Application Note AZURE_telemetry Example

Version 1.0.0

WIZnet

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

Contents

Figures

Tables

1 Introduction

This Application Note covers the implementation of AZURE telemetry 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 telemetry Example

4.1 Step 1: Prepare software

The following serial terminal program is required for AZURE telemetry 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 telemetry Example

To test the AZURE telemetry 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 telemetry 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_TELEMETRY 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 telemetry example in 'WIZnet-PICO-AZURE-C/examples/' directory.

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

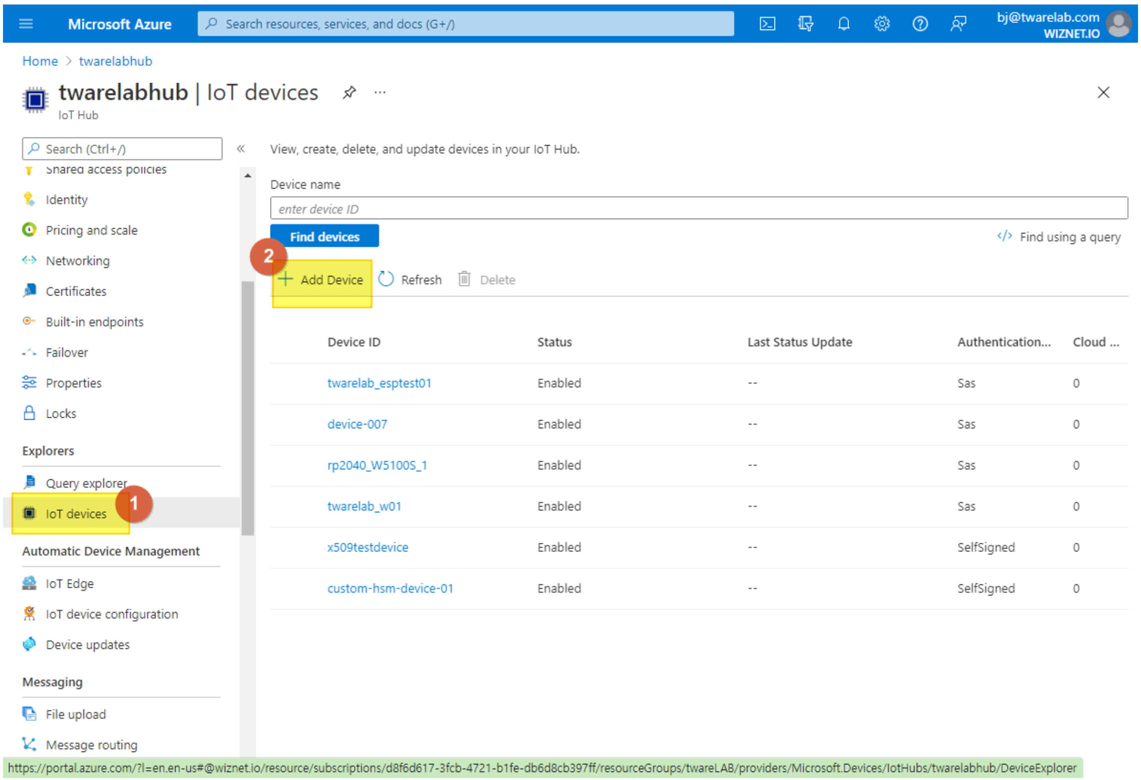
```
// The application you wish to use DHCP mode should be uncommented
#define _DHCP
static wiz_NetInfo g_net_info =
    {
        .mac = {0x00, 0x08, 0xDC, 0x12, 0x34, 0x56}, // MAC address
        .ip = {192, 168, 11, 2},                     // IP address
        .sn = {255, 255, 255, 0},                    // 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:



View, create, delete, and update devices in your IoT Hub.

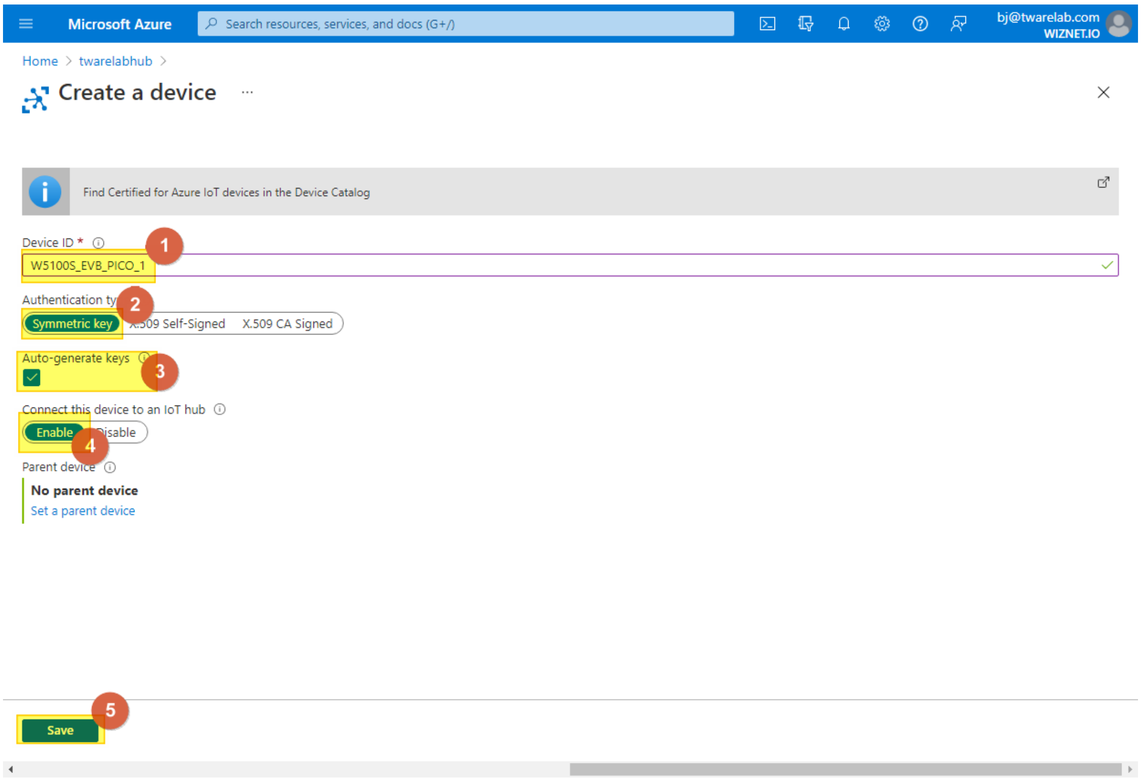
Device name: enter device ID

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

https://portal.azure.com/?l=en.en-us#@wiznet.io/resource/subscriptions/d8f6d617-3fcb-4721-b1fe-db6d8cb397ff/resourceGroups/twareLAB/providers/Microsoft.Devices/IotHubs/twarelabhub/DeviceExplorer

Figure 1. Add IoT devices

This example uses symmetric key



Create a device

Find Certified for Azure IoT devices in the Device Catalog

Device ID * ⓘ
W5100S_EVB_PICO_1

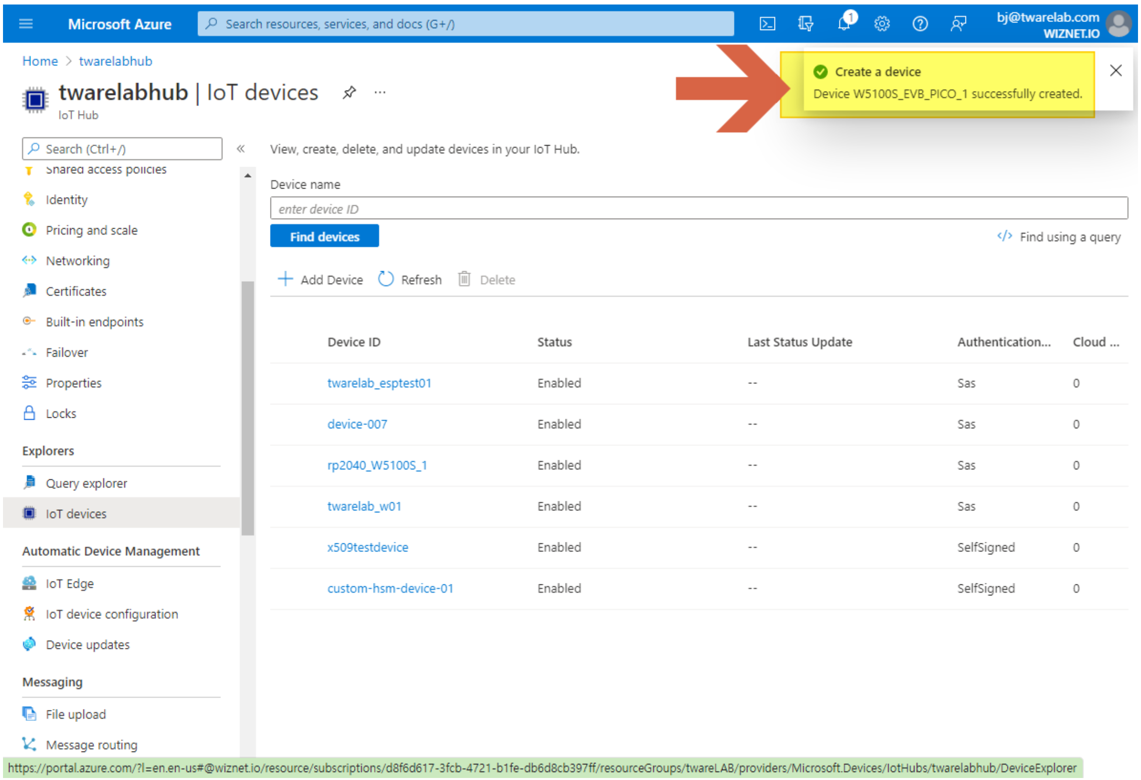
Authentication type ⓘ
[✓ Symmetric key] X.509 Self-Signed X.509 CA Signed

Auto-generate keys ⓘ
☑

Connect this device to an IoT hub ⓘ
[✓ Enable] Disable

Parent device ⓘ
No parent device
Set a parent device

Figure 2. Create a device



twarelabhub | IoT devices
IoT Hub

Create a device
Device W5100S_EVB_PICO_1 successfully created.

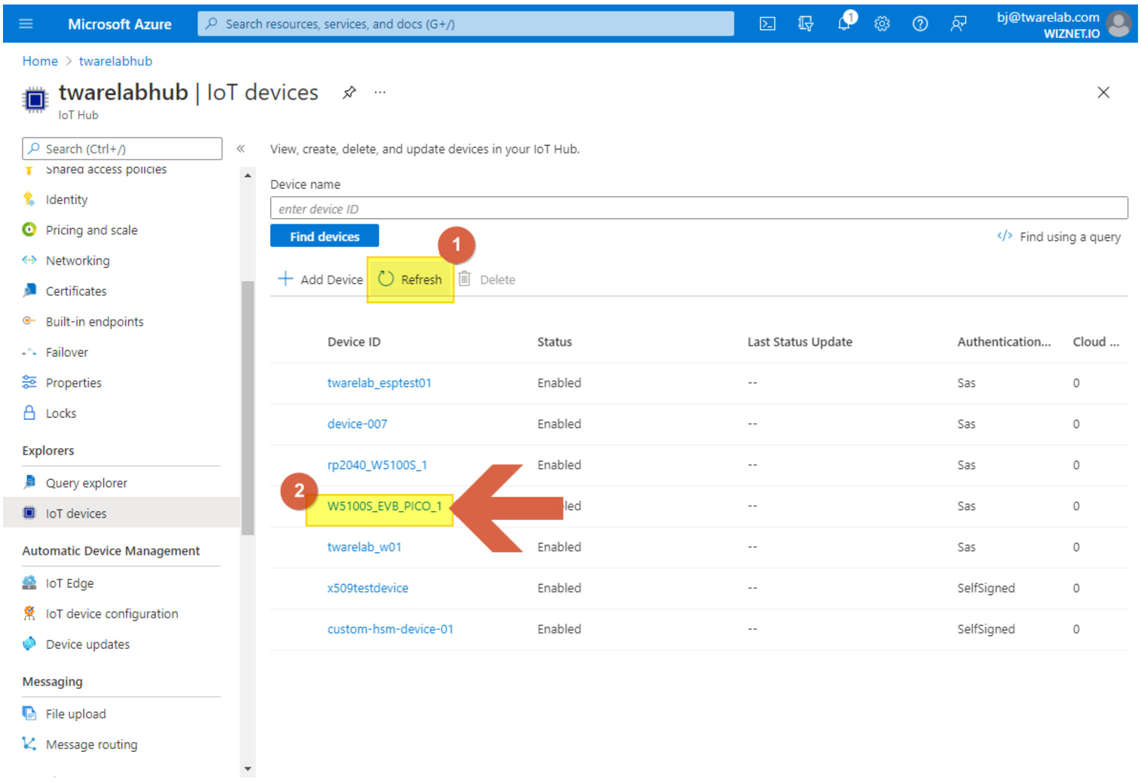
View, create, delete, and update devices in your IoT Hub.

Device name
enter device ID

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

https://portal.azure.com/?l=en.en-us#@wiznet.io/resource/subscriptions/d8f6d617-3fcb-4721-b1fe-db6d8cb397ff/resourceGroups/twareLAB/providers/Microsoft.Devices/IotHubs/twarelabhub/DeviceExplorer

Figure 3. Device successfully created



twarelabhub | IoT devices

IoT Hub

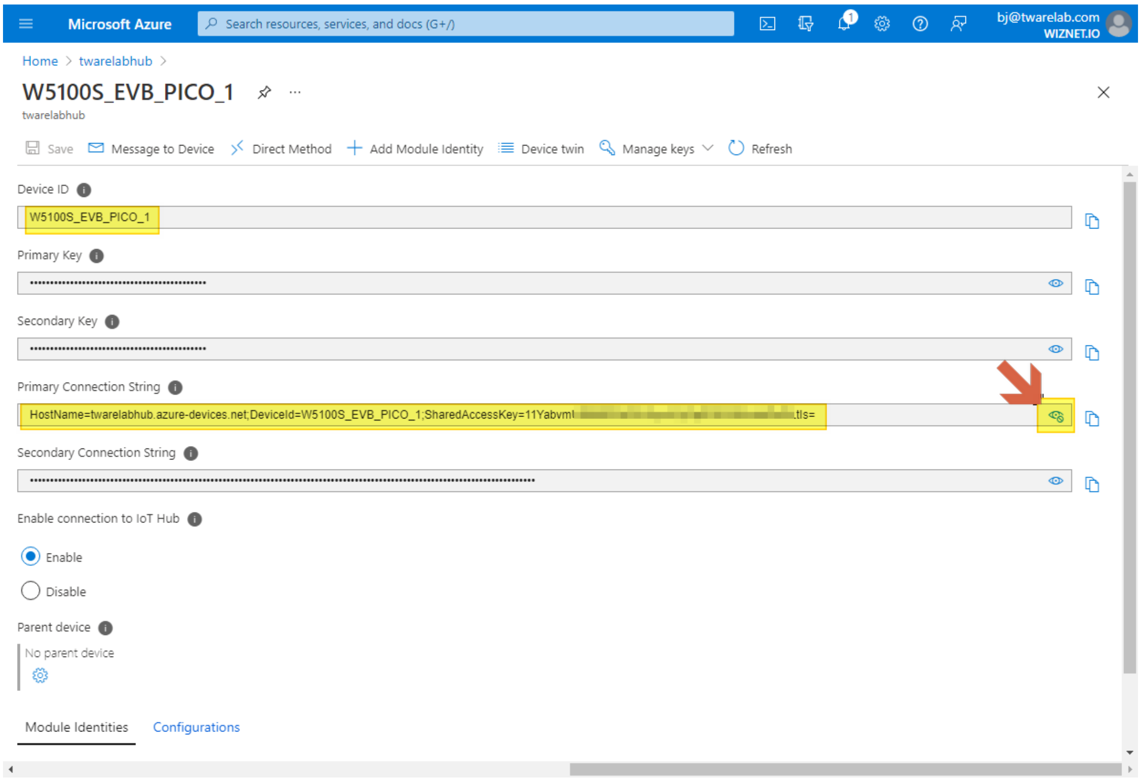
View, create, delete, and update devices in your IoT Hub.

Device name: enter device ID

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	led	--	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.



W5100S_EVB_PICO_1

twarelabhub

Device ID: W5100S_EVB_PICO_1

Primary Key: ••

Secondary Key: ••

Primary Connection String: HostName=twarelabhub.azure-devices.net;DeviceId=W5100S_EVB_PICO_1;SharedAccessKey=11Yabvml[redacted]tls=

Secondary Connection String: ••

Enable connection to IoT Hub

● Enable

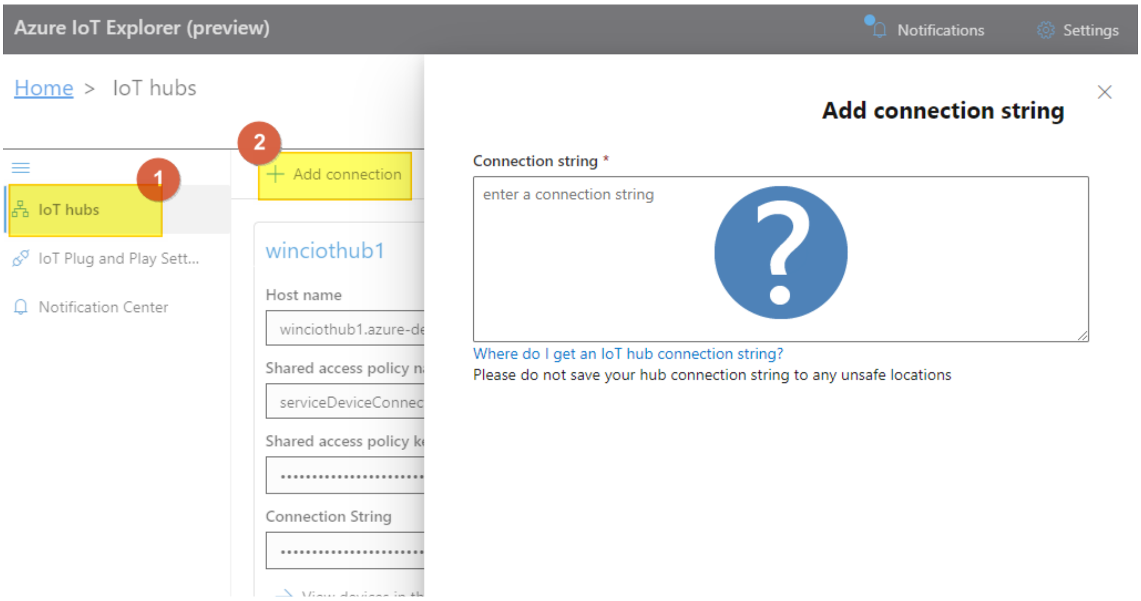
○ Disable

Parent device

No parent device

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:



winciothub1.azure-de

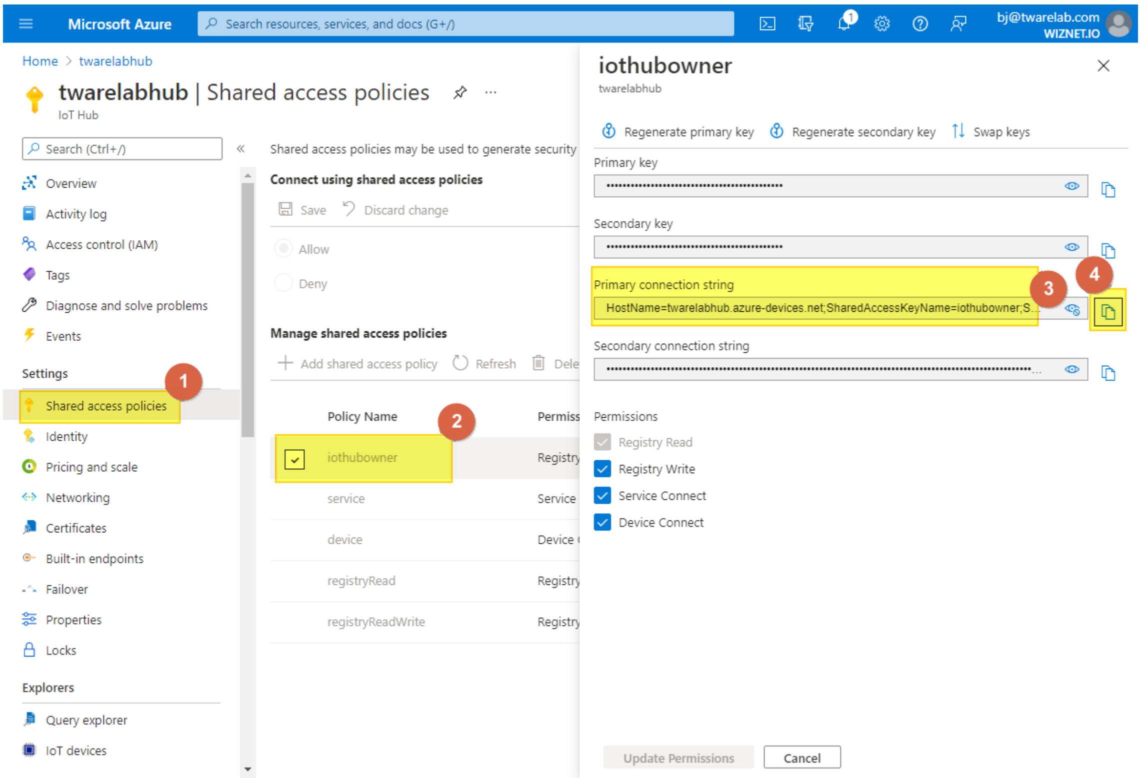
serviceDeviceConnec

enter a connection string

Please do not save your hub connection string to any unsafe locations

Figure 6. Set up Azure IoT Explorer

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

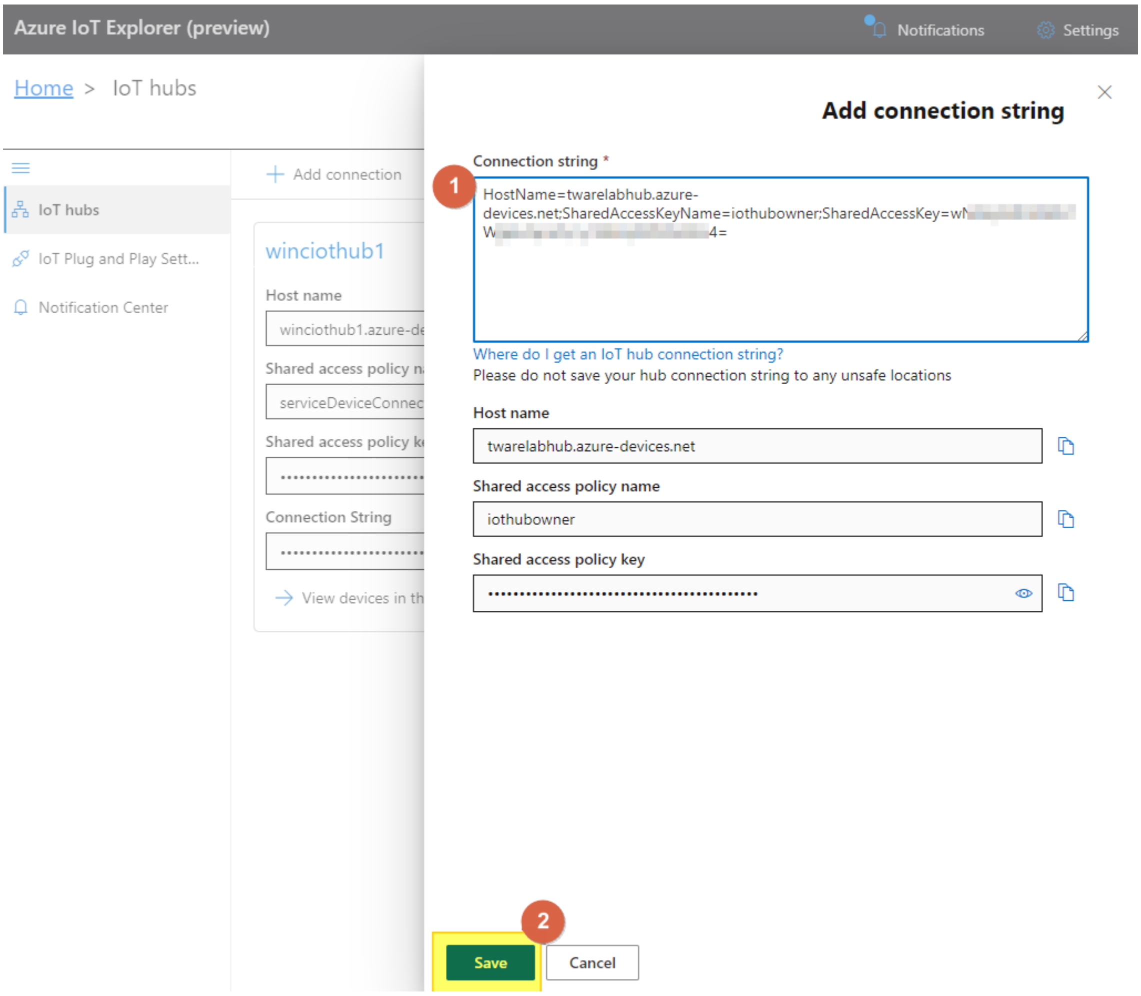


HostName=twarelabhub.azure-devices.net;SharedAccessKeyName=iothubowner;S...

☑ Registry Read
☑ Registry Write
☑ Service Connect
☑ Device Connect

Figure 7. Getting connection string

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



Azure IoT Explorer (preview)

Home > IoT hubs

winciothub1

Host name

winciothub1.azure-de

Shared access policy n

serviceDeviceConnec

Add connection string

Connection string *

HostName=twarelabhub.azure-devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=wN[redacted]4=

Where do I get an IoT hub connection string?

Please do not save your hub connection string to any unsafe locations

Host name

twarelabhub.azure-devices.net

Shared access policy name

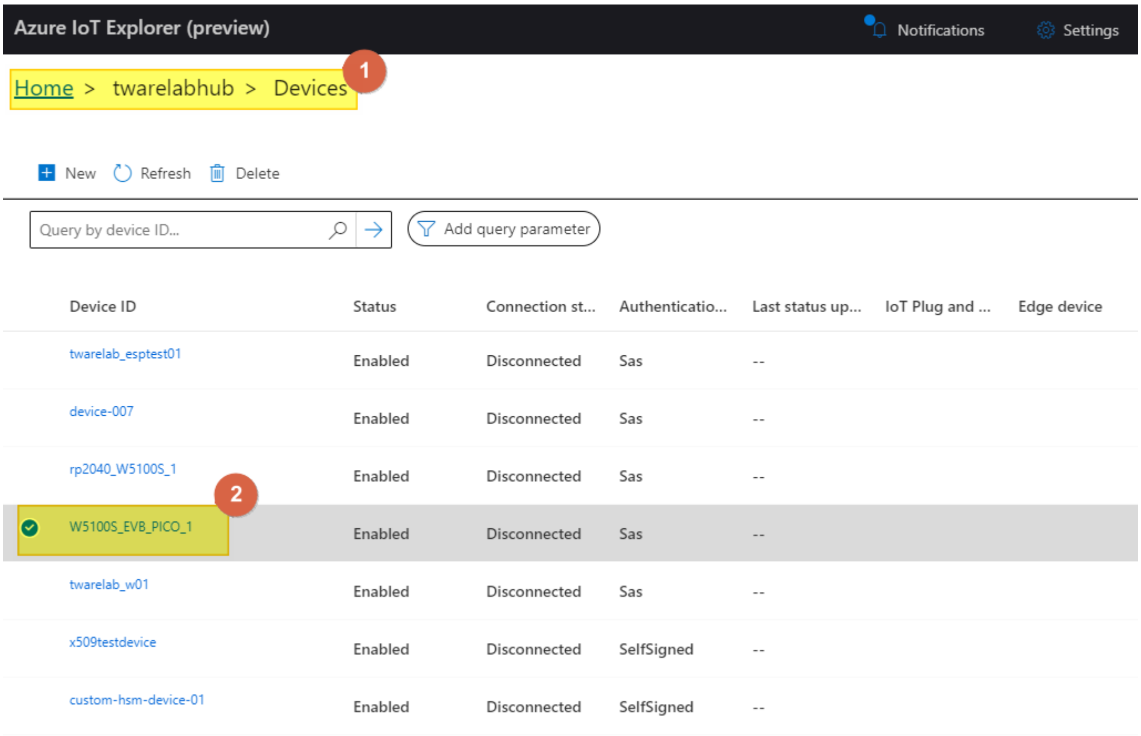
iothubowner

Shared access policy key

••

Figure 8. Add connection string

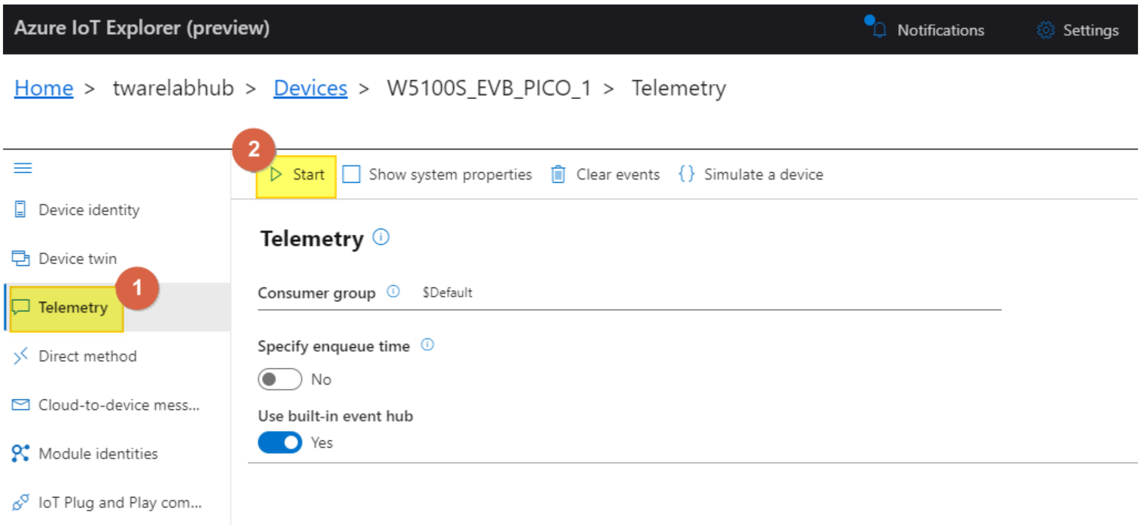
3. Find the device and click name.



Device ID	Status	Connection st...	Authenticatio...	Last status up...	IoT Plug and ...	Edge device
twarelab_esptest01	Enabled	Disconnected	Sas	--		
device-007	Enabled	Disconnected	Sas	--		
rp2040_W5100S_1	Enabled	Disconnected	Sas	--		
✓ W5100S_EVB_PICO_1	Enabled	Disconnected	Sas	--		
twarelab_w01	Enabled	Disconnected	Sas	--		
x509testdevice	Enabled	Disconnected	SelfSigned	--		
custom-hsm-device-01	Enabled	Disconnected	SelfSigned	--		

Figure 9. Select the device

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



Telemetry

Consumer group $Default

Specify enqueue time

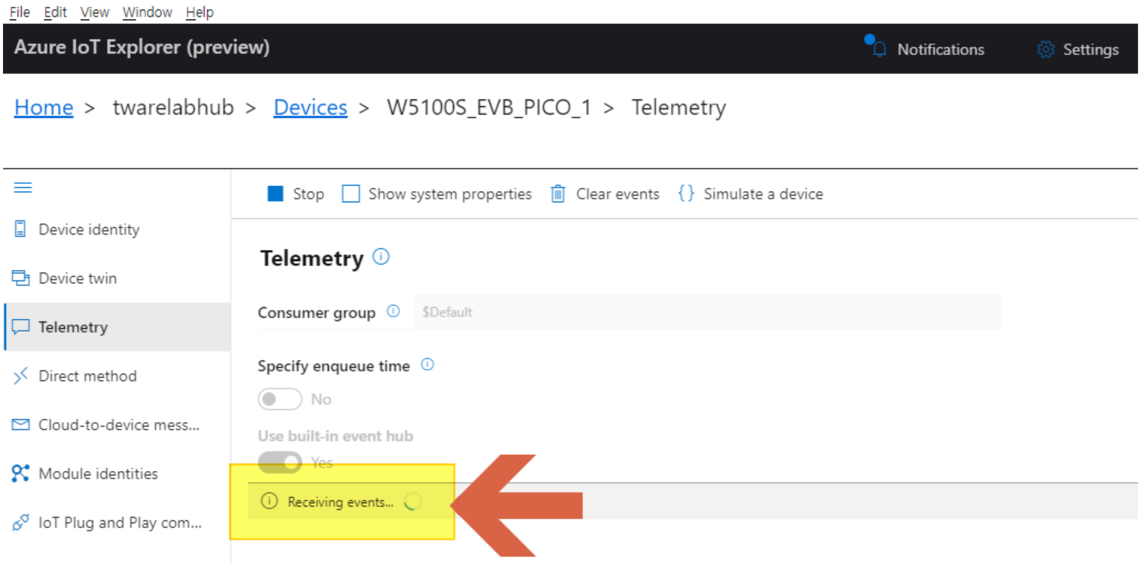
No

Use built-in event hub

Yes

Figure 10. Start Telemetry

5. Wait for incoming message from your IoT device.



Consumer group: $Default

Specify enqueue time: No

Use built-in event hub: Yes

Receiving events...

Figure 11. Receiving events

4.5 Step 5: Build

1. After completing the AZURE telemetry 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.

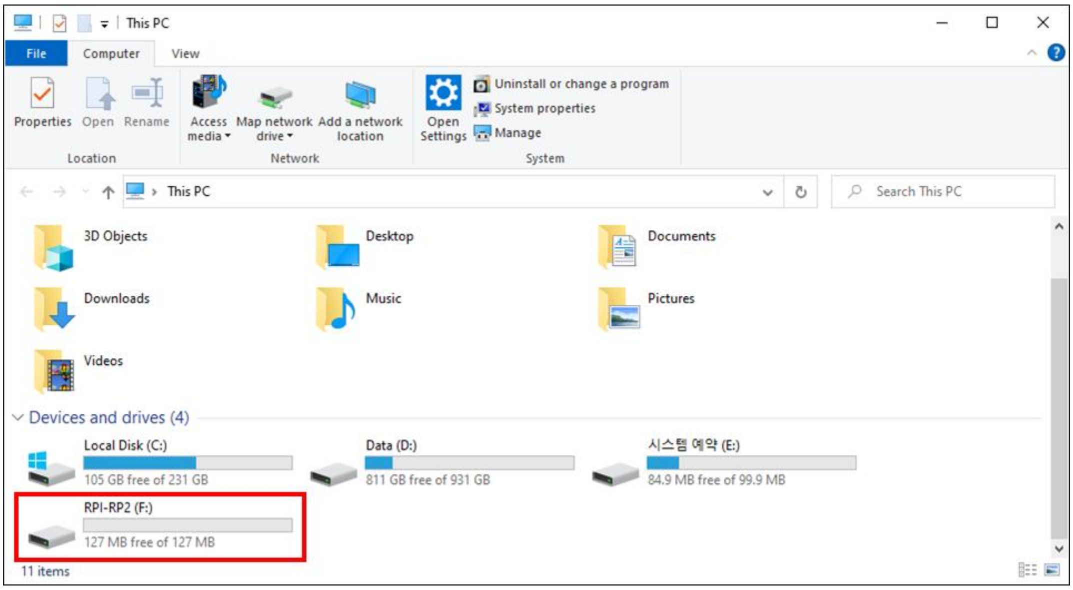
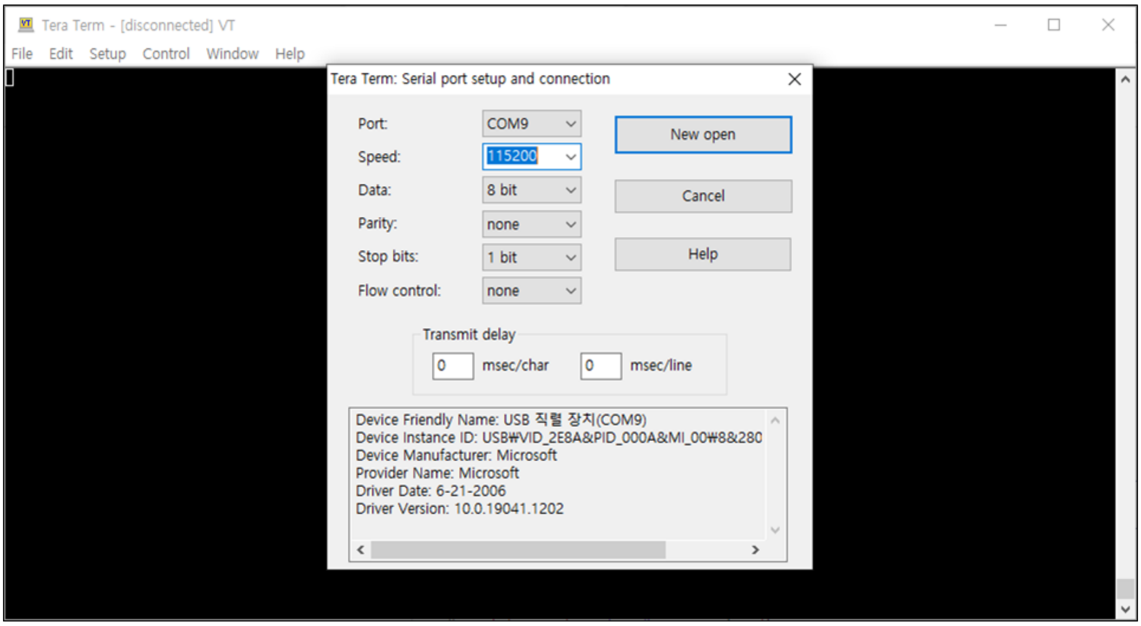


Figure 12. USB mass storage

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.


Tera Term - [disconnected] VT
File
Edit
Setup
Control
Window
Help
Tera Term: Serial port setup and connection
New open
Cancel
Help

Port: COM9
Speed: 115200
Data: 8 bit
Parity: none
Stop bits: 1 bit
Flow control: none

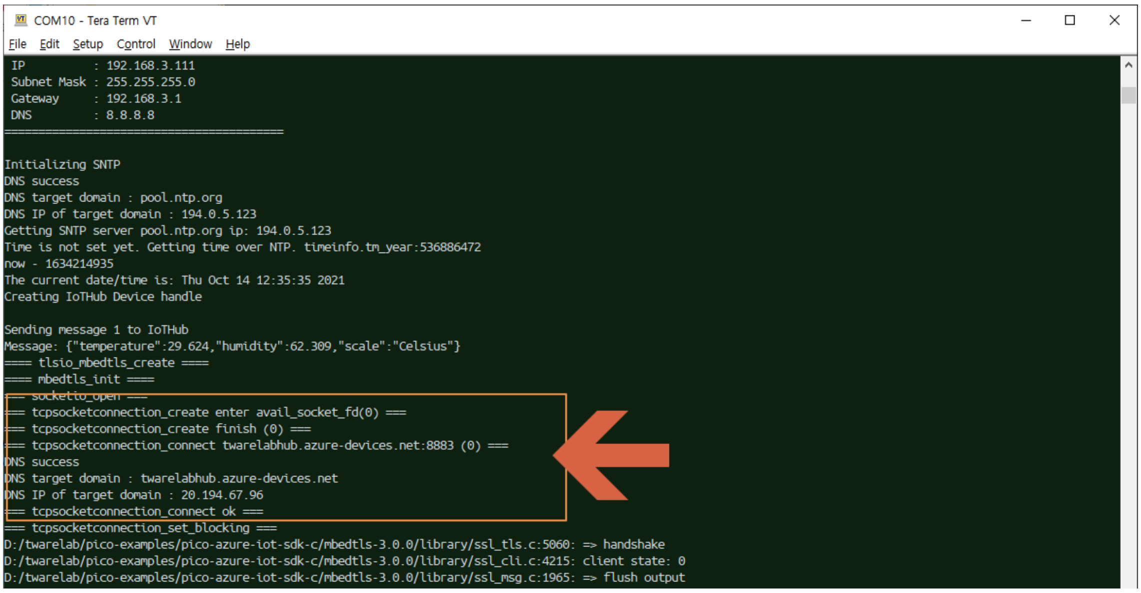
Transmit delay
0 msec/char 0 msec/line

```
Device Friendly Name: USB 직렬 장치(COM9)
Device Instance ID: USB#VID_2E8A&PID_000A&MI_00#8&280
Device Manufacturer: Microsoft
Provider Name: Microsoft
Driver Date: 6-21-2006
Driver Version: 10.0.19041.1202
```

Figure 13. Tera Term

4. Reset your board.

5. If the Azure telemetry 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.

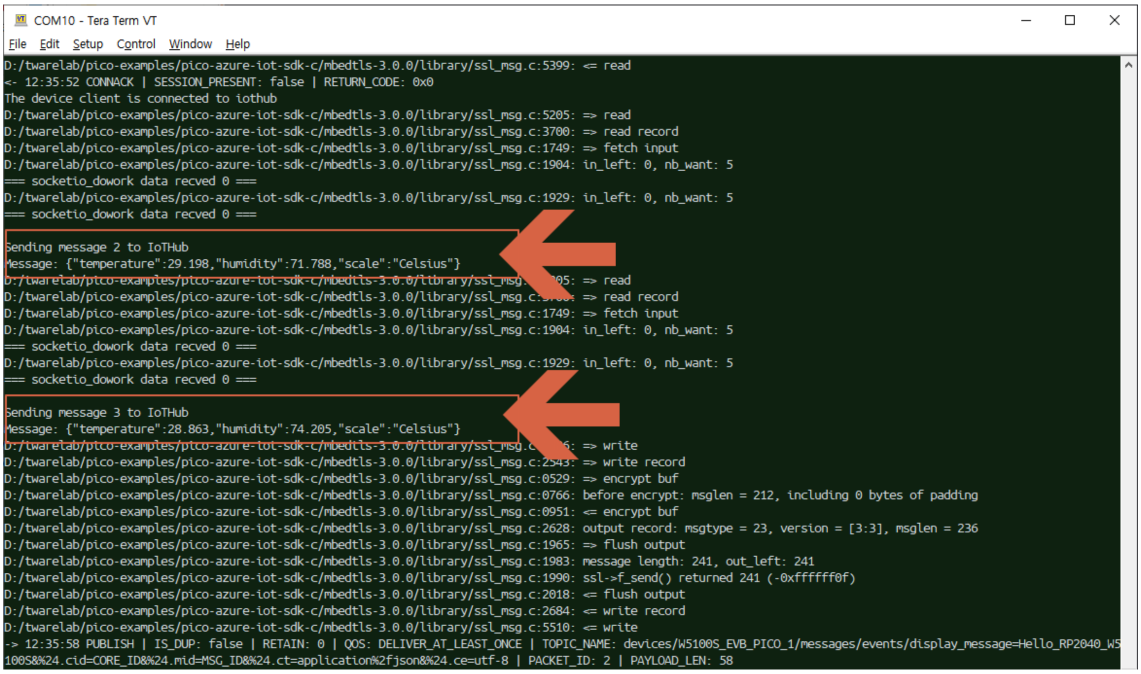


```
 IP          : 192.168.3.111
 Subnet Mask : 255.255.255.0
 Gateway     : 192.168.3.1
 DNS         : 8.8.8.8
=========================================

Initializing SNTP
DNS success
DNS target domain : pool.ntp.org
DNS IP of target domain : 194.0.5.123
Getting SNTP server pool.ntp.org ip: 194.0.5.123
Time is not set yet. Getting time over NTP. timeinfo.tm_year:536886472
now - 1634214935
The current date/time is: Thu Oct 14 12:35:35 2021
Creating IoTHub Device handle

Sending message 1 to IoTHub
Message: {"temperature":29.624,"humidity":62.309,"scale":"Celsius"}
==== tlsio_mbedtls_create ====
==== mbedtls_init ====
=== socketio_open ===
=== tcpsocketconnection_create enter avail_socket_fd(0) ===
=== tcpsocketconnection_create finish (0) ===
=== tcpsocketconnection_connect twarelabhub.azure-devices.net:8883 (0) ===
DNS success
DNS target domain : twarelabhub.azure-devices.net
DNS IP of target domain : 20.194.67.96
=== tcpsocketconnection_connect ok ===
=== tcpsocketconnection_set_blocking ===
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-3.0.0/library/ssl_tls.c:5060: => handshake
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-3.0.0/library/ssl_cli.c:4215: client state: 0
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-3.0.0/library/ssl_msg.c:1965: => flush output
```

Figure 14. Network Info and connect to Azure IoT Hub



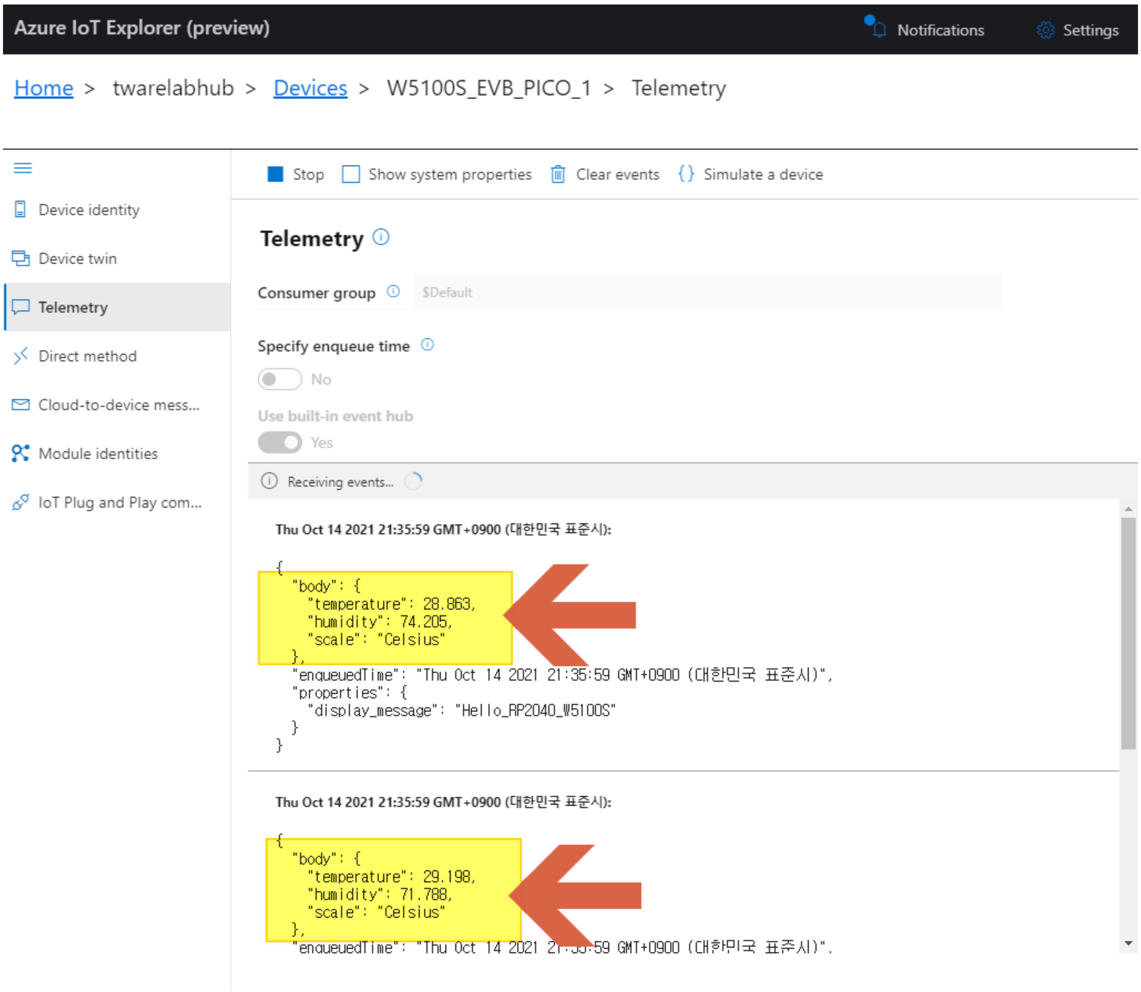
```
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-3.0.0/library/ssl_msg.c:5399: <= read
<- 12:35:52 CONNACK | SESSION_PRESENT: false | RETURN_CODE: 0x0
The device client is connected to iothub
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-3.0.0/library/ssl_msg.c:5205: => read
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-3.0.0/library/ssl_msg.c:3700: => read record
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-3.0.0/library/ssl_msg.c:1749: => fetch input
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-3.0.0/library/ssl_msg.c:1904: in_left: 0, nb_want: 5
=== socketio_dowork data recved 0 ===
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-3.0.0/library/ssl_msg.c:1929: in_left: 0, nb_want: 5
=== socketio_dowork data recved 0 ===

Sending message 2 to IoTHub
Message: {"temperature":29.198,"humidity":71.788,"scale":"Celsius"}
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-3.0.0/library/ssl_msg.[illegible]05: => read
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-3.0.0/library/ssl_msg.c:[illegible]: => read record
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-3.0.0/library/ssl_msg.c:1749: => fetch input
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-3.0.0/library/ssl_msg.c:1904: in_left: 0, nb_want: 5
=== socketio_dowork data recved 0 ===
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-3.0.0/library/ssl_msg.c:1929: in_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/mbedtls-3.0.0/library/ssl_msg.c[illegible]6: => write
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-3.0.0/library/ssl_msg.c:2543: => write record
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-3.0.0/library/ssl_msg.c:0529: => encrypt buf
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-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/mbedtls-3.0.0/library/ssl_msg.c:0951: <= encrypt buf
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-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/mbedtls-3.0.0/library/ssl_msg.c:1965: => flush output
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-3.0.0/library/ssl_msg.c:1983: message length: 241, out_left: 241
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-3.0.0/library/ssl_msg.c:1990: ssl->f_send() returned 241 (-0xffffff0f)
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-3.0.0/library/ssl_msg.c:2018: <= flush output
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-3.0.0/library/ssl_msg.c:2684: <= write record
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbedtls-3.0.0/library/ssl_msg.c:5510: <= write
-> 12:35:58 PUBLISH | IS_DUP: false | RETAIN: 0 | QOS: DELIVER_AT_LEAST_ONCE | TOPIC_NAME: devices/W5100S_EVB_PICO_1/messages/events/display_message=Hello_RP2040_W5
100S&%24.cid=CORE_ID&%24.mid=MSG_ID&%24.ct=application%2fjson&%24.ce=utf-8 | PACKET_ID: 2 | PAYLOAD_LEN: 58
```

Figure 15. Send messages to Azure IoT Hub

6. From the Azure IoT Hub configured in Step 4, you can confirm that the Raspberry Pi Pico, W5100S-EVB-Pico, W5500-EVB-Pico, W55RP20-EVB-Pico, W5100S-EVB-Pico2, or W5500-EVB-Pico2 has sent a message to the Azure IoT Hub.



Telemetry

Consumer group: $Default

Specify enqueue time: No

Use built-in event hub: Yes

Receiving events...

```
Thu Oct 14 2021 21:35:59 GMT+0900 (대한민국 표준시):

{
  "body": {
    "temperature": 28.863,
    "humidity": 74.205,
    "scale": "Celsius"
  },
  "enqueuedTime": "Thu Oct 14 2021 21:35:59 GMT+0900 (대한민국 표준시)",
  "properties": {
    "display_message": "Hello_RP2040_W5100S"
  }
}

Thu Oct 14 2021 21:35:59 GMT+0900 (대한민국 표준시):

{
  "body": {
    "temperature": 29.198,
    "humidity": 71.788,
    "scale": "Celsius"
  },
  "enqueuedTime": "Thu Oct 14 2021 21:35:59 GMT+0900 (대한민국 표준시)",
```

Figure 16. Getting device messages from Azure IoT Hub

Revision history

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

Table 1. Revision history

Copyright Notice

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

Sales & Distribution: sales@wiznet.io

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