

# Application Note

## AZURE\_client\_X.509

### 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>3</b>
<b>2 Github Link</b> .....	<b>3</b>
<b>3 Applicable products</b> .....	<b>3</b>
<b>4 How to Test AZURE client X.509 Example</b> .....	<b>3</b>
4.1 Step 1: Prepare software .....	3
4.2 Step 2: Prepare hardware .....	3
4.3 Step 3: Setup AZURE client X.509 Example.....	4
4.4 Step 4: Setup Device self-signed certificates .....	6
4.5 Step 5: Setup Azure IoT Explorer .....	9
4.6 Step 6: Build .....	13
4.7 Step 7: Upload and Run .....	13
<b>Revision history</b> .....	<b>18</b>

## Figures

FIGURE 1. ADD IOT DEVICES.....	9
FIGURE 2. CREATE A DEVICE WITH X.509 SELF-SIGNED .....	10
FIGURE 3. REFRESH AND CHECK THE DEVICE .....	10
FIGURE 4. CHECK THE DEVICE.....	11
FIGURE 5. SELECT THE DEVICE .....	11
FIGURE 6. START TELEMETRY .....	12
FIGURE 7. RECEIVING EVENTS.....	12
FIGURE 8. USB MASS STORAGE.....	13
FIGURE 9. TERA TERM .....	14
FIGURE 10. START TO VERIFY THE DEVICE .....	15
FIGURE 11. SEND MESSAGES TO AZURE IOT HUB .....	16
FIGURE 12. GETTING DEVICE MESSAGES FROM AZURE IOT HUB.....	17

## Tables

TABLE 1. REVISION HISTORY .....	18
---------------------------------	----

## 1 Introduction

This Application Note covers the implementation of AZURE client X.509 authentication and the generation of X.509 certificates 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 client X.509 Example

### 4.1 Step 1: Prepare software

The following serial terminal program is required for AZURE client X.509 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 client X.509 Example

To test the AZURE client X.509 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 client X.509 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\_CLI\_X509 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 client X.509 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
};
```

## 4.4 Step 4: Setup Device self-signed certificates

Please follow up [tutorial-x509-self-sign](#).

1. For your reference, prepare example log as below:

Notice! device ID = "W5100S\_EVB\_PICO\_X509"

- Provide the Device ID that matches the subject name of your two certificates. In this example, "W5100S\_EVB\_PICO\_X509"
- Select the X.509 Self-Signed authentication type.
- Paste the hex string thumbprints that you copied from your device primary and secondary certificates. Make sure that the hex strings have no colon delimiters.

```
MINGW64 ~
$ mkdir certi

MINGW64 ~
$ cd certi/

MINGW64 ~/certi
$ openssl genpkey -out device1.key -algorithm RSA -pkeyopt
rsa_keygen_bits:2048
.....+++++
.....+++++

MINGW64 ~/certi
$ openssl req -new -key device1.key -out device1.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:.
State or Province Name (full name) [Some-State]:.
Locality Name (eg, city) []:.
Organization Name (eg, company) [Internet Widgits Pty Ltd]:.
Organizational Unit Name (eg, section) []:.
Common Name (e.g. server FQDN or YOUR name) []:W5100S_EVB_PICO_X509
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

MINGW64 ~/certi
$ openssl req -text -in device1.csr -noout
Certificate Request:
Data:
  Version: 1 (0x0)
  Subject: CN = W5100S_EVB_PICO_X509
  Subject Public Key Info:
```

```

Public Key Algorithm: rsaEncryption
RSA Public-Key: (2048 bit)
Modulus:
    00:bb:ba:cb:62:7a:ce:ac:4d:ff:88:c7:1a:ad:6a:
    b4:6b:83:cc:30:74:94:7b:d2:8c:ed:6f:37:bf:c2:
...
    ff:17:35:fb:78:d8:a8:31:04:a6:dd:89:f5:d6:fd:
    a2:8e:e2:b3:62:d4:96:f2:9b:80:b5:22:4a:e2:6f:
    88:e3
Exponent: 65537 (0x10001)
Attributes:
    a0:00
Signature Algorithm: sha256WithRSAEncryption
    7e:de:0e:58:a6:44:c4:a6:76:12:be:a5:e0:80:35:90:ec:cb:
...
    73:ca:29:5f:36:d9:cd:1c:1e:34:98:c3:9a:a8:93:ef:28:f4:
    a9:45:f9:4e

MINGW64 ~/certi
$ openssl x509 -req -days 365 -in device1.csr -signkey device1.key -out
device1.crt
Signature ok
subject=CN = W5100S_EVB_PICO_X509
Getting Private key

MINGW64 ~/certi
$ openssl genpkey -out device2.key -algorithm RSA -pkeyopt
rsa_keygen_bits:2048
.....
.....+++++
.....+++++

MINGW64 ~/certi
$ openssl req -new -key device2.key -out device2.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:.
State or Province Name (full name) [Some-State]:.
Locality Name (eg, city) []:.
Organization Name (eg, company) [Internet Widgits Pty Ltd]:.
Organizational Unit Name (eg, section) []:.
Common Name (e.g. server FQDN or YOUR name) []:W5100S_EVB_PICO_X509
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

MINGW64 ~/certi
$ openssl x509 -req -days 365 -in device2.csr -signkey device2.key -out
device2.crt
Signature ok

```

```
subject=CN = W5100S_EVB_PICO_X509
Getting Private key
```

```
MINGW64 ~/certi
$ openssl x509 -in device1.crt -noout -fingerprint
SHA1 Fingerprint=F3:61:90:1F:B5:76:xx:xx:xx:xx:9B:51:4F:51
```

```
MINGW64 ~/certi
$ openssl x509 -in device2.crt -noout -fingerprint
SHA1 Fingerprint=09:75:4F:7F:14:xx:xx:xx:xx:38:8B:D5:0D
```

```
MINGW64 ~/certi
$
```

2. Get the key value from files (device1.crt, device1.key). And edit 'WIZnet-PICO-AZURE-C/exmaples/sample\_certs.c' with generated certificates as upper.  
Connection string for this example is "HostName=twarelabhub.azure-devices.net;DeviceId=W5100S\_EVB\_PICO\_X509;x509=true"

```
#include "azure_samples.h"

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

const char pico_az_x509connectionString[] = "HostName=my-rp2040-hub.azure-devices.net;DeviceId=my-rp2040-device-cli-x509;x509=true";

const char pico_az_x509certificate[] =
"-----BEGIN CERTIFICATE-----" "\n"
"MIIDrTCCApUCFEjR3/7wNgnU0qY5hxGBR92pVjZ3MA0GCSqGSIb3DQEBCwUAMIGS" "\n"
"MQswCQYDVQQGEWJLUjEUMBIGA1UECAwLR3l1b25nZ2ktZG8xFDASBgNVBACMC1N1" "\n"
"...
"v7wvi4IZvXDFtF+CIE8L3Ym13V+gp2ZJhA7eeeY0BHgr0fcNqCEJScQTopZNFzjA" "\n"
"OgWA3VyB8jR6Pxx5DmLwsFm0aYnu+f6xA1lHJs+xeajb" "\n"
"-----END CERTIFICATE-----";

const char pico_az_x509privatekey[] =
"-----BEGIN PRIVATE KEY-----" "\n"
"MIIEvAIBADANBgkqhkiG9w0BAQEFAASCByYggSiAgEAAoIBAQDAekFjSy6DRyxI" "\n"
"B7nSN8znN3Ki9iZM066Zm8VVmm/LRk+TqZ1kfGTS97SzdAX7xuQDCJG0vqlyd+BP" "\n"
"...
"w6ffc61aVKczE4xiVdIcUh5lOFTK9gi9pOuHvPDHy9ilWGmmetric/bFRmHcjlW7I" "\n"
"o4rWl809TIKUL0ViCDsGSg==" "\n"
"-----END PRIVATE KEY-----";
```



## 4.5 Step 5: Setup Azure IoT Explorer

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

1. Add device in your Azure IoT Hub.

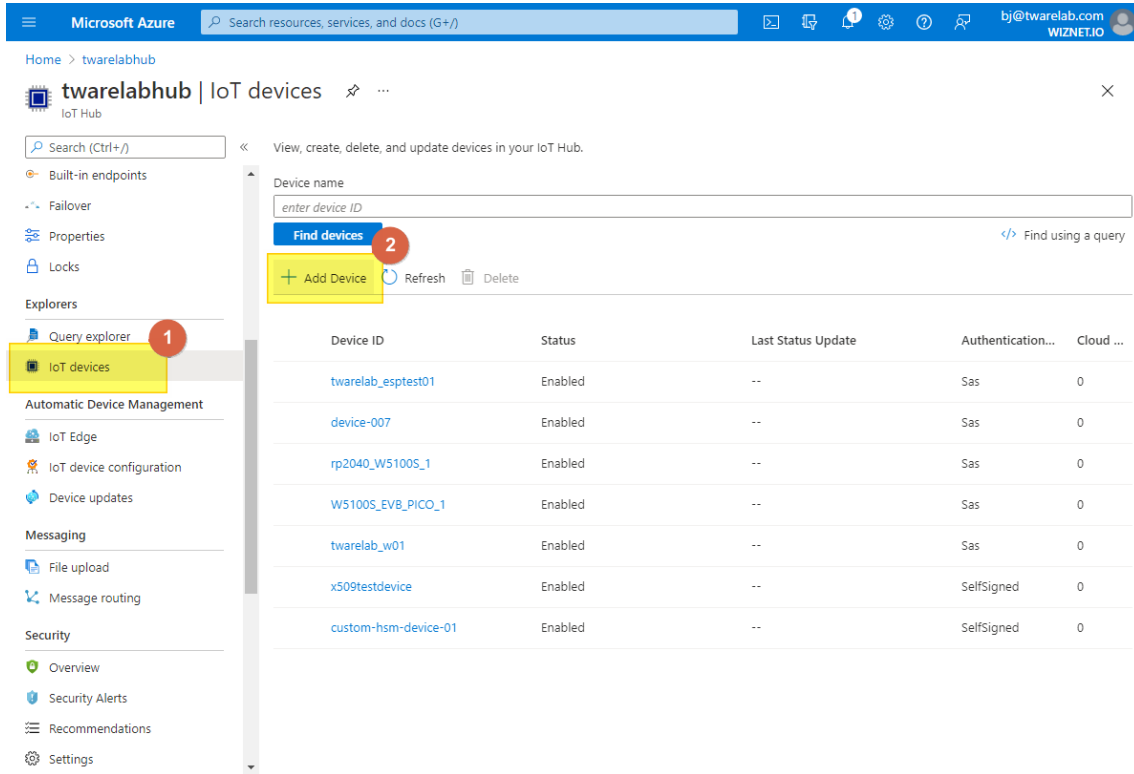


Figure 1. Add IoT devices

2. Create a device with X.509 Self-Signed. Enter the fingerprint of the crt file obtained in Step 4.

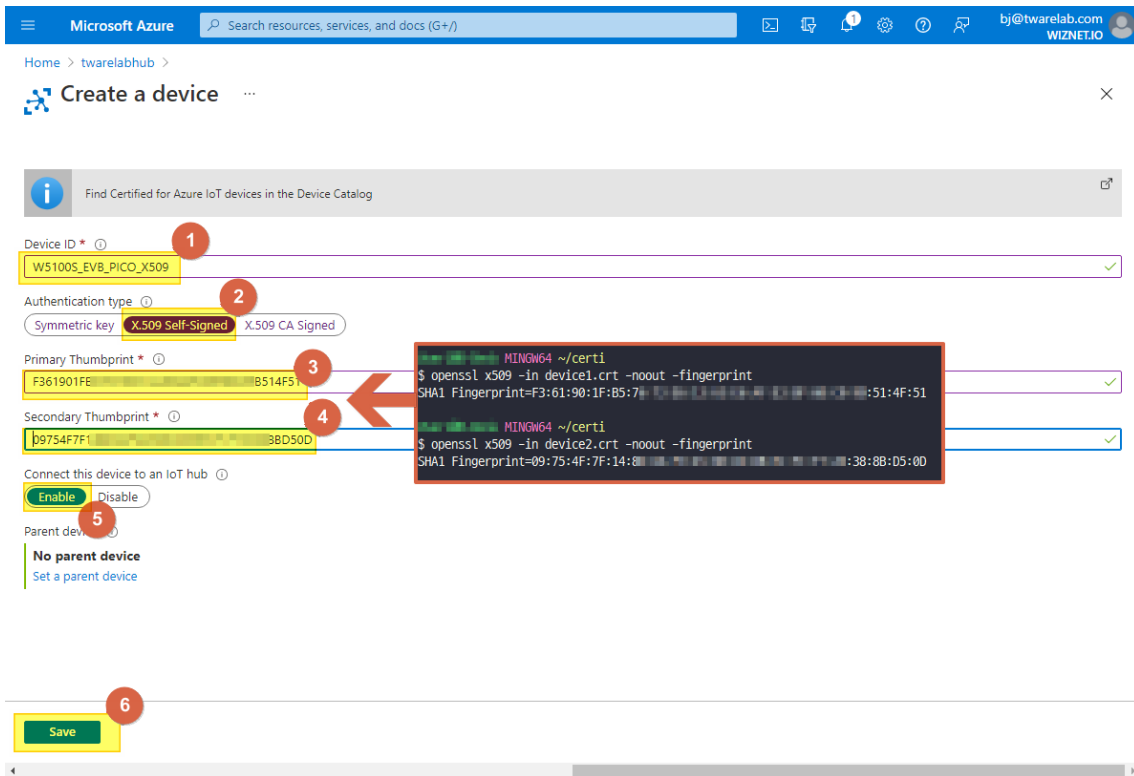


Figure 2. Create a device with X.509 Self-Signed

3. Check the device in the 'device list'.

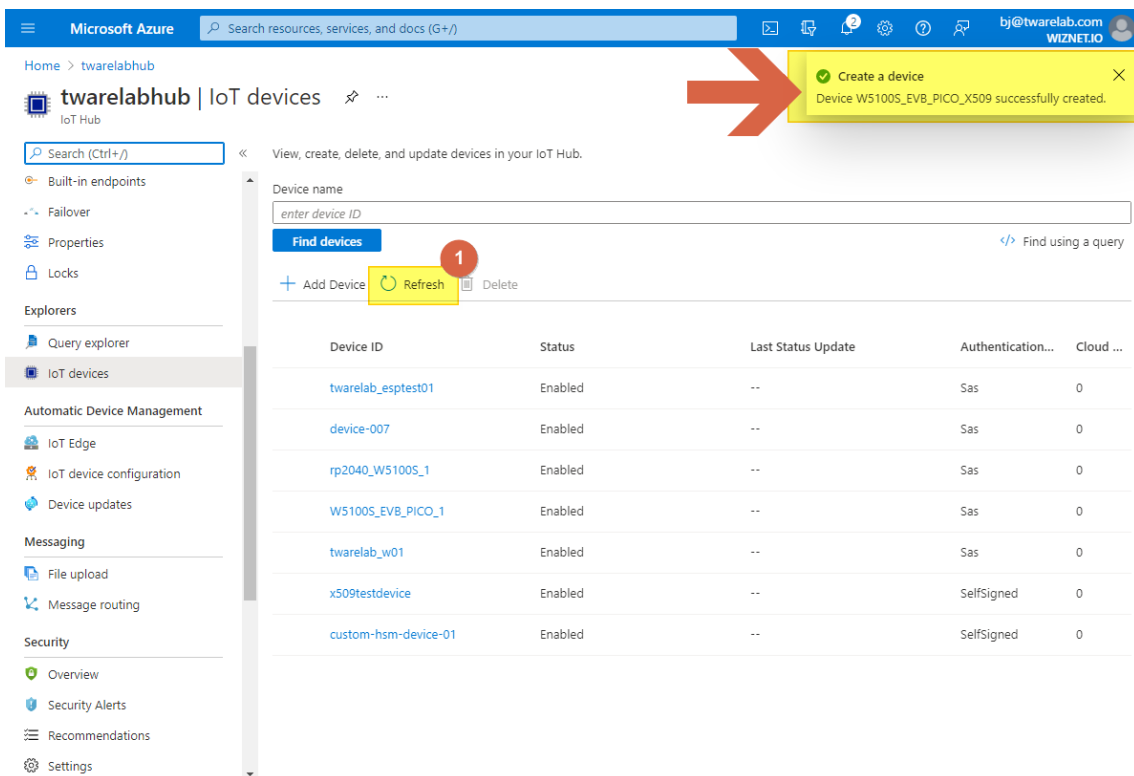


Figure 3. Refresh and Check the device

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
<input checked="" type="checkbox"/> W5100S_EVB_PICO_X509	Enabled	--	SelfSigned	0

Figure 4. Check the device

4. Click the device name created in the previous section.

Azure IoT Explorer (preview) Notifications Settings

Home > twarelabhub > Devices 1

+ New Refresh Delete

Query by device ID... Add query parameter

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	--		
<input type="radio"/> W5100S_EVB_PICO_X509	Enabled	Disconnected	SelfSigned	--		

Figure 5. Select the device

5. Go to "Telemetry" menu and click "Start".

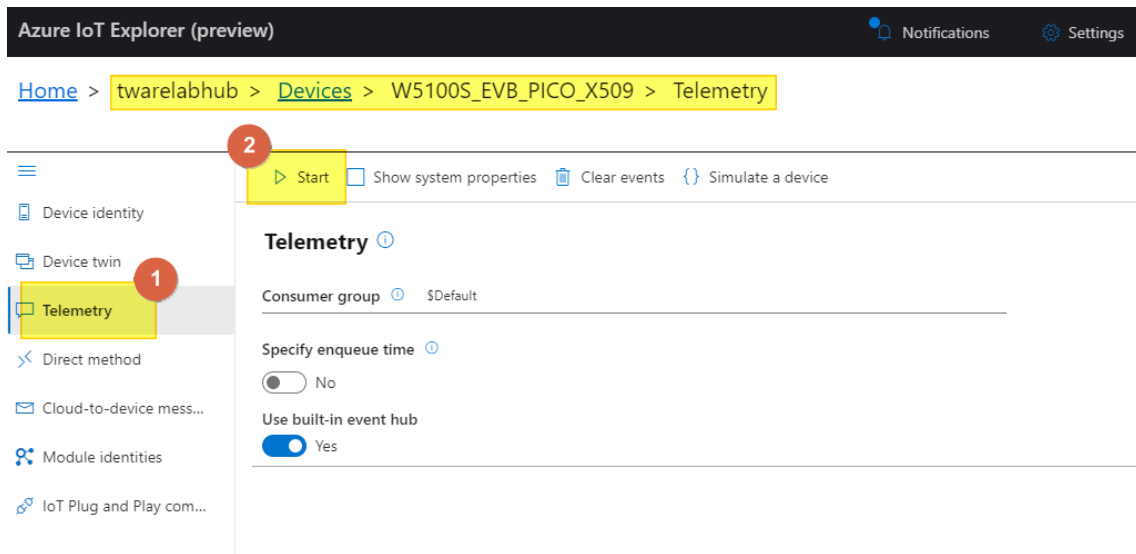


Figure 6. Start Telemetry

6. Wait for incoming messages.

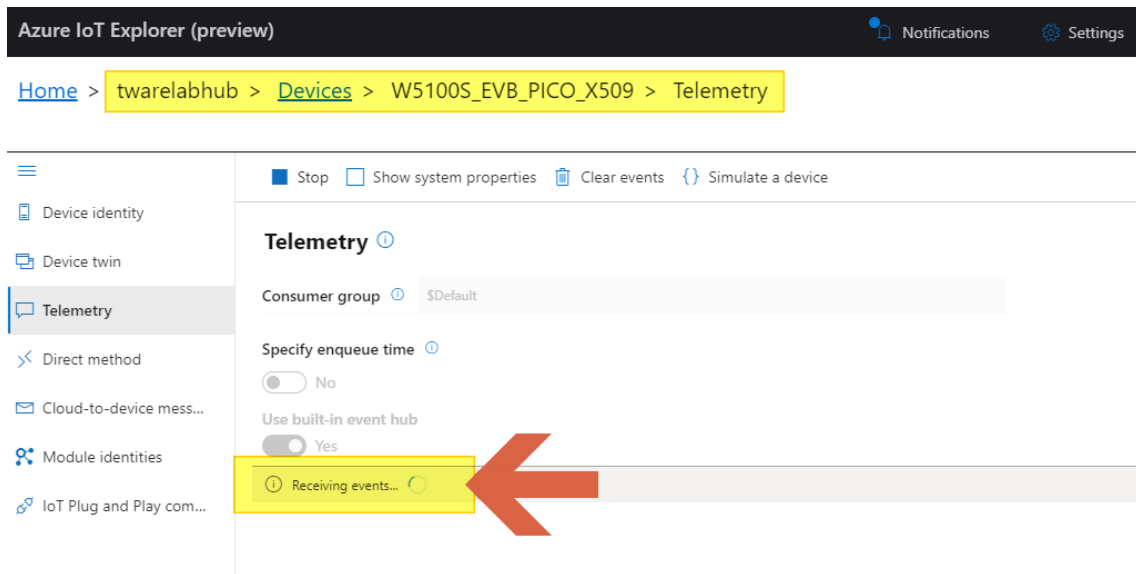


Figure 7. Receiving events

## 4.6 Step 6: Build

1. After completing the AZURE client X.509 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.

## 4.7 Step 7: 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.

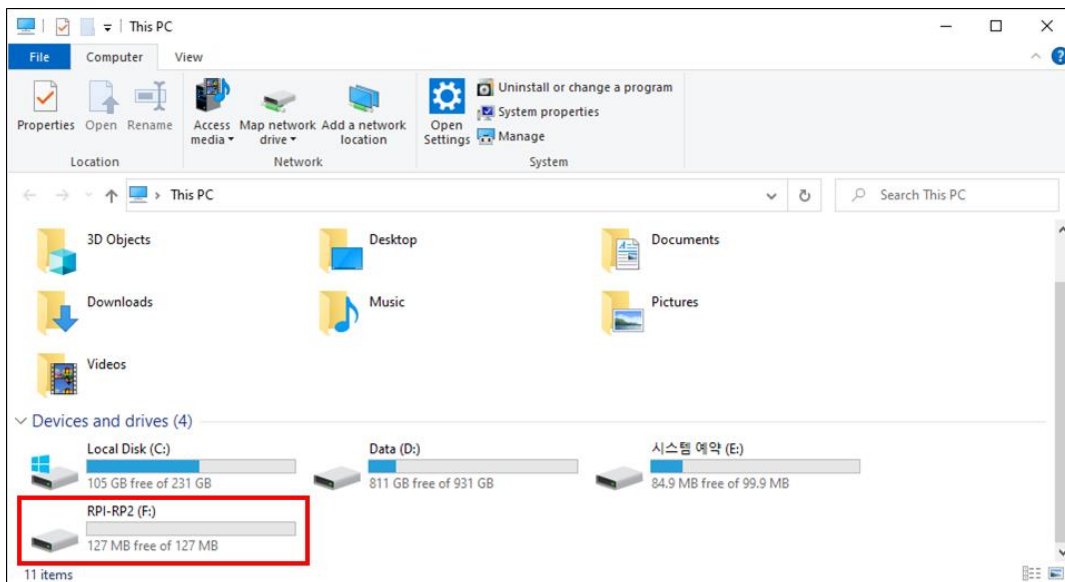


Figure 8. USB mass storage

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.

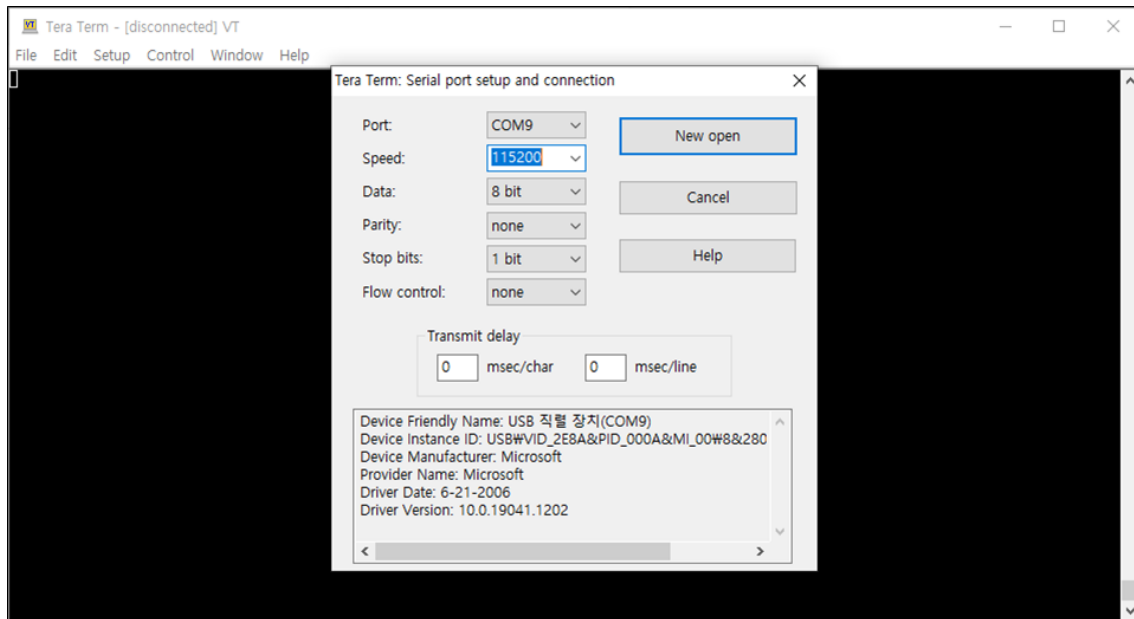
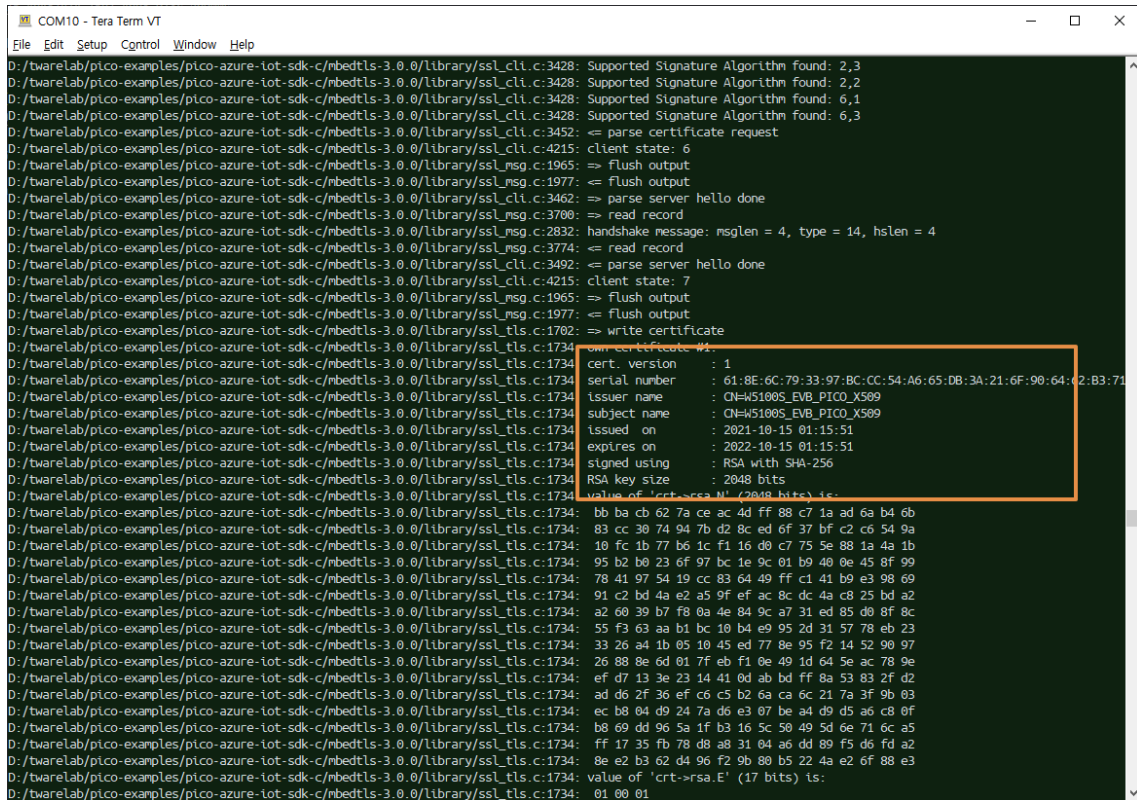


Figure 9. Tera Term

4. Reset your board.

- If the Azure client X.509 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 start to verify the device with X.509 authentication.



```

COM10 - Tera Term VT
File Edit Setup Control Window Help
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_cli.c:3428: Supported Signature Algorithm found: 2,3
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_cli.c:3428: Supported Signature Algorithm found: 2,2
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_cli.c:3428: Supported Signature Algorithm found: 6,1
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_cli.c:3428: Supported Signature Algorithm found: 6,3
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_cli.c:3452: => parse certificate request
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_cli.c:4215: client state: 6
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:1965: => flush output
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:1977: => flush output
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_cli.c:3462: => parse server hello done
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:3700: => read record
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:2832: handshake message: msglen = 4, type = 14, hslen = 4
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:3774: => read record
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_cli.c:3492: => parse server hello done
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_cli.c:4215: client state: 7
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:1965: => flush output
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_msg.c:1977: => flush output
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1762: => write certificate
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: Certificate #1:
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: cert. version      : 1
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: serial number     : 61:8E:6C:79:33:97:BC:CC:54:A6:65:DB:3A:21:6F:90:64:02:B3:71
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: issuer name       : CN=W5100S_EVB_PICO_X509
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: subject name      : CN=W5100S_EVB_PICO_X509
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: issued on        : 2021-10-15 01:15:51
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: expires on       : 2022-10-15 01:15:51
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: signed using     : RSA with SHA-256
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: RSA key size     : 2048 bits
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: value of 'cert->rsa.N' (2048 bits) is:
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: bb ba cb 62 7a ce ac 4d ff 88 c7 1a ad 6a b4 6b
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: 83 cc 30 74 94 7b d2 8c ed 6f 37 bf c2 c6 54 9a
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: 10 fc 1b 77 b6 1c f1 16 d0 c7 75 5e 88 1a 4a 1b
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: 95 b2 b0 23 6f 97 bc 1e 9c 01 b9 40 0e 45 8f 99
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: 78 41 97 54 19 cc 83 64 49 ff c1 41 b9 e3 98 69
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: 91 c2 bd 4a e2 a5 9f ef ac 8c dc 4a c8 25 bd a2
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: a2 60 39 b7 f8 0a 4e 84 9c a7 31 ed 85 d0 8f 8c
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: 55 f3 63 aa b1 bc 10 b4 e9 95 2d 31 57 78 eb 23
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: 33 26 a4 1b 05 10 45 ed 77 8e 95 f2 14 52 90 97
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: 26 88 0e 6d 01 7f eb f1 0e 49 1d 64 5e ac 78 9e
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: ef d7 13 3e 23 14 41 0d ab bd ff 8a 53 83 2f d2
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: ad d6 2f 36 ef c6 c5 b2 6a ca 6c 21 7a 3f 9b 03
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: ec b8 04 d9 24 7a d6 e3 07 be a4 d9 d5 a6 c8 0f
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: b8 69 dd 96 5a 1f b3 16 5c 59 49 5d 6e 71 6c a5
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: ff 17 35 fb 78 d8 a8 31 04 a6 dd 89 f5 d6 fd a2
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: 8e e2 b3 62 d4 96 f2 9b 80 b5 22 4a e2 6f 88 e3
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: value of 'cert->rsa.E' (17 bits) is:
D:/twarelab/pico-examples/pico-azure-tot-sdk-c/nbedt1s-3.0.0/library/ssl_tls.c:1734: 01 00 01
  
```

Figure 10. Start to verify the device

6. After completing the X.509 authentication verification, proceed to send messages to Azure IoT Hub.

```
COM10 - Tera Term VT
File Edit Setup Control Window Help
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
-> 02:28:32 CONNACK | SESSION_PRESENT: false | RETURN_CODE: 0x0
The device client is connected to tothub
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: in_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: in_left: 0, nb_want: 5
== socketio_dowork data recved 0 ==

Sending message 2 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:1905: => 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: in_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: in_left: 0, nb_want: 5
== socketio_dowork data recved 0 ==

Sending message 3 to IoTHub
Message: {"temperature":22.259,"humidity":68.489,"scale":"Celsius"}
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:1906: => 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 = 215, 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 = 239
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: 244, out_left: 244
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:1990: ssl->f_send() returned 244 (-0xfffff6c)
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:2618: == 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
-> 02:28:38 PUBLISH | IS_DUP: false | RETAIN: 0 | QOS: DELIVER_AT_LEAST_ONCE | TOPIC_NAME: devices/W5100S_EVB_PICO_X509/messages/events/display_message=Hello_RP2040
_W5100S_EVB_PICO_X509/messages/events/display_message=Hello_RP2040
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 = 215, 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 = 239
D:/twarelab/pico-examples/pico-azure-iot-sdk-c/mbdrtls-3.0.0/library/ssl_msg.c:1965: => flush output
```

Figure 11. Send messages to Azure IoT Hub



## 7. You can see the incoming messages from your IoT device

The screenshot shows the Azure IoT Explorer interface. At the top, it says "Azure IoT Explorer (preview)" with "Notifications" and "Settings" icons. The breadcrumb navigation is "Home > twarelabhub > Devices > W5100S\_EVB\_PICO\_X509 > Telemetry".

On the left sidebar, the "Telemetry" option is selected. The main area shows the "Telemetry" section for the device. It includes controls for "Stop", "Show system properties", "Clear events", and "Simulate a device". The "Consumer group" is set to "\$Default". The "Specify enqueue time" toggle is set to "No". The "Use built-in event hub" toggle is set to "Yes".

Under "Receiving events...", two messages are displayed, each with a red arrow pointing to the "body" field:

```
Fri Oct 15 2021 11:28:39 GMT+0900 (대한민국 표준시):  
{  
  "body": {  
    "temperature": 22.259,  
    "humidity": 68.489,  
    "scale": "Celsius"  
  },  
  "enqueueTime": "Fri Oct 15 2021 11:28:39 GMT+0900 (대한민국 표준시)",  
  "properties": {  
    "display_message": "Hello_RP2040_W5100S"  
  }  
}
```

```
Fri Oct 15 2021 11:28:38 GMT+0900 (대한민국 표준시):  
{  
  "body": {  
    "temperature": 28.863,  
    "humidity": 74.205,  
    "scale": "Celsius"  
  },  
  "enqueueTime": "Fri Oct 15 2021 11:28:38 GMT+0900 (대한민국 표준시)".
```

Figure 12. 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

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/>