:0951: <= encrypt buf
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:2628: output record: msgtype = 23, version = [3:3], msglen = 236
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:1965: => flush output
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:1983: message length: 241, out_left: 241
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:1990: ssl->f_send() returned 241 (-0xfffffffff0f)
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:2018: <= flush output
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:2684: <= write record
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:5510: <= write
<- 23:37:32 PUBLISH | IS_DUP: false | RETAIN: 0 | QOS: DELIVER_AT_LEAST_ONCE | TOPIC_NAME: devices/W5100S_EVB_PICO_1/messages/events/display_message=Hello_RP2040_WS
```

Figure 14. Network Info and connect to Azure IoT Hub

The screenshot shows the Azure IoT Explorer interface for a device named 'W5100S\_EVB\_PICO\_1'. The left sidebar contains navigation options: Device identity, Device twin, Telemetry (selected), Direct method, Cloud-to-device message, Module identities, and IoT Plug and Play components. The main area is titled 'Telemetry' and includes controls for 'Stop', 'Show system properties', 'Clear events', and 'Simulate a device'. Below these are settings for 'Consumer group' (set to '\$Default'), 'Specify enqueue time' (set to 'No'), and 'Use built-in event hub' (set to 'Yes'). A scrollable list of events is shown, with the first event highlighted in yellow and a red arrow pointing to it. The event data is as follows:

```
Fri Oct 15 2021 08:37:33 GMT+0900 (대한민국 표준시):  
{  
  "body": {  
    "temperature": 28.863,  
    "humidity": 74.205,  
    "scale": "Celsius"  
  },  
  "enqueueTime": "Fri Oct 15 2021 08:37:33 GMT+0900",  
  "properties": {  
    "display_message": "Hello_PP2040_W5100S"  
  }  
}
```

The second event in the list is:

```
Fri Oct 15 2021 08:37:32 GMT+0900 (대한민국 표준시):  
{  
  "body": {  
    "temperature": 29.198,  
    "humidity": 71.788,  
    "scale": "Celsius"  
  },  
  "enqueueTime": "Fri Oct 15 2021 08:37:32 GMT+0900 (대한민국 표준시)",  
  "properties": {  
    "display_message": "Hello_PP2040_W5100S"  
  }  
}
```

Figure 15. Getting device messages from Azure IoT Hub



6. you can send C2D messages to your device with "Azure IoT Explorer" program as follows:

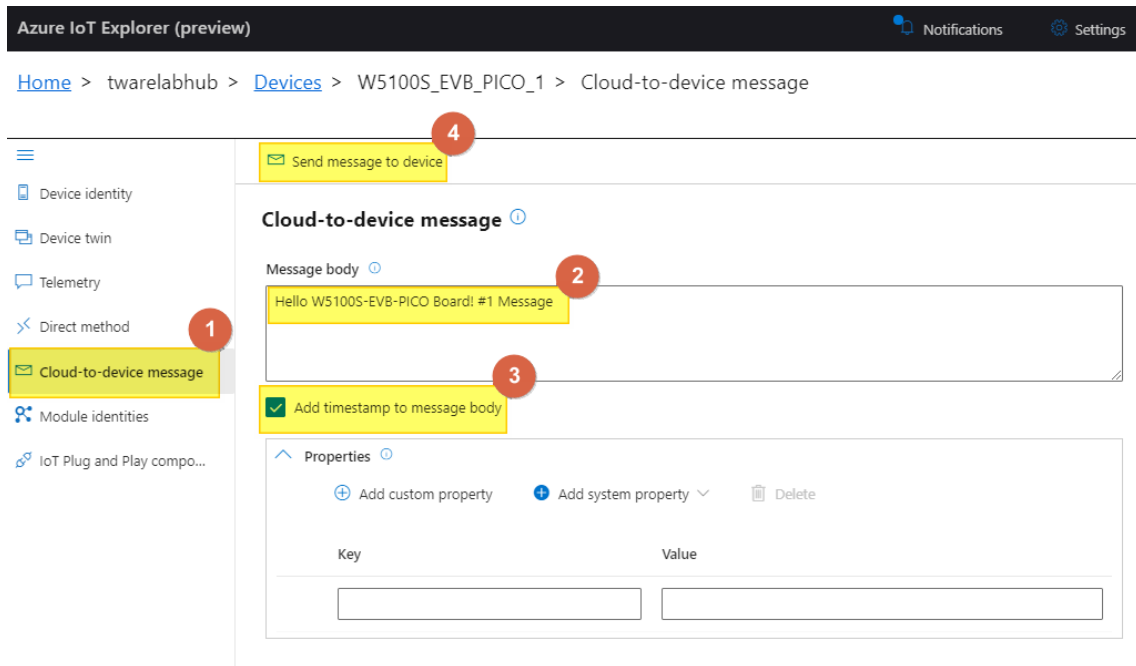


Figure 16. Send Cloud-to-device message

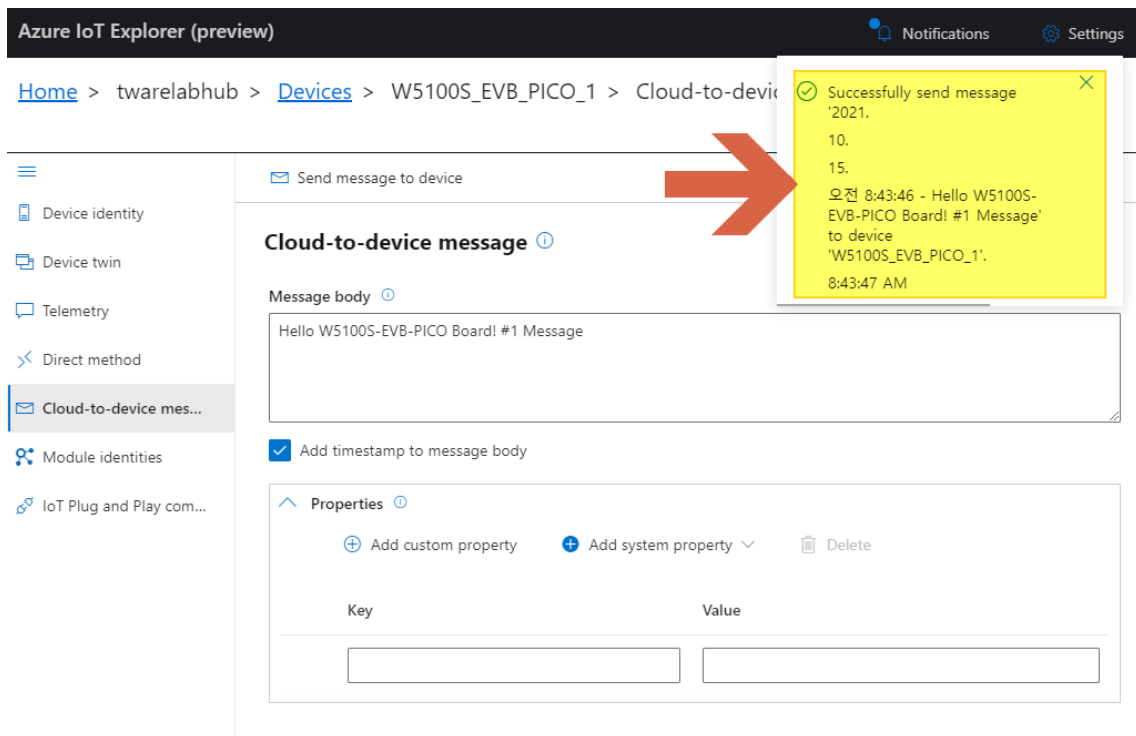


Figure 17. Check the send message

7. Then, you can see the received C2D message through your "Serial Terminal" window as below:

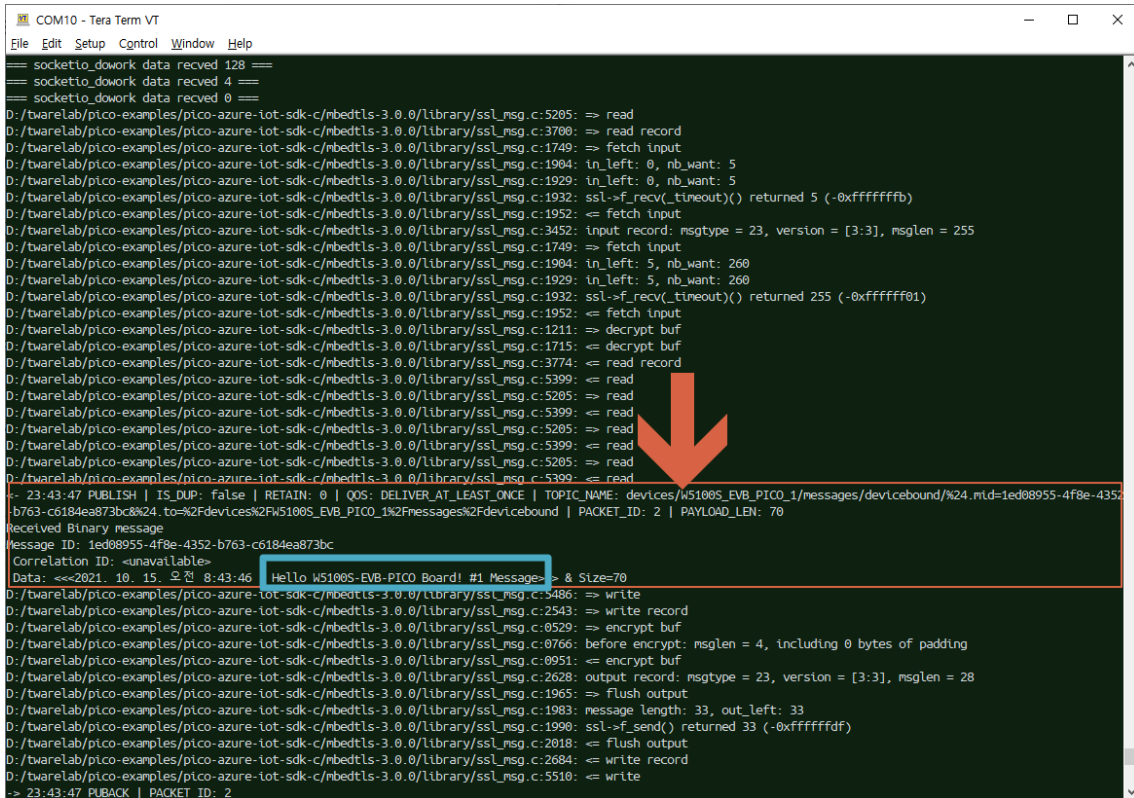


Figure 18. received the C2D message

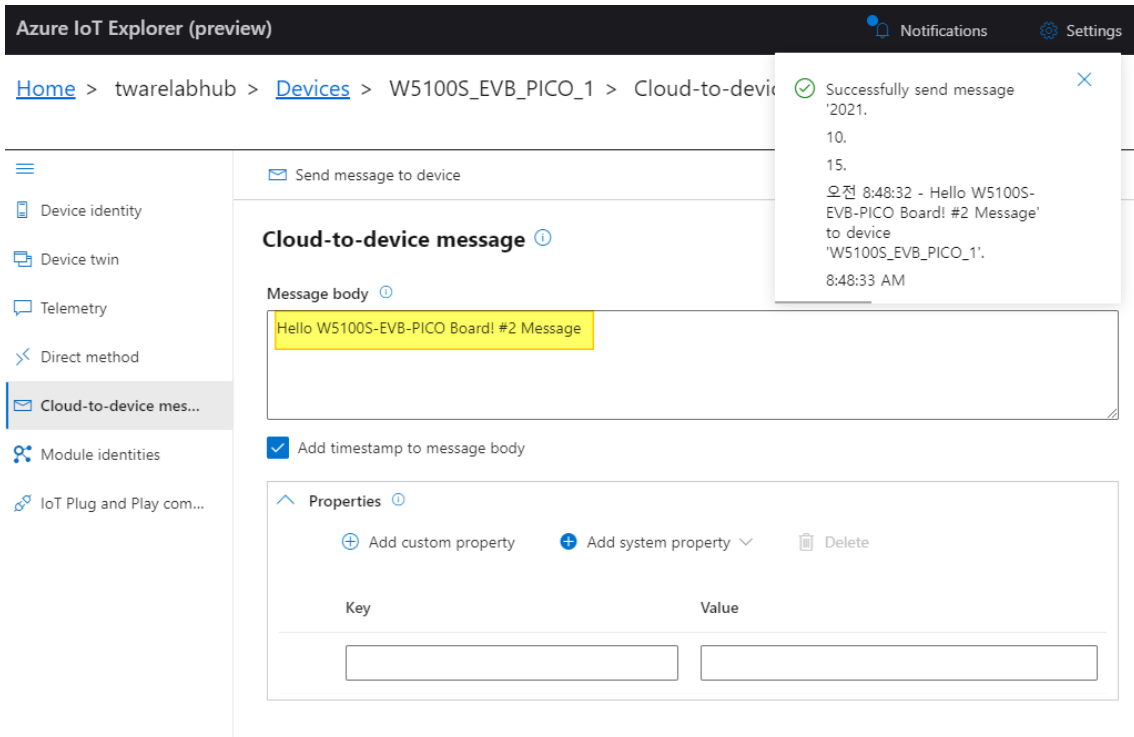


Figure 19. Send Cloud-to-device message 2

```

COM10 - Tera Term VT
File Edit Setup Control Window Help
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:3700: => read record
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:1749: => fetch input
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:1904: in_left: 0, nb_want: 5
==== socketto_dowork data recved 128 ====
==== socketto_dowork data recved 128 ====
==== socketto_dowork data recved 4 ====
==== socketto_dowork data recved 0 ====
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:1929: in_left: 0, nb_want: 5
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:1932: ssl->f_recv(timeout()) returned 5 (-0xfffffff)
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:1952: => fetch input
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:3452: input record: msgtype = 23, version = [3:3], msglen = 255
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:1749: => fetch input
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:1904: in_left: 5, nb_want: 260
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:1929: in_left: 5, nb_want: 260
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:1932: ssl->f_recv(timeout()) returned 255 (-0xfffffff01)
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:1952: => fetch input
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:1211: => decrypt buf
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:1715: => decrypt buf
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:3774: => read record
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:5399: => read
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:5205: => read
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:5399: => read
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:5205: => read
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:5399: => read
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:5205: => read
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:5399: => read
<- 23:48:32 PUBLISH | IS_DUP: False | RETAIN: 0 | QOS: DELIVER_AT_LEAST_ONCE | TOPIC_NAME: devtces/w5100s_EVB_PICO_1/messages/devtcebound/%24.mtd=1ac365d9-b2dd-474a-84a3-1f2ea904b358%24.tos=%2Fdevices%2FW5100S_EVB_PICO_1%2Fmessages%2Fdevtcebound | PACKET_ID: 3 | PAYLOAD_LEN: 70
Received Binary message
Message ID: 1ac365d9-b2dd-474a-84a3-1f2ea904b358
Correlation ID: <unavailable>
Data: <<2021. 10. 15. 오전 8:48:32 Hello W5100S-EVB-PICO Board! #2 Message>> & Size=70
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:3486: => write
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:2543: => write record
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:0529: => encrypt buf
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:0766: before encrypt: msglen = 4, including 0 bytes of padding
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:0951: => encrypt buf
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:2628: output record: msgtype = 23, version = [3:3], msglen = 28
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:1965: => flush output
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:1903: message length: 33, out_left: 33
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:1900: ssl->f_send() returned 33 (-0xfffffffdf)
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:2818: => flush output
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:2684: => write record
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/rbedtls-3.0.0/library/ssl_msg.c:5510: => write
-> 23:48:33 PUBACK | PACKET_ID: 3
  
```

Figure 20. Received the C2D message 2

The screenshot shows the Azure IoT Explorer interface. The breadcrumb navigation is: Home > twarelabhub > Devices > W5100S\_EVB\_PICO\_1 > Cloud-to-device. The 'Send message to device' section is active, showing a 'Cloud-to-device message' with a message body of 'Hello W5100S-EVB-PICO Board! #3 Message'. There is a checked option for 'Add timestamp to message body'. Below this is a 'Properties' section with buttons to 'Add custom property', 'Add system property', and 'Delete'. A notification popup in the top right corner states: 'Successfully send message '2021. 10. 15. 오전 8:50:15 - Hello W5100S-EVB-PICO Board! #3 Message' to device 'W5100S\_EVB\_PICO\_1'. 8:50:15 AM'.

Figure 21. Send Cloud-to-device message 3

```

COM10 - Tera Term VT
File Edit Setup Control Window Help
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:1749: => fetch input
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:1904: in_left: 0, nb_want: 5
== socketio_dwork data received 128 ==
== socketio_dwork data received 128 ==
== socketio_dwork data received 4 ==
== socketio_dwork data received 0 ==
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:1929: in_left: 0, nb_want: 5
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:1932: ssl->f_recv(_timeout()) returned 5 (-0xffffffffb)
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:1952: <= fetch input
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:3452: input record: msgtype = 23, version = [3:3], msglen = 255
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:1749: => fetch input
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:1904: in_left: 5, nb_want: 260
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:1929: in_left: 5, nb_want: 260
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:1932: ssl->f_recv(_timeout()) returned 255 (-0xffffffff01)
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:1952: <= fetch input
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:1211: => decrypt buf
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:1715: <= decrypt buf
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:3774: <= read record
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:5399: <= read
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:5205: => read
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:5399: <= read
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:5205: => read
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:5399: <= read
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:5205: => read
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:5399: <= read
<- 23:50:15 PUBLISH | IS_DUP: false | RETAIN: 0 | QOS: DELIVER_AT_LEAST_ONCE | TOPIC_NAME: devices/W5100S_EVB_PICO_1/messages/devicebound/%24_mid=61bcd12-cfbc-4d91-9f4a-c77192948e768%24. to=%2Fdevices%2F%5100S_EVB_PICO_1%2Fmessages%2Fdevicebound | PACKET_ID: 4 | PAYLOAD_LEN: 70
Received Binary message
Message ID: 61bcd12-cfbc-4d91-9f4a-c77192948e76
Correlation ID: <unavailable>
Data: <<<2021. 10. 15 오전 8:50:15 [redacted] Hello W5100S-EVB-PICO Board! #3 Message>>> & Size=70
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:486: => write
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:2543: => write record
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:0529: => encrypt buf
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:0766: before encrypt: msglen = 4, including 0 bytes of padding
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:0951: <= encrypt buf
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:2628: output record: msgtype = 23, version = [3:3], msglen = 28
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:1965: => flush output
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:1983: message length: 33, out_left: 33
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:1990: ssl->f_send() returned 33 (-0xffffffffdf)
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:2018: <= flush output
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:2684: <= write record
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:5310: <= write
-> 23:50:16 PUBACK | PACKET_ID: 4
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:5205: => read

```

Figure 22. Received the C2D message 3

## Revision history

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

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](mailto:sales@wiznet.io)

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