HTTPS Web Server with Virtual File System Application

This application note demonstrates the HTTPS web server for Virtual File System.

Contents

1. Introduction **2**
2. Prerequisites for HTTPS Web Server Application **2**
2.1 Software Requirements 2
2.2 Hardware Requirements 2
2.3 Firmware Image 2
3. HTTPS Web Server Virtual File System Application **3**
3.1 View File on Linux Machine 3
3.2 View File on Windows Machine 6
4. Add New File **12**
5. Revision History **14**

Figures

Figure 1. Change SSID and password 3
Figure 2. Certificate file 3
Figure 3. Console logs 4
Figure 4. Static index.html page 5
Figure 5. 404.html Page 5
Figure 6. sics.gif 6
Figure 7. Renamed certificate file 6
Figure 8. Certificate information 7
Figure 9. Select Local Machine 8
Figure 10. Certificate Import Wizard 8
Figure 11. Select Certificate Store 9
Figure 12. Certificate Store 9
Figure 13. Certificate settings 10
Figure 14. Certificate import success 10
Figure 15. Installation success 11
Figure 16. Sample HTML page 12
Figure 17. Structure array 12
Figure 18. Structure format 13
Figure 19. Increment number of files 13
Figure 20. New added file 13

© 2026 Renesas Electronics

Terms and Definitions

HTML	HyperText Markup Language
HTTPS	HyperText Transfer Protocol Secure
IP	Internet Protocol
lwIP	Lightweight IP
RTOS	Real Time Operating System
SDK	Software Development Kit
TLS	Transport Layer Security
Wi-Fi®	Wireless Fidelity

1. Introduction

This application note demonstrates an HTTPS web server running on the Renesas RA6W1/RA6W2 platform over Wi-Fi, using the Renesas FSP middleware stack and FreeRTOS. The application connects to a Wi-Fi access point in Station mode, obtains an IPv4 address from the network, and serves a default web page over TLS.

2. Prerequisites for HTTPS Web Server Application

For running the HTTPS Web Server application, it is essential to set up the Renesas Flexible Software Package (FSP), required development tools, and supported hardware environment, as explained in the following sections.

2.1 Software Requirements

- A Linux/Windows machine is used, and a web browser for viewing the file must be present in the machine
- e^2 studio IDE: Version 2026-04.3 is required for building and flashing the firmware to the RA6W1/RA6W2. See *RA6W1*/RRQ61001 *Application Development Guide*.
- RAFW FSP Version: 2.2.0 and later
- GCC ARM Embedded Toolchain: Version 13.3.1.arm-13-24
- Serial Terminal Application: Tera Term or similar tools for logging and monitoring output.

2.2 Hardware Requirements

- An access point is required for Wi-Fi connectivity.

2.3 Firmware Image

In this project, either `wifi_https_server_ek_ra6w1` or `wifi_https_server_ek_ra6w2` can be used depending on the daughterboard being used.

3. HTTPS Web Server Virtual File System Application

Follow a step-by-step procedure for HTTPS web server virtual file system application. The procedures for viewing the file in both Linux and Windows machines are demonstrated in the following sections.

3.1 View File on Linux Machine

1. Import the project and update the access point's SSID and password in the `http_svr.h` file. Then, build the project using e^2 studio.

```
/***********************************************************************************************************
* File Name    : http_svr.h
* Description  : Contains macros, data structures and functions used common to the EP
***********************************************************************************************************/
/***********************************************************************************************************
* Copyright (c) 2020 - 2025 Renesas Electronics Corporation and/or its affiliates*
* SPDX-License-Identifier: BSD-3-Clause
***********************************************************************************************************/
#ifndef HTTP_SVR_H_
#define HTTP_SVR_H_

#include "common_utils.h"

#define SSID "ASUS_KEYSGT_24G"
#define PASSPHRASE "          "
```

Figure 1. Change SSID and password

2. After successfully building the project, copy `ca_cert.pem` under the `certificate` folder into `/usr/local/share/ca-certificates` in the Linux machine, as shown in Figure 2. Update certificates on the Linux machine using the command:
   `sudo update-ca-certificates`

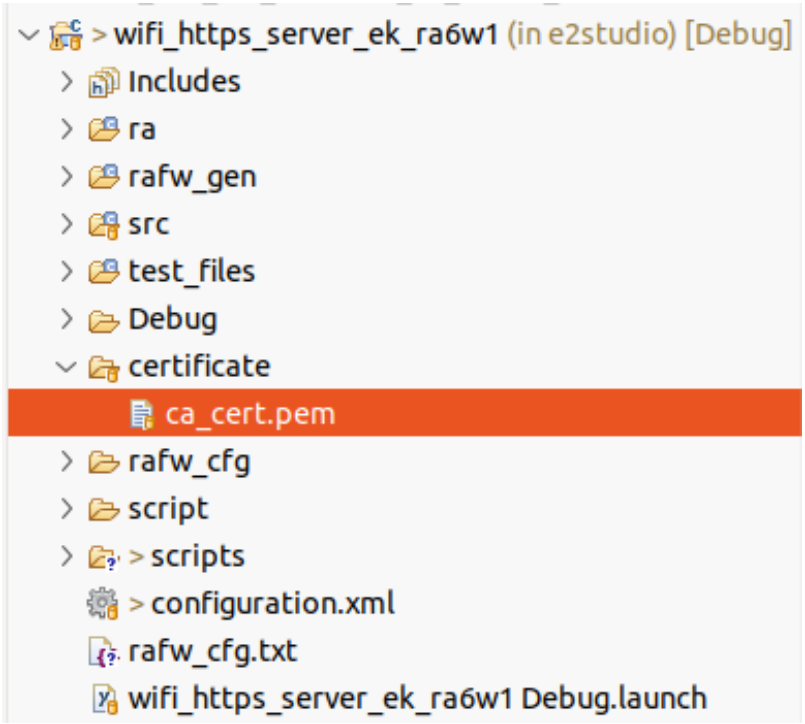


Figure 2. Certificate file

3. After successful programming of the firmware, press the **Reset** button on the motherboard and connect the RA6W1/RA6W2 to an access point, which has Wi-Fi connectivity. Here, RA6W1/RA6W2 acts as an HTTPS server. Note the assigned IP address after connection from the console logs, as shown in Figure 3.

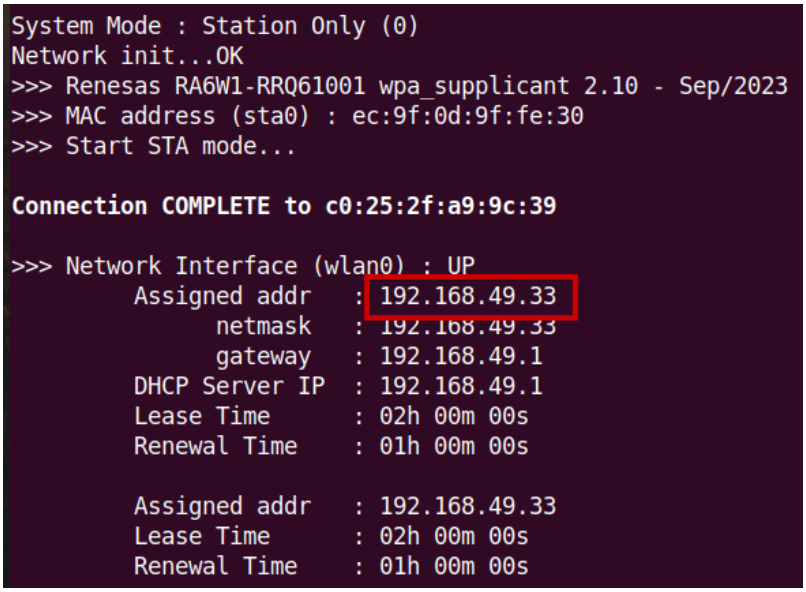


Figure 3. Console logs

4. Add a line `<ip address> mydevice.local` to `/etc/hosts` in the Linux machine. For example, `192.168.49.33 mydevice.local`.
5. Connect the Linux machine to the same Wi-Fi network as the RA6W1/RA6W2.

6. Open any web browser and browse for https://mydevice.local.
   A static `index.html` page is loaded with the information, as shown in Figure 4. The default page contains a brief description of lwIP.

NOTE
A security warning appears. Select **Proceed** to view the page.

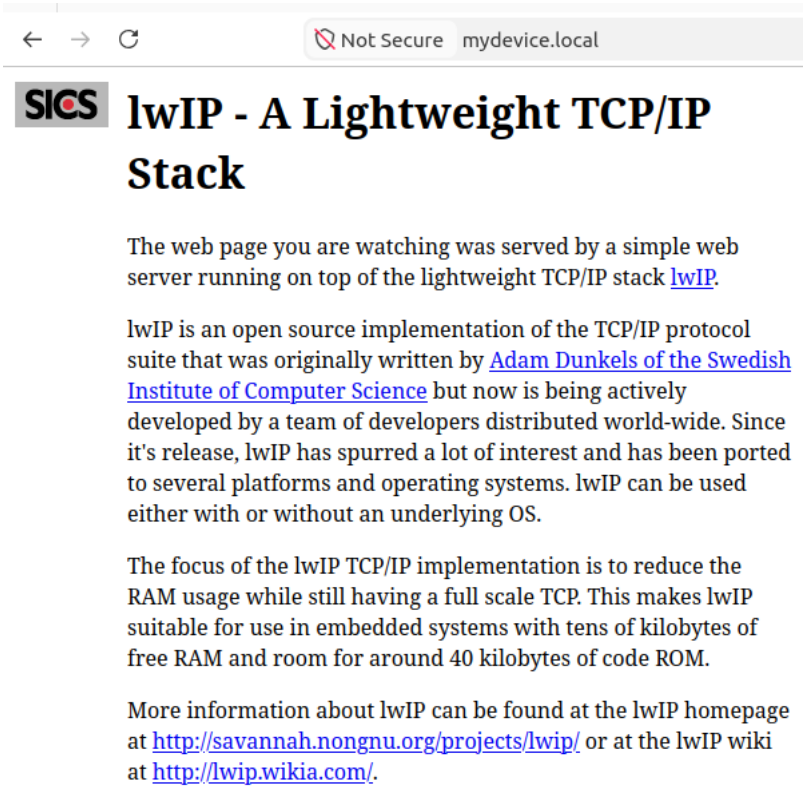


Figure 4. Static index.html page

7. Apart from `index.html`, two more pages, `404.html` and `sics.gif`, can be viewed which are part of the SDK. To view those pages, insert the following links to the browser, for example, https://mydevice.local/404.html and https://mydevice.local/img/sics.gif. See Figure 5 and Figure 6.



Figure 5. 404.html Page

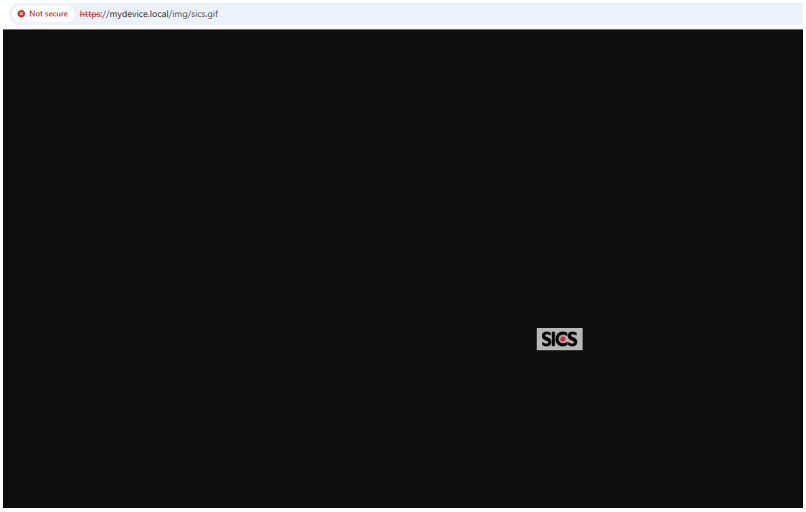


Figure 6. sics.gif

3.2 View File on Windows Machine

1. Import the project and update the access point's SSID and password in the `http_svr.h` file, as shown in Figure 1. Then, build the project using e^2 studio.
2. Rename the `ca_cert.pem` generated in the `certificate` folder (see Figure 2) to `ca_cert.crt`, as shown in Figure 7.



Figure 7. Renamed certificate file

3. Double-click the `ca_cert.crt` file. Windows opens a built-in Certificate dialog box, as shown in Figure 2. It displays an error message that CA Root certificate is not trusted. Click **Install Certificate**, and then click **OK**.

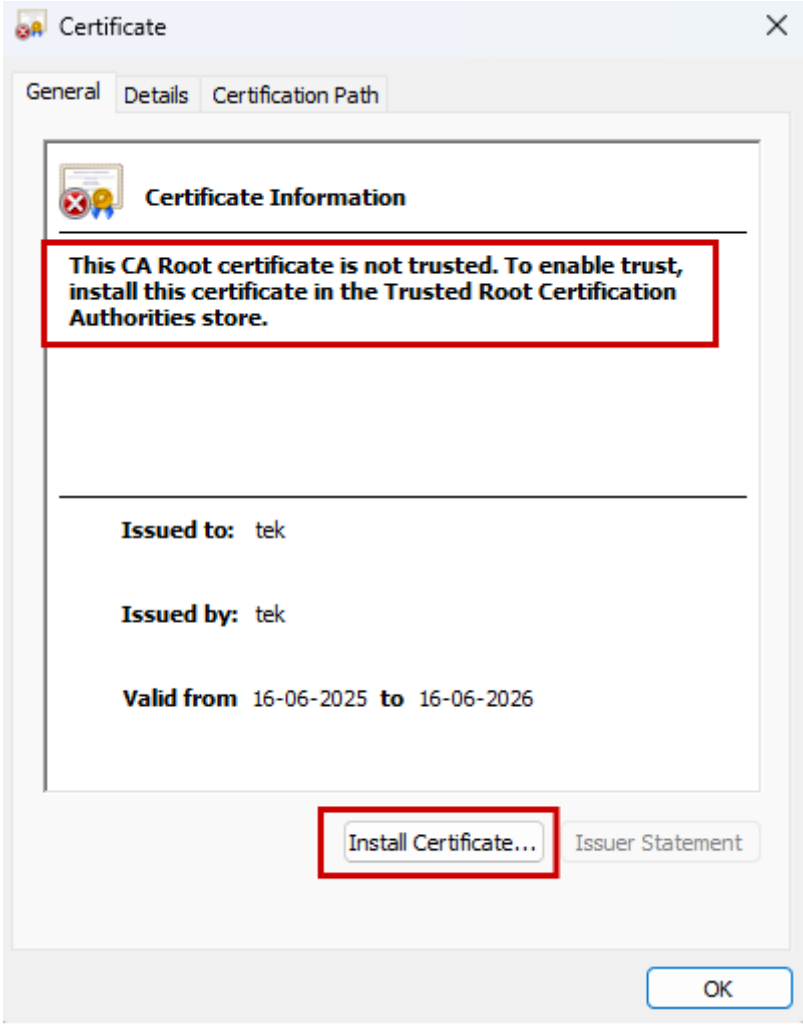


Figure 8. Certificate information

4. Select **Local Machine**, and then click **Next**.

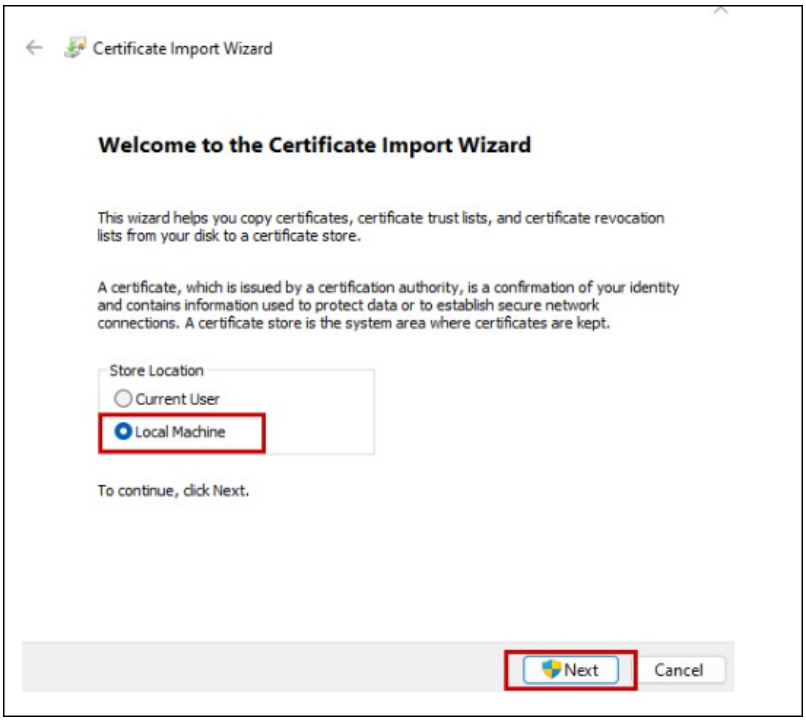


Figure 9. Select Local Machine

5. In the Certificate Import Wizard, select **Place all certificates in the following store**, and then click **Browse**.

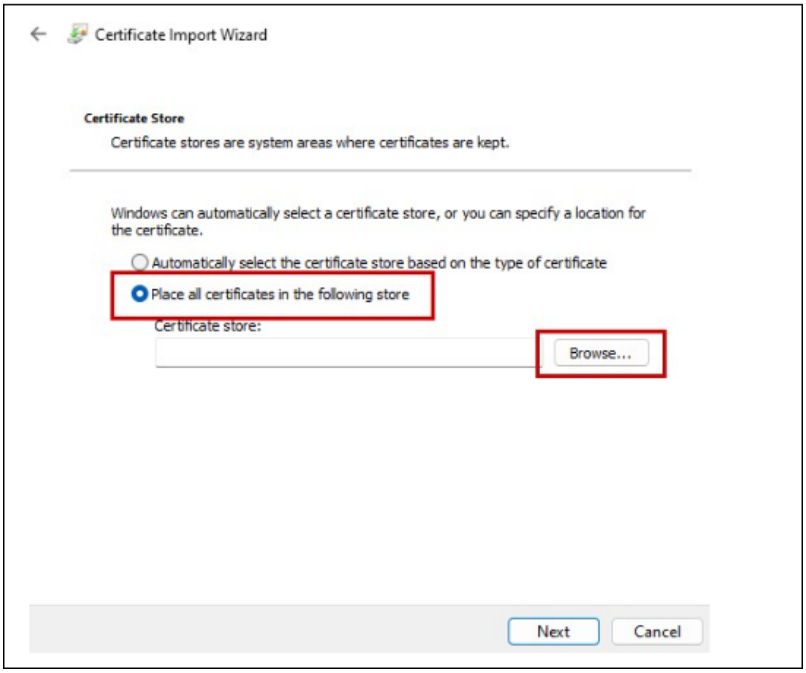


Figure 10. Certificate Import Wizard

6. Select **Trusted Root Certification Authorities** in the certificate store, and then click **OK**.

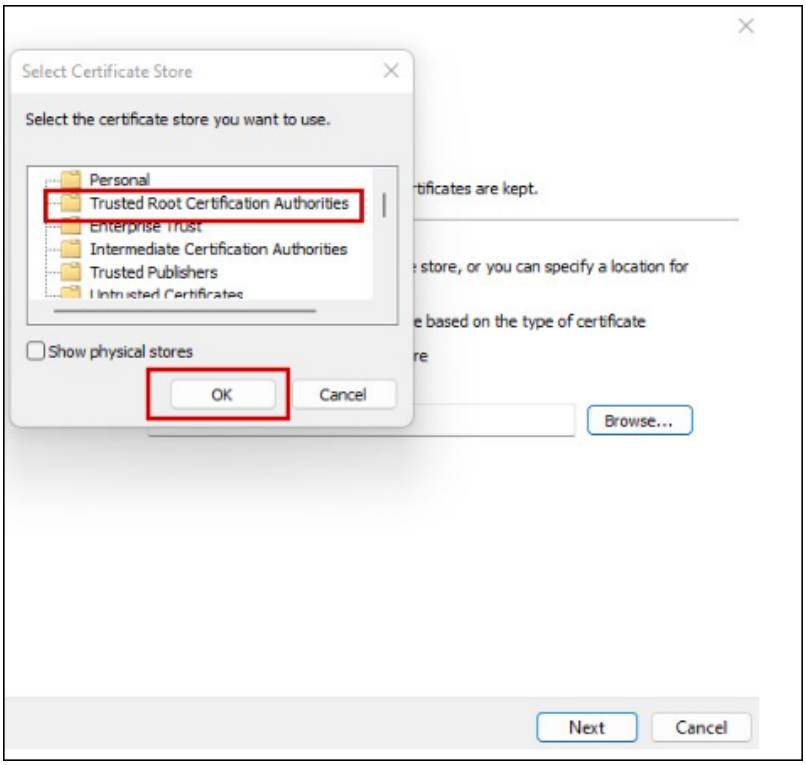


Figure 11. Select Certificate Store

7. Ensure that **Trusted Root Certification Authorities** is selected under **Certificate Store**, as shown in Figure 12, and then click **Next**.

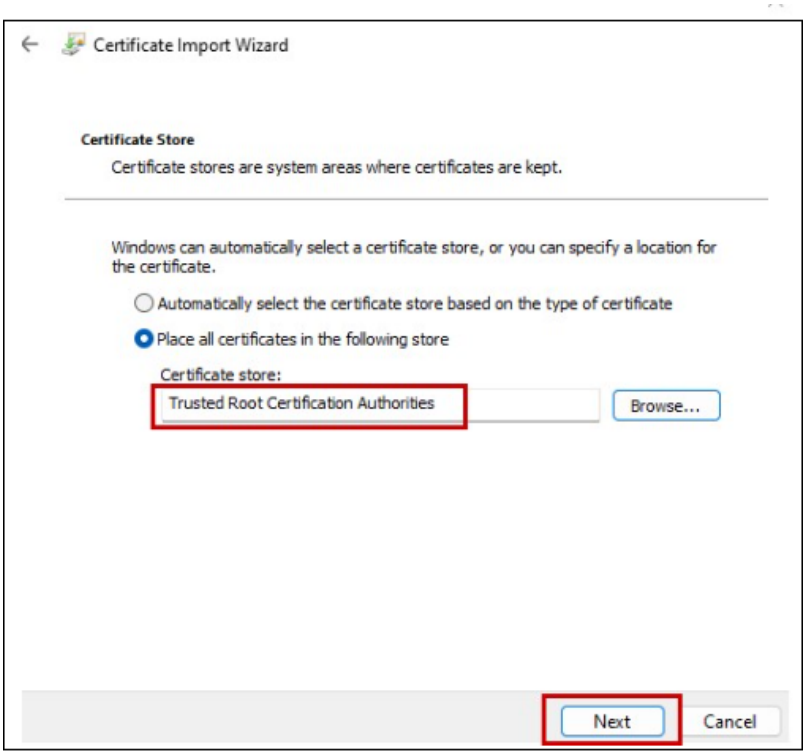


Figure 12. Certificate Store

8. Verify that the selected certificate settings are correct, as shown in Figure 13, and then click **Finish**.

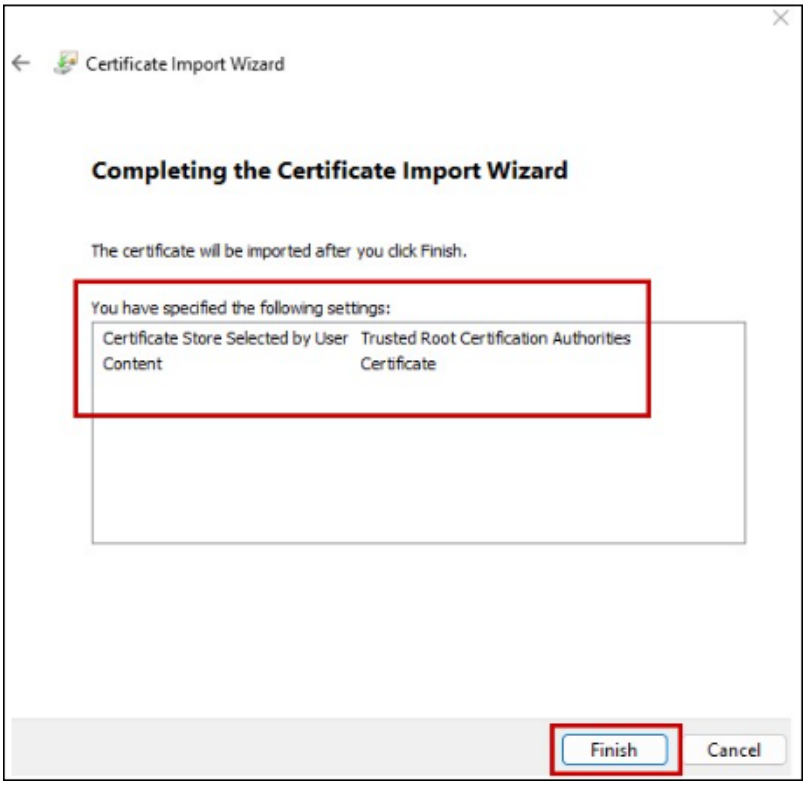


Figure 13. Certificate settings

9. After the certificate is imported successfully, click **OK** to continue.

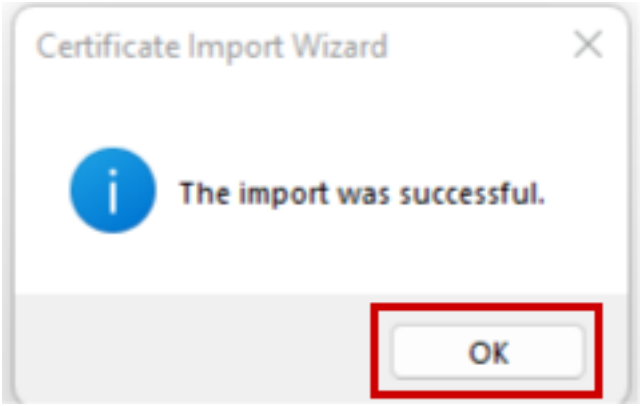


Figure 14. Certificate import success

10. After installation, Windows automatically uses the installed certificate, as shown in Figure 15.

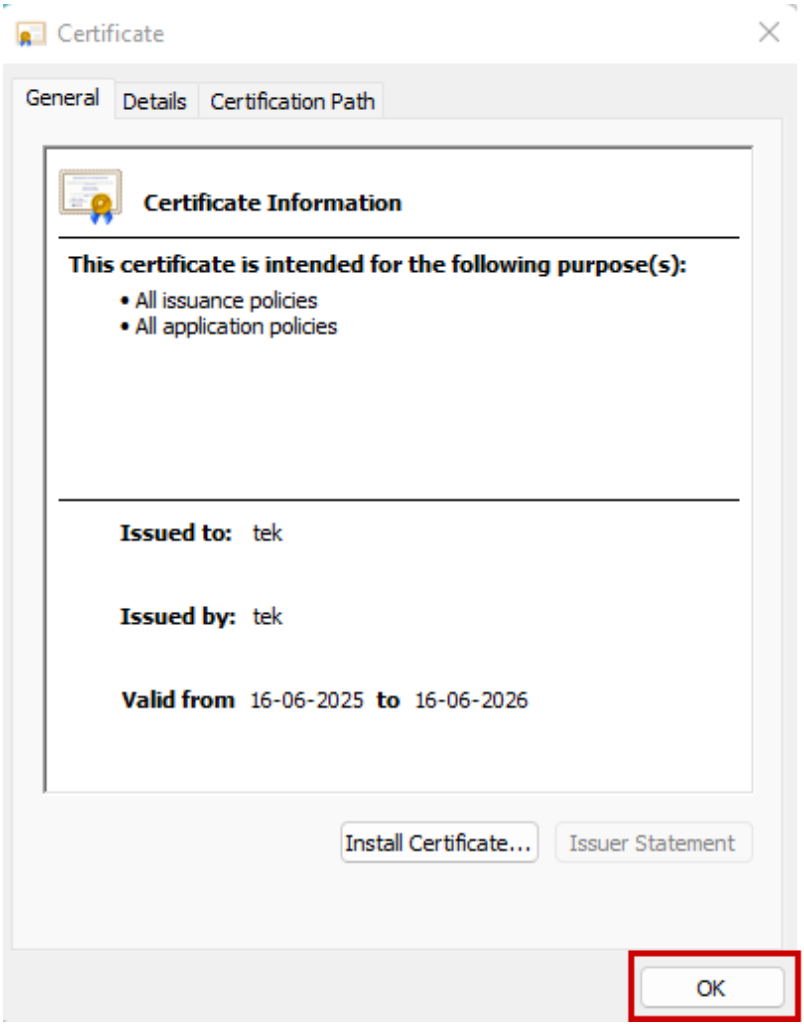


Figure 15. Installation success

11. Run Notepad as administrator and open the following file: `C:\Windows\System32\drivers\etc\hosts`. At the end of the file, add a line `<ip_address> mydevice local` , for example, `192.168.49.33 mydevice.local`
12. Connect the Windows machine to the same network as RA6W1/RA6W2.
13. Open https://mydevice.local in a web browser. A static HTML webpage is loaded, which contains a brief explanation on lwIP, as shown in Figure 3.

NOTE
A security warning will appear. Select **Proceed** to view the page.

14. Other pages that are part of the SDK and can be viewed are `404.html` and `img/sics.gif`, as shown in Figure 4 and Figure 5.

4. Add New File

Modifying a file or adding a new static file needs changes in the SDK code.
To add a new file:

1. Place the new file inside the path given below. The default pages such as `index.html`, `404.html`, and `img/sics.gif` are found in the `fs` folder.
   `<root_dir>ra/fsp/src/rm_lwip_w/apps/http/fs/`
   A sample HTML page, `new.html` is added for reference, as shown in Figure 16.

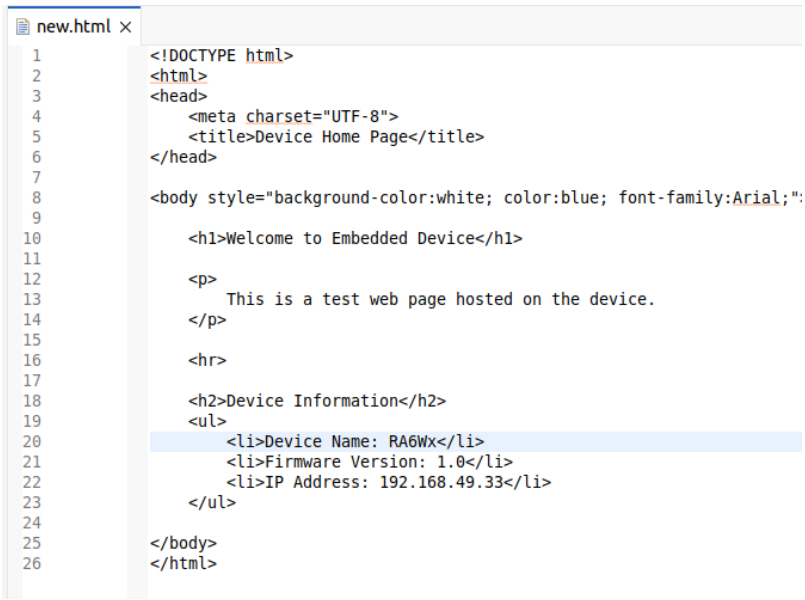


Figure 16. Sample HTML page

2. Run the `makefsdata` Perl script to generate the `fsdata.c` file using the following commands. This script converts the file content to its corresponding hex byte array.
   - cd `ra/fsp/src/rm_lwip_w/apps/http`
   - perl `<root_dir>ra/fsp-rrq-3rd-party/lwip/src/apps/http/makefsdata/makefsdata`
3. Copy the hex byte array of the new file to the `fsdata.c.h` in the correct format followed in the `fsdata.c.h` file. For example, if the new file added is `new.html`, change the structure name to `data__new_html` and add `dummy_align__new_html = 3,` as shown in Figure 17.

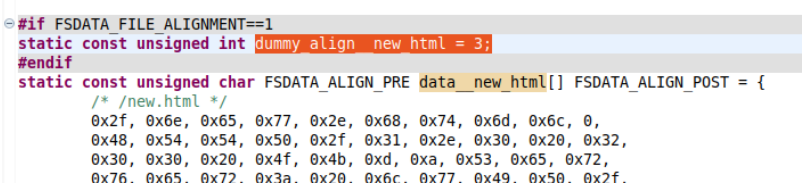


Figure 17. Structure array

4. Inside `file__img_sics.gif[]`, instead of `file_NULL`, give `file_new_html`. Copy the `file_new_html` from `fsdata.c` to `fsdata.c.h`..
   Then, add `FS_FILE_FLAGS_HEADER_INCLUDED | FS_FILE_FLAGS_HEADER_PERSISTENT` to the structure, as shown in Figure 18.

```
const struct fsdata_file file_new_html[] = {{
file_NULL,
data__new_html,
data__new_html + 10,
sizeof(data__new_html) - 10,
FS_FILE_FLAGS_HEADER_INCLUDED | FS_FILE_FLAGS_HEADER_PERSISTENT,
}};

const struct fsdata_file file__img_sics_gif[] = { {
file_new_html,
data__img_sics_gif,
data__img_sics_gif + 16,
sizeof(data__img_sics_gif) - 16,
FS_FILE_FLAGS_HEADER_INCLUDED | FS_FILE_FLAGS_HEADER_PERSISTENT,
}};
```

Figure 18. Structure format

5. Increment `FS_NUMFILES` to 4 in `fsdata.c.h`

```
#define FS_ROOT file__index_html
#define FS_NUMFILES 4
```

Figure 19. Increment number of files

6. After making all the changes listed above, delete the generated `fsdata.c` file. Then, build the project using e^2 studio. View the new file in the browser (see Figure 20) using this link: https://mydevice.local/new.html.

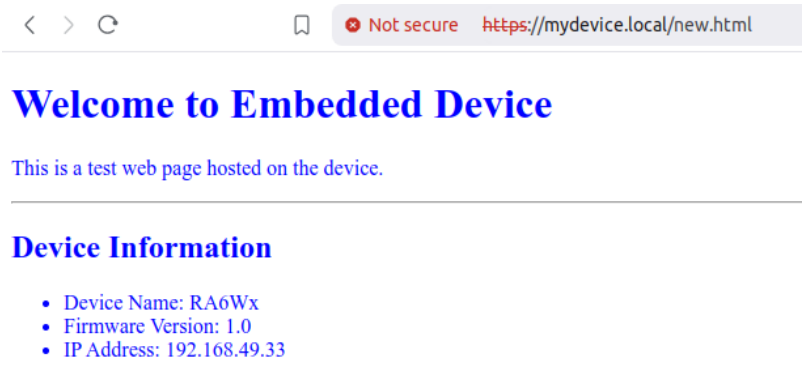


Figure 20. New added file

5. Revision History

Revision	Date	Description
1.02	Sep 21, 2026	Modified the e^2 studio version. Added FSP pack version.
1.01	July 21, 2026	Modified the e^2 studio version. Added FSP pack version. Added description for prerequisites.
1.00	May 19, 2026	Initial release.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01)

CorporateHeadquarters

TOYOSU FORESIA, 3-2-24 Toyosu
Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/

© 2026 Renesas Electronics Corporation. All rights reserved.