

# Application Note

# TCP\_Client\_over\_SSL

# 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 TCP Client over SSL Example</b> .....	<b>3</b>
4.1 Step 1: Prepare software .....	3
4.2 Step 2: Prepare hardware .....	3
4.3 Step 3: Setup TCP Client over SSL Example.....	4
4.4 Step 4: Build .....	5
4.5 Step 5: Upload and Run .....	5
4.6 Appendix .....	10
<b>Revision history</b> .....	<b>11</b>

## Figures

FIGURE 1. USB MASS STORAGE.....	5
FIGURE 2. TERA TERM .....	6
FIGURE 3. RUN OPENSLL .....	6
FIGURE 4. CREATE SSL SERVER USING OPENSLL .....	7
FIGURE 5. CONNECT TO SSL SERVER AND SENDING MESSAGE 1 .....	8
FIGURE 6. CONNECT TO SSL SERVER AND SENDING MESSAGE 2.....	8
FIGURE 7. RECEIVE SENT MESSAGE 1.....	9
FIGURE 8. RECEIVE SENT MESSAGE 2.....	9

## Tables

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

## 1 Introduction

This Application Note covers the implementation of TCP Client over SSL on WIZnet's TOE Chip.

## 2 Github Link

[https://github.com/WIZnet-ioNIC/WIZnet-PICO-C/tree/main/examples/tcp\\_client\\_over\\_ssl](https://github.com/WIZnet-ioNIC/WIZnet-PICO-C/tree/main/examples/tcp_client_over_ssl)

## 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 TCP Client over SSL Example

### 4.1 Step 1: Prepare software

The following serial terminal program and SSL server are required for TCP Client over SSL example test, download and install from below links.

- [Tera Term](#)
- [OpenSSL](#)

### 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 TCP Client over SSL Example

To test the TCP Client over SSL example, minor settings shall be done in code.

1. Setup SPI port and pin in 'w5x00\_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 TCP Client over SSL 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. Setup network configuration such as IP in 'w5x00\_tcp\_client\_over\_ssl.c', which is the TCP Client over SSL example in 'WIZnet-PICO-C/examples/tcp\_client\_over\_ssl/' directory.

- 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
```

```
};
    .dhcp = NETINFO_STATIC           // DHCP enable/disable
```

3. Setup TCP Client over SSL configuration in 'w5x00\_tcp\_client\_over\_ssl.c' in 'WIZnet-PICO-C/examples/tcp\_client\_over\_ssl/' directory.
  - In the TCP client over SSL configuration, the target IP is the IP of your desktop or laptop where SSL server will be created.

```
#define PORT_SSL 443

static uint8_t g_ssl_target_ip[4] = {192, 168, 11, 3};
```

In order to change SSL settings, modify 'ssl\_config.h' located in the same directory with TCP Client over SSL example.

## 4.4 Step 4: Build

1. After completing the TCP Client over SSL 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, 'w5x00\_tcp\_client\_over\_ssl.uf2' is generated in 'WIZnet-PICO-C/build/examples/tcp\_client\_over\_ssl' 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 or W5500-EVB-Pico2 power on the board, the USB mass storage 'RPI-RP2' is automatically mounted.

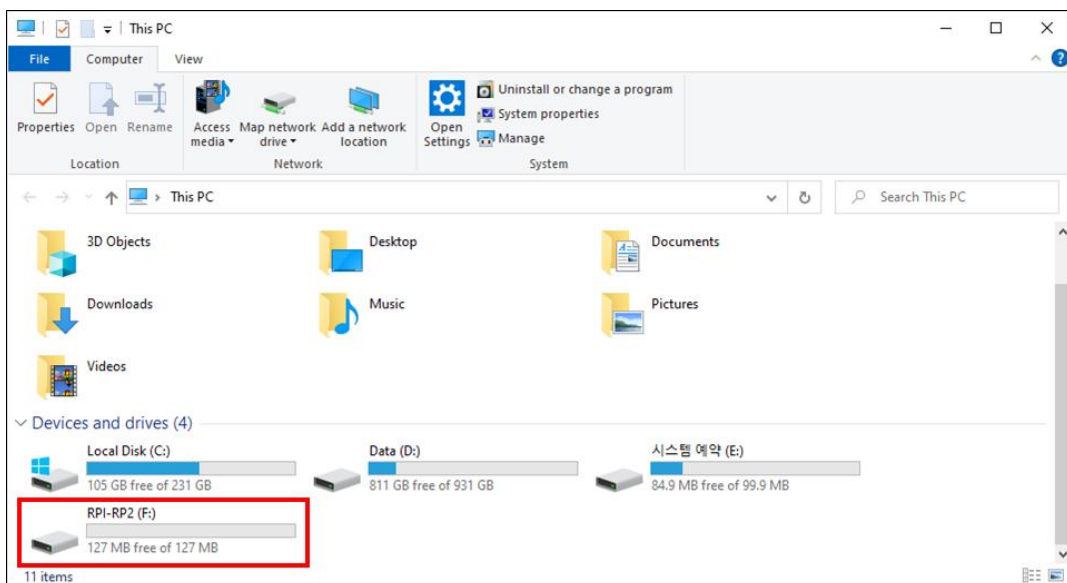


Figure 1. USB mass storage

2. Drag and drop 'w5x00\_tcp\_client\_over\_ssl.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.

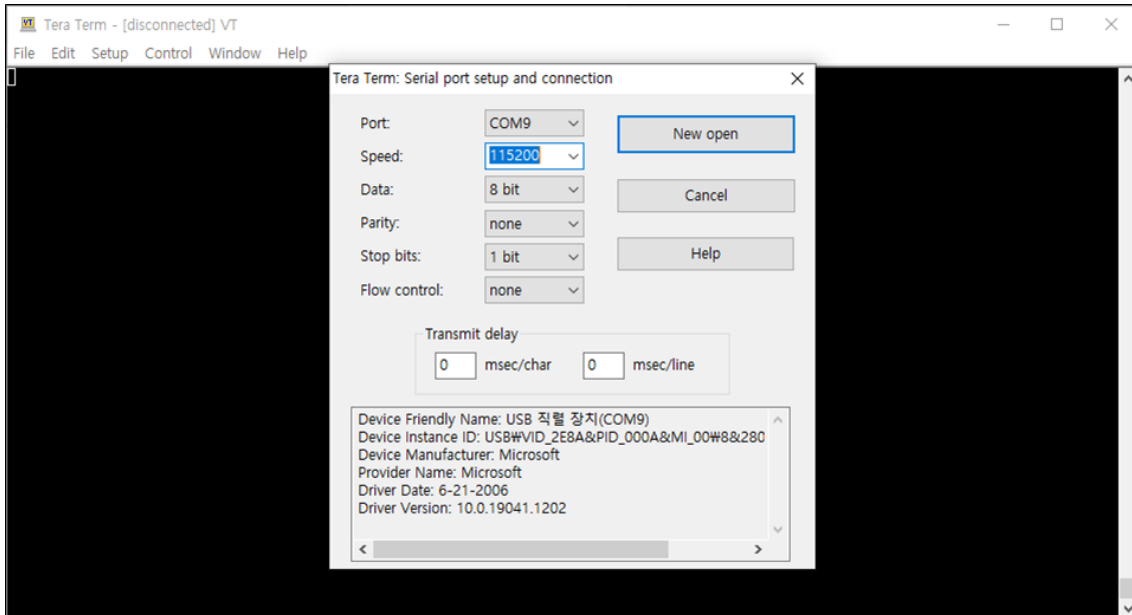


Figure 2. Tera Term

4. Run OpenSSL to be used as the SSL server.

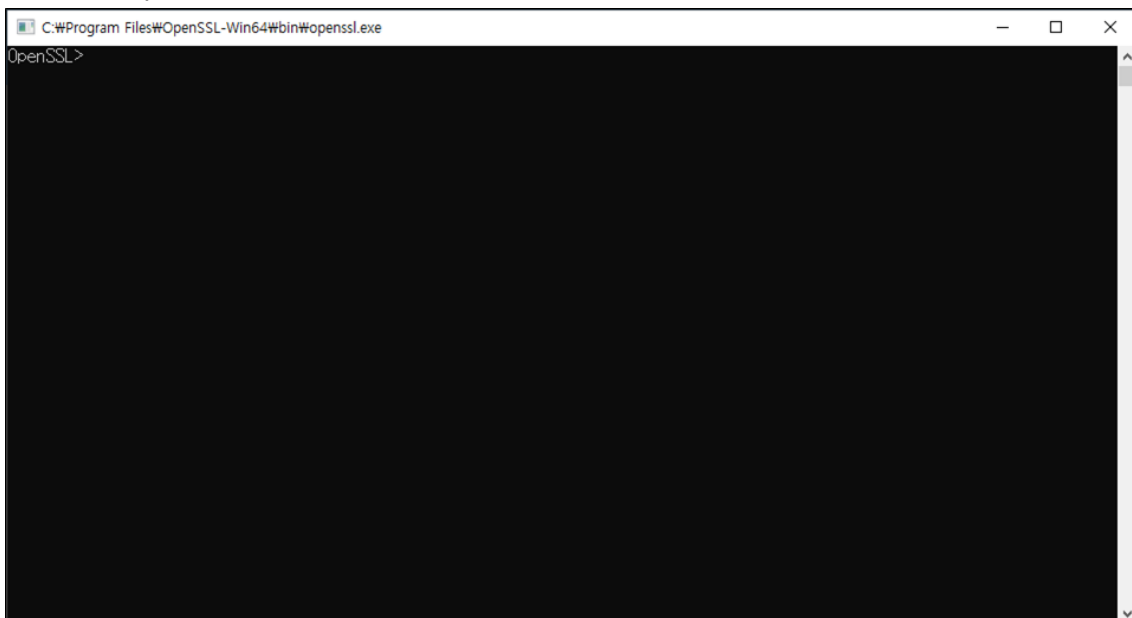


Figure 3. Run OpenSSL

5. Create SSL server using OpenSSL by executing the following command. If the SSL server is created normally, the SSL server's IP is the current IP of your desktop or laptop, and the port is 443 by default.

- Setup the SSL server

```
/* Setup the SSL server */
// create the private key
genrsa -des3 -out [key name].key 2048

// create the CSR(required for certificate signing request)
req -new -key [key name].key -out [csr name].csr

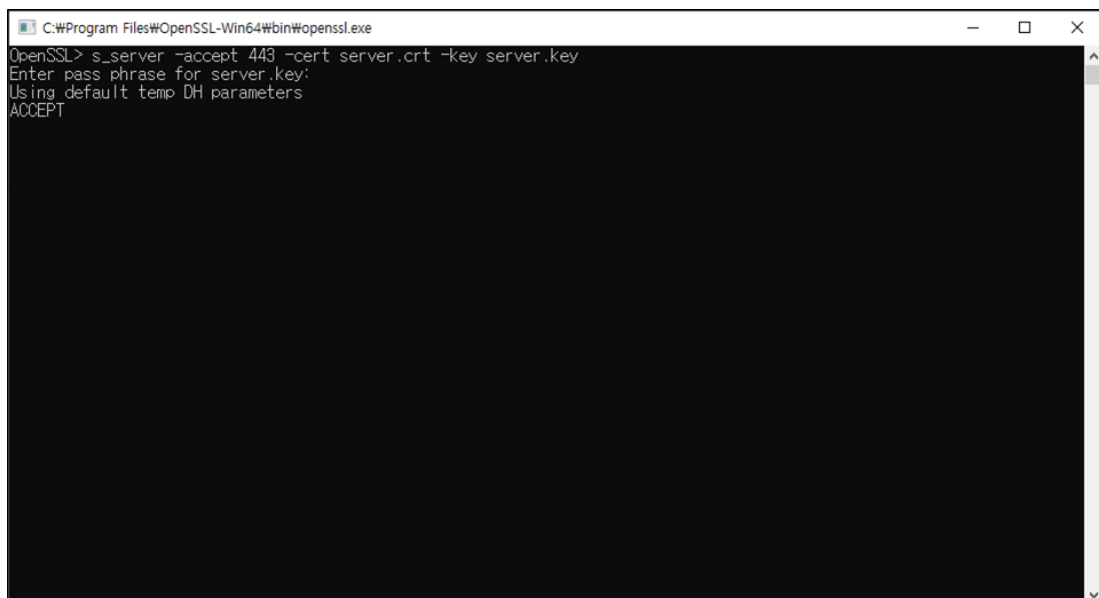
// create the certificate
x509 -req -days [days] -in [csr name].csr -signkey [key name].key -out
[crt name].crt
```

```
// e.g.
genrsa -des3 -out server.key 2048
req -new -key server.key -out server.csr
x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
```

- Run the SSL server

```
/* Run the SSL server */
s_server -accept [port] -cert [crt name].crt -key [key name].key
```

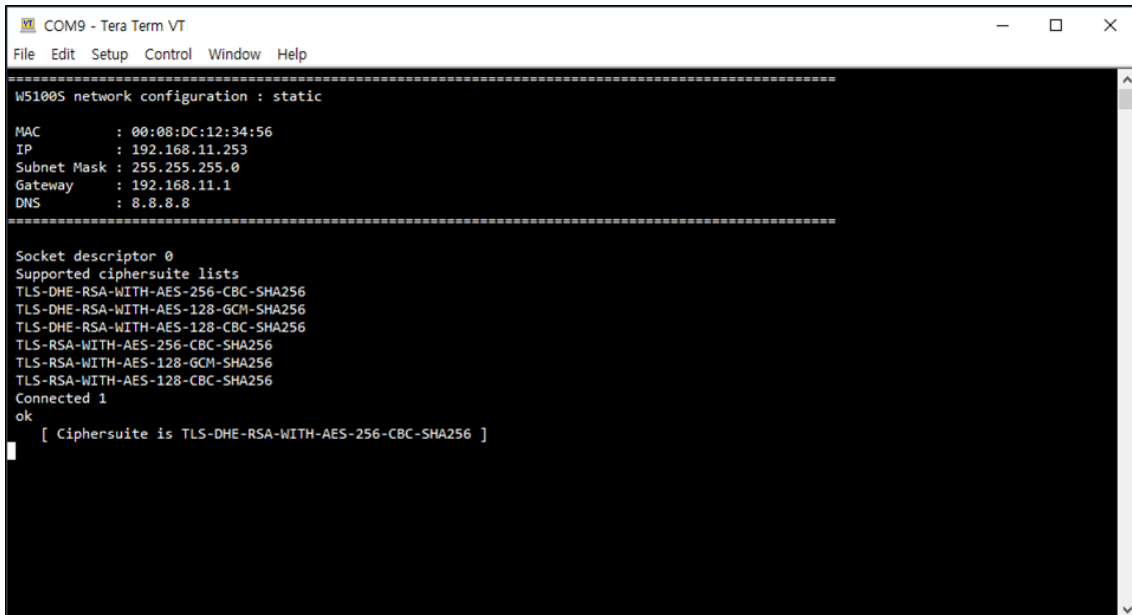
```
// e.g.
s_server -accept 443 -cert server.crt -key server.key
```



```
C:\Program Files\OpenSSL-Win64\bin\openssl.exe
OpenSSL> s_server -accept 443 -cert server.crt -key server.key
Enter pass phrase for server.key:
Using default temp DH parameters
ACCEPT
```

Figure 4. Create SSL server using OpenSSL

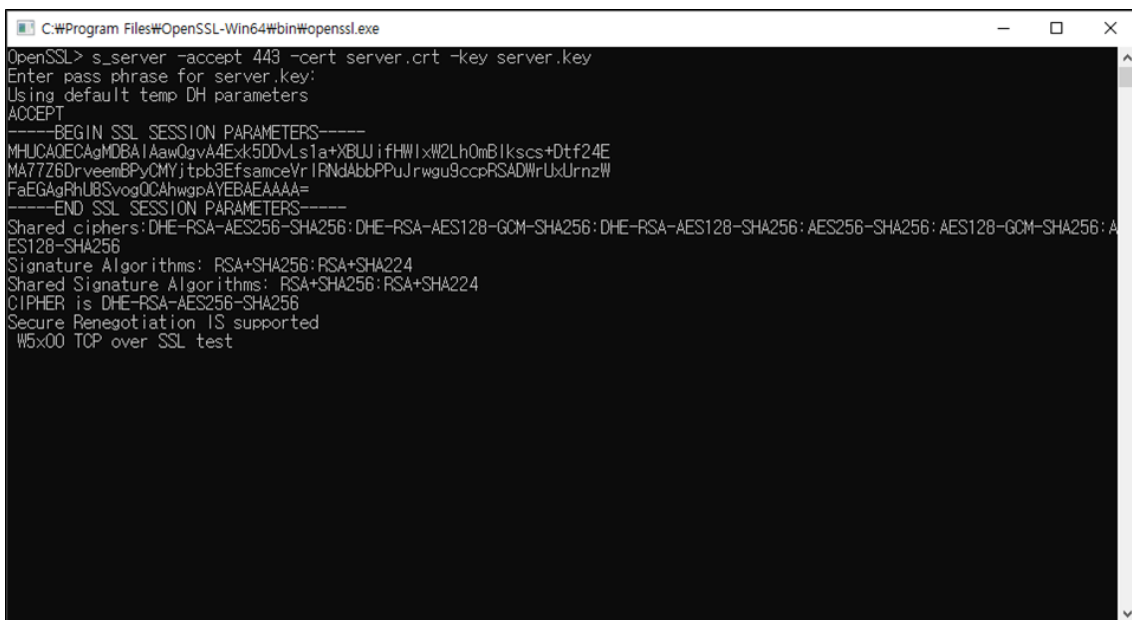
6. Reset your board.
7. If the TCP Client over SSL example works normally on Raspberry Pi Pico, W5100S-EVB-Pico, W5500-EVB-Pico, W55RP20-EVB-Pico, W5100S-EVB-Pico2 or W5500-EVB-Pico2, you can see the network information of Raspberry Pi Pico, W5100S-EVB-Pico, W5500-EVB-Pico, W55RP20-EVB-Pico, W5100S-EVB-Pico2 or W5500-EVB-Pico2, connecting to the SSL server and sending 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
=====
Socket descriptor 0
Supported ciphersuite lists
TLS-DHE-RSA-WITH-AES-256-CBC-SHA256
TLS-DHE-RSA-WITH-AES-128-GCM-SHA256
TLS-DHE-RSA-WITH-AES-128-CBC-SHA256
TLS-RSA-WITH-AES-256-CBC-SHA256
TLS-RSA-WITH-AES-128-GCM-SHA256
TLS-RSA-WITH-AES-128-CBC-SHA256
Connected 1
ok
[ Ciphersuite is TLS-DHE-RSA-WITH-AES-256-CBC-SHA256 ]
  
```

Figure 5. Connect to SSL server and sending message 1



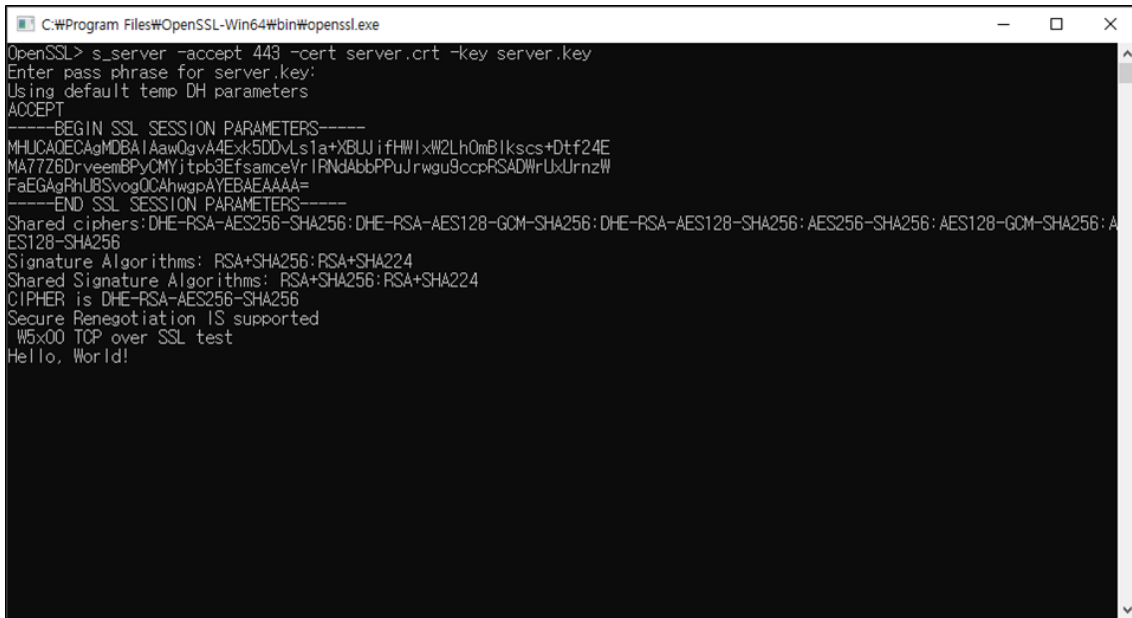
```

C:\Program Files\OpenSSL-Win64\bin\openssl.exe
OpenSSL> s_server -accept 443 -cert server.crt -key server.key
Enter pass phrase for server.key:
Using default temp DH parameters
ACCEPT
-----BEGIN SSL SESSION PARAMETERS-----
MHUCAQECAGMBAIAawQgvA4Exk5DDvLs1a+XBUJifHwIxW2Lh0mBIkscs+Dtf24E
MA77Z6DroveemBPvCMYjtpb3EfsanceYrIRNdAbbPPuJrwgu9ccpRSADWrlxUrnzW
FaEGAgRhUBSvogQCAhwgpAYEBAEAAAA=
-----END SSL SESSION PARAMETERS-----
Shared ciphers: DHE-RSA-AES256-SHA256: DHE-RSA-AES128-GCM-SHA256: DHE-RSA-AES128-SHA256: AES256-SHA256: AES128-GCM-SHA256: AES128-SHA256
Signature Algorithms: RSA+SHA256: RSA+SHA224
Shared Signature Algorithms: RSA+SHA256: RSA+SHA224
CIPHER is DHE-RSA-AES256-SHA256
Secure Renegotiation IS supported
W5x00 TCP over SSL test
  
```

Figure 6. Connect to SSL server and sending message 2

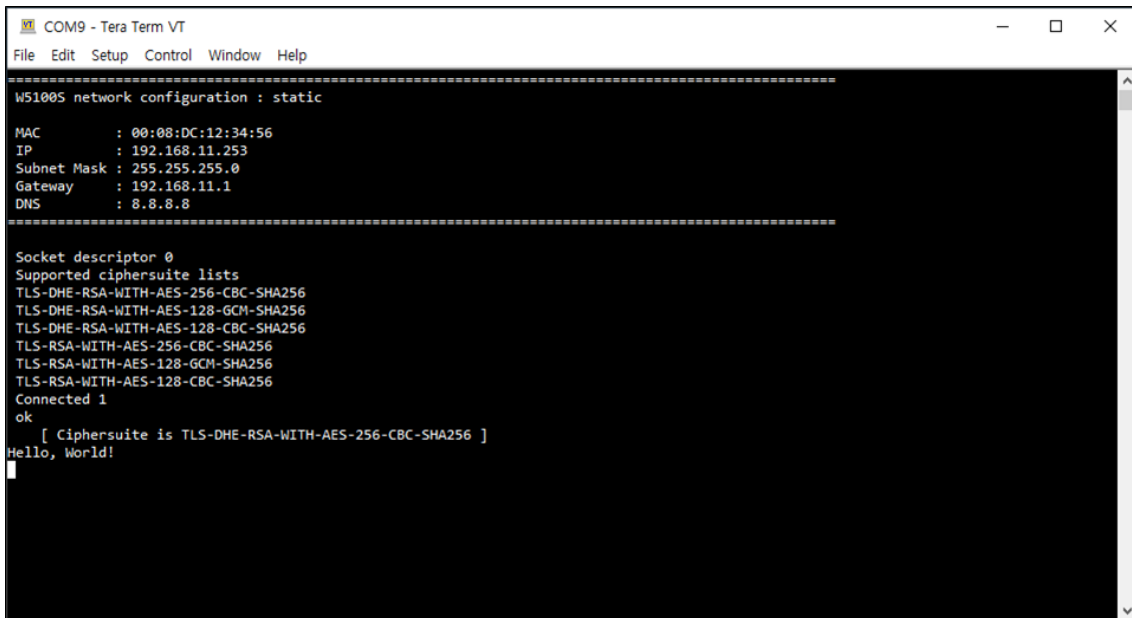


8. Once connected if you send data to the Raspberry Pi Pico, W5100S-EVB-Pico, W5500-EVB-Pico, W55RP20-EVB-Pico, W5100S-EVB-Pico2 or W5500-EVB-Pico2 from the SSL server, you should be able to receive the sent message on Raspberry Pi Pico, W5100S-EVB-Pico, W5500-EVB-Pico, W55RP20-EVB-Pico, W5100S-EVB-Pico2 or W5500-EVB-Pico2.



```
C:\Program Files\OpenSSL-Win64\bin>openssl.exe
OpenSSL> s_server -accept 443 -cert server.crt -key server.key
Enter pass phrase for server.key:
Using default temp DH parameters
ACCEPT
-----BEGIN SSL SESSION PARAMETERS-----
MHUCAQECAGMDBAIAaw0gvA4E:k5DDvLs1a+XBLUjifHwIxW2Lh0mB1kscs+Dtf24E
MA77Z6DrveemBPvCMYjtpb3EfsamceYrIRNdAbbPPuJrWgu9ccpRSADWrUxUrnzW
FaEGAgRHUBSvogQCAhwgpAYEBAEAAAA=
-----END SSL SESSION PARAMETERS-----
Shared ciphers: DHE-RSA-AES256-SHA256: DHE-RSA-AES128-GCM-SHA256: DHE-RSA-AES128-SHA256: AES256-SHA256: AES128-GCM-SHA256: AES128-SHA256
Signature Algorithms: RSA+SHA256: RSA+SHA224
Shared Signature Algorithms: RSA+SHA256: RSA+SHA224
CIPHER is DHE-RSA-AES256-SHA256
Secure Renegotiation IS supported
.W5x00 TCP over SSL test
Hello, World!
```

Figure 7. Receive sent message 1



```
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
=====

Socket descriptor 0
Supported ciphersuite lists
TLS-DHE-RSA-WITH-AES-256-CBC-SHA256
TLS-DHE-RSA-WITH-AES-128-GCM-SHA256
TLS-DHE-RSA-WITH-AES-128-CBC-SHA256
TLS-RSA-WITH-AES-256-CBC-SHA256
TLS-RSA-WITH-AES-128-GCM-SHA256
TLS-RSA-WITH-AES-128-CBC-SHA256
Connected 1
ok
 [ Ciphersuite is TLS-DHE-RSA-WITH-AES-256-CBC-SHA256 ]
Hello, World!
```

Figure 8. Receive sent message 2

## 4.6 Appendix

MBEDTLS library was ported to use SSL, please refer to following link to find version of ported mbed TLS.

- [mbed TLS](#)

## Revision history

Version	Date	Descriptions
Ver. 1.0.0	Nov, 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/>