

Renesas Synergy™ Platform

Wi-Fi Framework on PE-HMI1

Introduction

This document enables you to effectively use the Wi-Fi Framework module in your own design. On completion of this guide, you will be able to add the Wi-Fi Framework module to your own design, configure it correctly for the target application, and write code using the included application example code as a reference and an efficient starting point. For advanced API descriptions and more complex application projects, see the Renesas Synergy™ Knowledge Base in the References section of this document. These valuable resources demonstrate how to create more complex designs.

The Wi-Fi Framework provides a generic application interface for Wi-Fi modules with or without on-chip networking capability. Currently, only Qualcomm GT202 is supported. The Wi-Fi Framework communicates through the SPI with the underlying GT202. This application note describes the features of the Application Project, and its implementation details on the PE-HMI1 Kit, when using the Synergy™ Software Package (SSP) and Synergy™ Wi-Fi Framework.

The application project included with this application note is the Synergy Thermostat Application. It is integrated with the onboard GT-202 module present on the PE-HMI1 to demonstrate it as a Wi-Fi-based Thermostat Application.

Required Resources

This application example targets the Renesas Synergy™ S7G2 MCU board. To build and run the project, you need:

- Renesas Synergy™ starter kit PE-HMI1 v2.0 and later
- e² studio ISDE v7.5.1 or later
- Synergy Software Package (SSP) v1.7.0. or later
- IAR Embedded Workbench® for Renesas Synergy™ (IAR EW for Synergy) v8.23.3 or later
- Synergy Standalone Configurator (SSC) v7.5.1 or later
- iOS or Android-based Smart Phone or Tablet (optional)
- Mini USB cable with J-Link® Lite
- A Wi-Fi router with 2.4 GHz bandwidth, with DHCP server capability to be used as an access point
- Download all the required Renesas Synergy Software from the Renesas Synergy Software website at www.renesas.com/synergy/software.

Prerequisites and Intended Audience

This application project assumes you have some experience with the Synergy e² studio ISDE and the Synergy Software Package (SSP). Before you perform the procedure in this application note, follow the procedure in the *SSP User's Manual* to build and run the Blinky project. Doing so enables you to become familiar with the e² studio and the SSP, and ensures that the debug connection to your board functions properly. In addition, this application project assumes you have some knowledge on Wi-Fi and its communication protocols.

The intended audience are users who want to develop applications with Wi-Fi interface using the Synergy S3/S5/S7 MCU Series.

Contents

1. Application Overview	2
2. An Application Using the Synergy Wi-Fi Framework	3
2.1 Features	3
2.2 Wi-Fi Framework Architecture Overview	3
2.3 Application Architecture	4
2.4 Hardware Requirements	5
3. Creating Wi-Fi Application with Existing Thermostat Application Project	5
3.1 Adding NetX DHCP Client to the Thread from the Configurator	6
3.2 NetX HTTP Server	10
3.3 LED Control	11
4. Wi-Fi Application Implementation Details	11
4.1 Wi-Fi Application Thread	12
4.2 HTTP Server Thread	12
4.3 LED Control Thread	13
4.4 Code Organization	13
5. Application Setup	14
5.1 Importing the Existing Project	14
5.2 Modifying the User Credentials in the Existing Project	14
6. Next Steps	17
7. References	17
Revision History	19

1. Application Overview

This application project shows Wi-Fi connectivity using the onboard GT-202 module on the PE-HMI1 Synergy MCU Kit. It demonstrates the Thermostat Application, along with the NetX Webserver application running on the PE-HMI1 kit. The user interface for controlling the Thermostat uses the GUI with GLCD and a touch screen interface. The Webserver application can also be used to control and get the status of the kit. By connecting to the IP address of the board where the Webserver is running, the status information can be viewed on a PC web browser or smart devices.

This application is developed by using the Synergy Wi-Fi Framework to provide you with easy provisioning, and configuration of the Wi-Fi Module. The framework is also integrated with the NetX Networking Stack (TCP/IP) using the NSAL layer.

2. An Application Using the Synergy Wi-Fi Framework

2.1 Features

This Wi-Fi application project consists of multiple SSP modules along with the Wi-Fi Framework used to demonstrate the Wi-Fi Connectivity features of the PE-HMI1 board.

The project currently provides the following:

- Supports client mode, connecting to an Access point/Router with user-configured credentials
- Supports NetX™ DHCP client to get IP address from the Server
- IP address leased from the DHCP server is displayed on the LCD screen
- Supports NetX™ HTTP server that runs on the PE-HMI1 board
- Allows you to set the LED and get the board status on the web page using the web interface

The application is fully integrated with Express Logic ThreadX® RTOS, NetX™, GUIX™, and SSP framework modules such as Wi-Fi Framework and Touch Framework.

2.2 Wi-Fi Framework Architecture Overview

The Wi-Fi implementation on Synergy is done with the framework module; it is designed to provide a vendor-independent, uniform set of APIs to users, while addressing the common functions required to initialize, configure, provision, and integrate the protocol stack in the Wi-Fi applications. The framework has generic APIs to configure and control the Wi-Fi module. The networking stack is integrated with the NetX through the Network Stack Abstraction Layer (NSAL). At the driver level, it interfaces to the Wi-Fi driver using a uniform Wi-Fi driver interface.

Figure 1 shows a layered architectural overview of the Synergy Wi-Fi Framework and how it fits with the SSP. Other details on the Wi-Fi Framework can be found in section 4.1.29 of the *SSP User's Manual*.

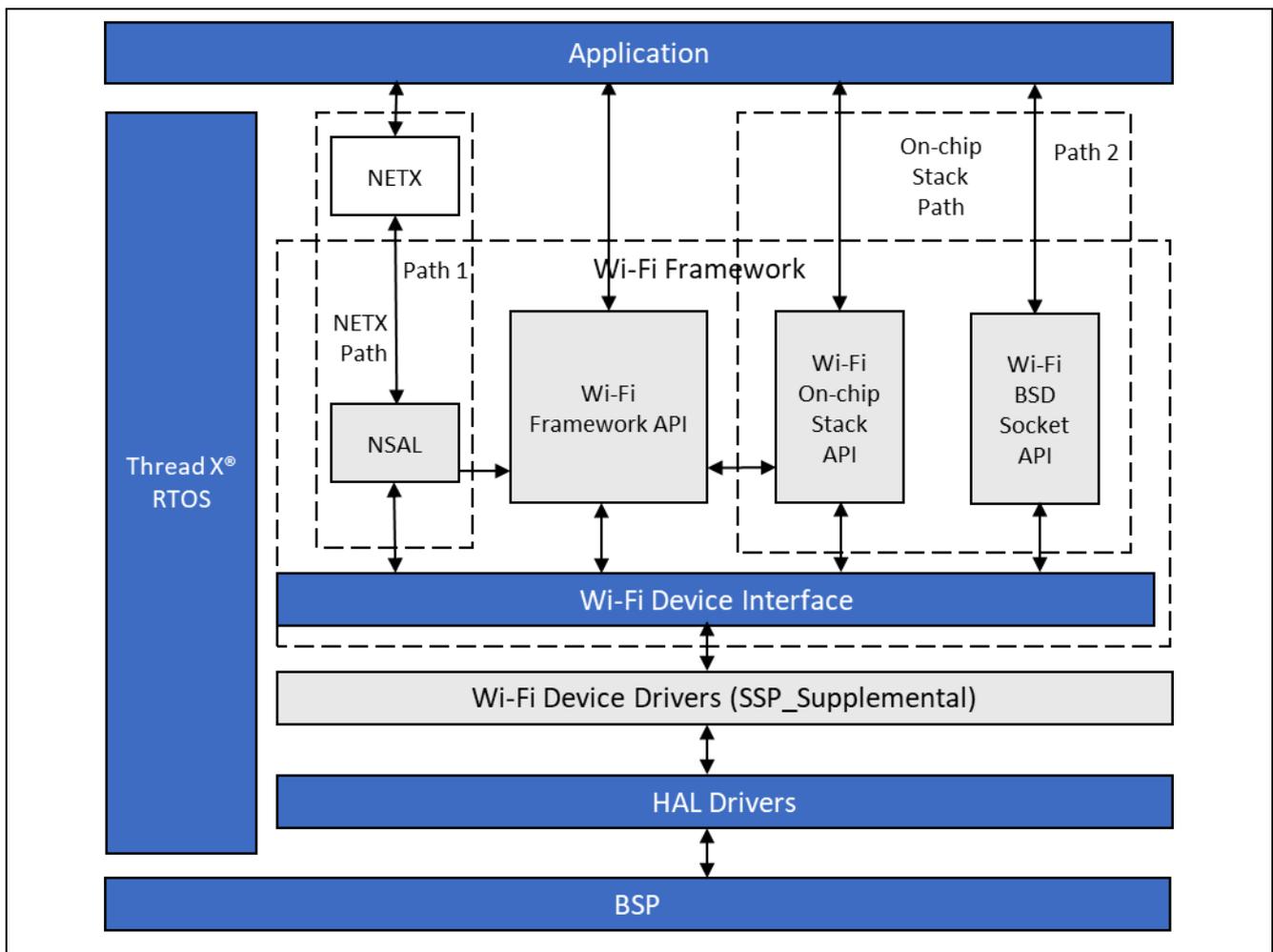


Figure 1. Synergy Wi-Fi Framework Architecture Overview

2.3 Application Architecture

The high-level architecture of the Wi-Fi application is shown in Figure 2. The Wi-Fi application runs in client mode. In this case, it connects to the Access Point (AP) or Router, gets the leased IP address from the AP/Router, and then becomes available for any IP application that uses TCP or UDP.

The application uses the NetX application protocols, DHCP client and HTTP server, that sit on top of the Wi-Fi Framework. The Wi-Fi Framework exposes the generic APIs to provisioning and connects to the GT202 Wi-Fi module. The Wi-Fi device driver communicates with the module using the SSP SPI drivers.

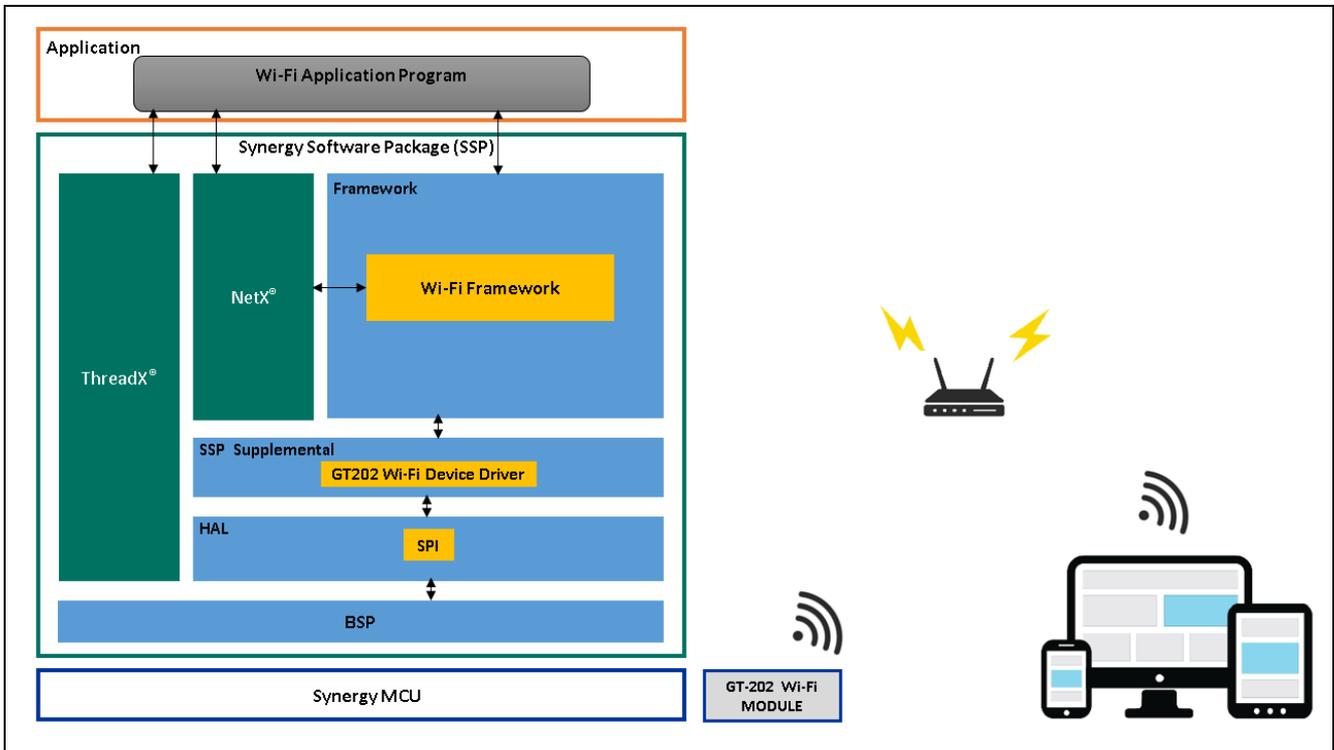


Figure 2. Synergy Wi-Fi Application Project Components

2.4 Hardware Requirements

This application requires the PE-HMI board. Figure 3 shows the PE-HMI1 Synergy MCU board and the onboard Wi-Fi module. Required along with the board is the Mini USB Cable, J-Link® Lite, and the power supply that are part of the kit.

If you are new to the PE-HMI1 Synergy MCU Kit, see the *PE-HMI1 Quick Start Guide* and *PE-HMI1 User's Manual* listed in the Reference section for more details.

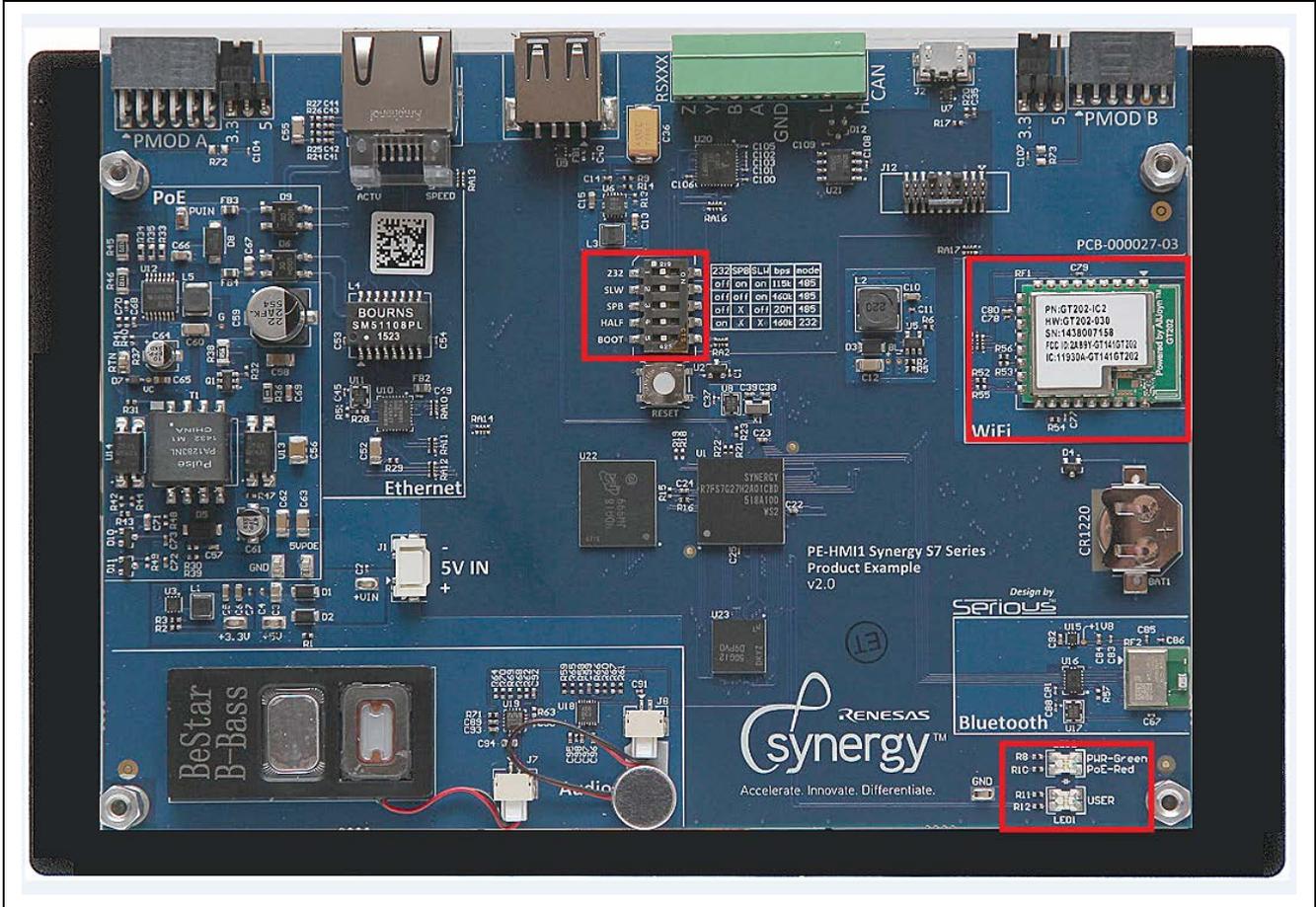


Figure 3. PE-HMI1 Synergy MCU

3. Creating Wi-Fi Application with Existing Thermostat Application Project

Use the existing Synergy Thermostat application as a base to create the Wi-Fi Application. Import the Synergy Thermostat Application for PE-HMI1 (see the References section), then make changes to the thermostat.gxp file (in Project\guix_studio\GNU\) to add the Wi-Fi icon and IP address prompt using GUIX™ Studio.

Open the configuration.xml file using the ISDE configurator. Start adding new threads and stacks.

Note: Adding GUIX widgets to the project is not covered. You can refer to the attached project and open the thermostat.gxp file using GUIX Studio to view details.

Adding the HTTP thread and HTTP server stack is not covered in detail. For information, you can look into the Thread Stack properties for the HTTP Server using the configurator for the attached project.

3.1 Adding NetX DHCP Client to the Thread from the Configurator

Create a thread and name it **wifi_app_thread**. Once the Wi-Fi application thread has been created, add DHCP Client to the newly created thread using the **New Stack > X-Ware > NetX > NetX DHCP Client** as shown in the following figure.

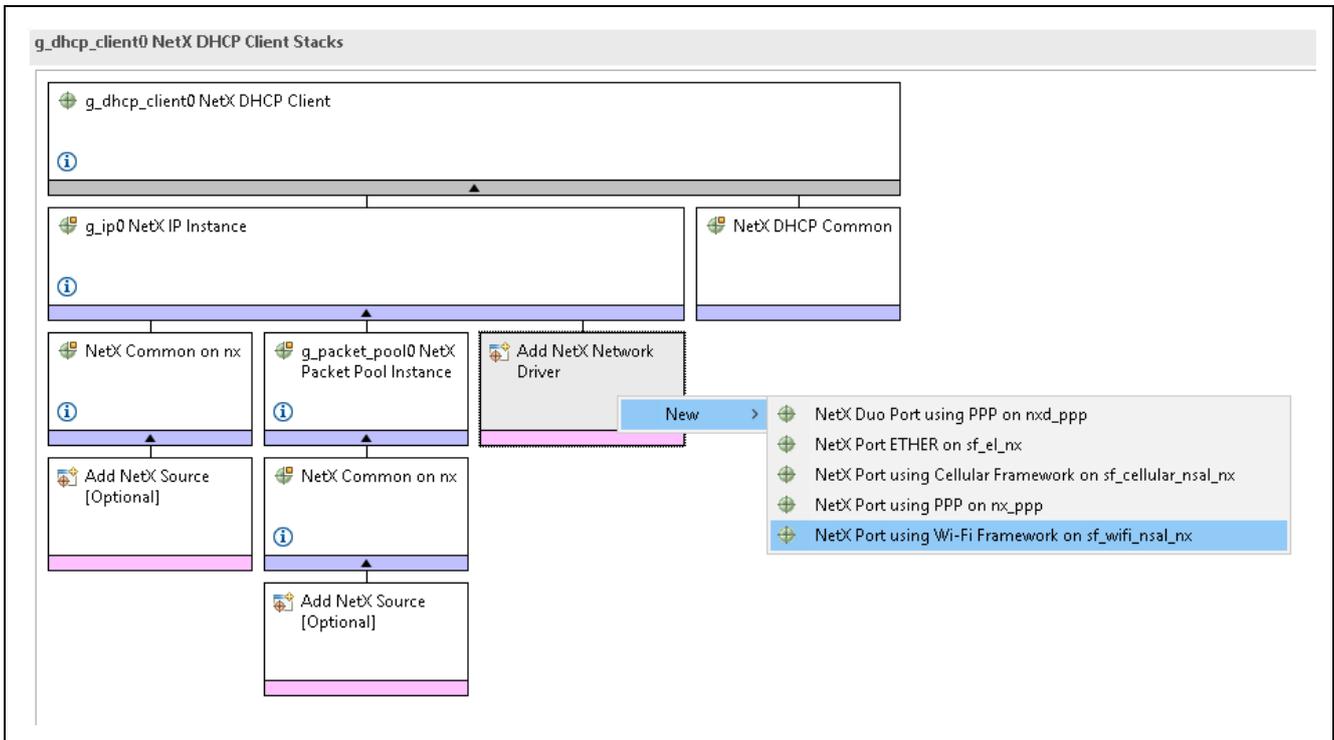


Figure 4. Adding Wi-Fi Framework with Application Protocols

Figure 4 shows an option to add the NetX network driver. Add **NetX Port using Wi-Fi Framework on sf_wifi_nsal_nx**. Adding this option brings in the Wi-Fi Framework.

Figure 5 shows the Wi-Fi Framework and the selection to add the Wi-Fi driver for the modules.

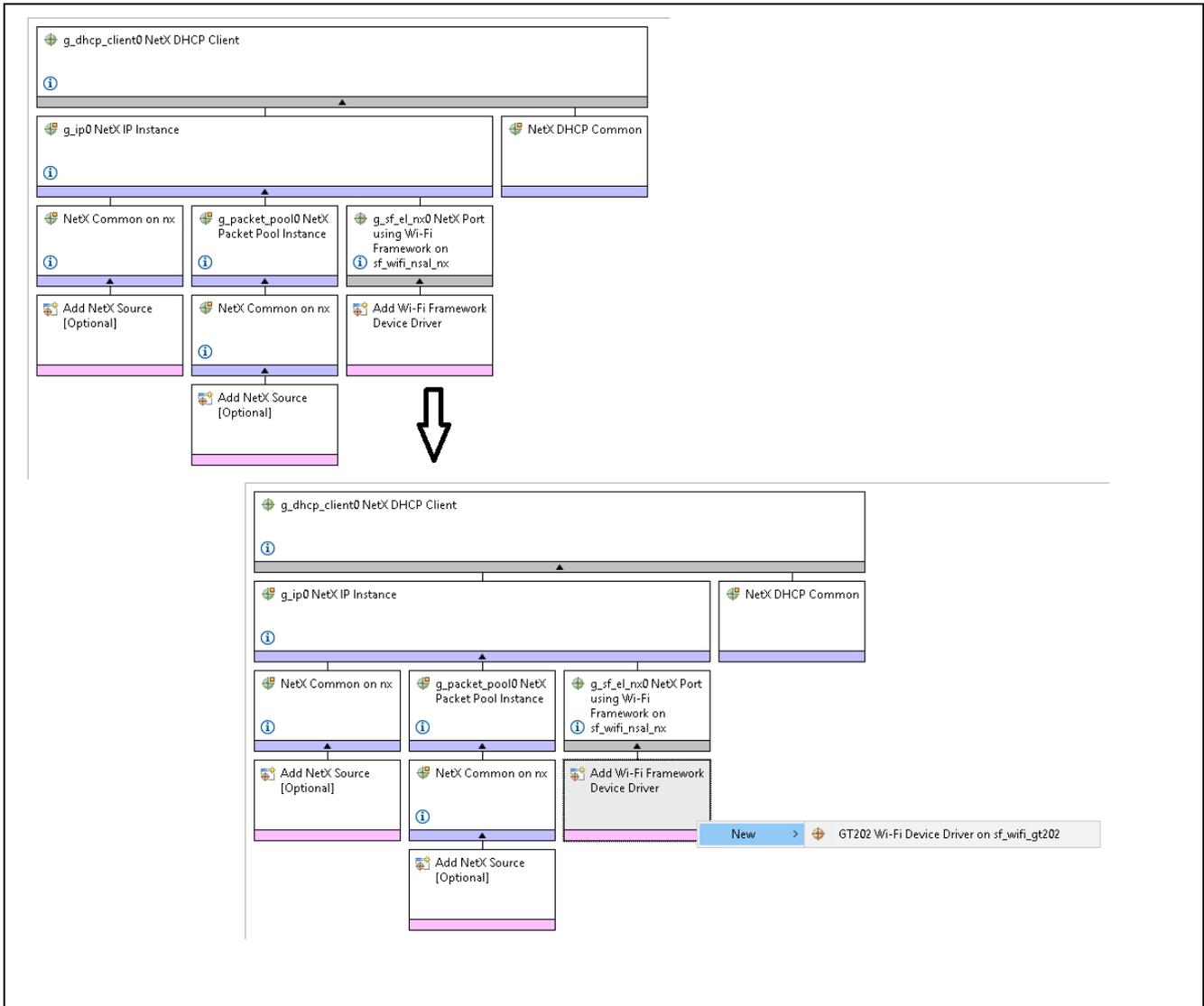


Figure 5. Synergy Wi-Fi Driver Module Add-on

You can select the driver for the intended Wi-Fi module. Figure 6 shows the added **GT202 Wi-Fi Device Driver**. Since the GT202 module uses the SPI interface, it gives the option to select the SPI driver available with the SSP. You need to select from one of two options for SPI, either SCI SPI or RSPI, depending on the SPI interface available on the kit.

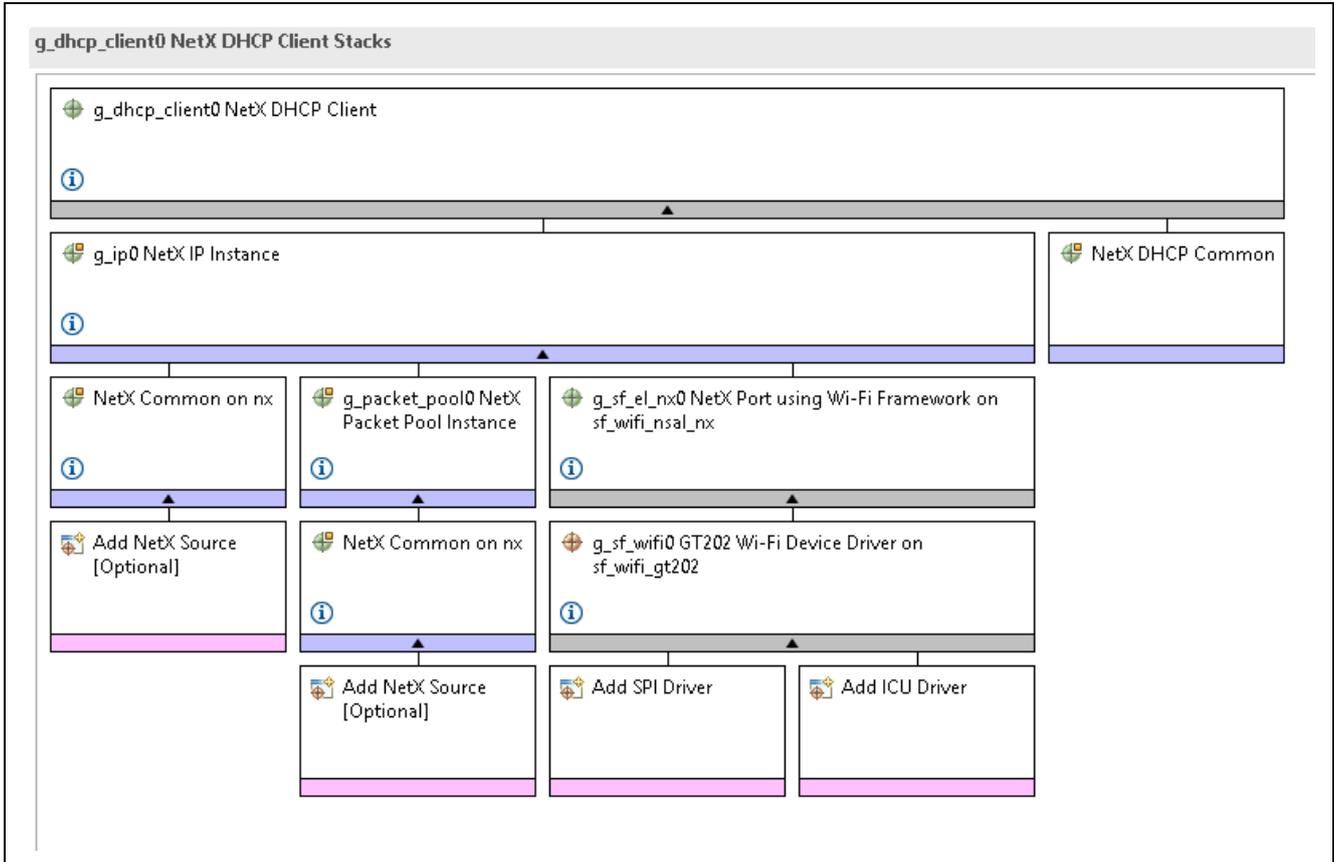


Figure 6. Adding SPI Driver and IRQ Driver

Once the RSPI is selected as the SPI interface for the GT202, the interrupt pin for the RSPI is configured, and Figure 7 shows the final DHCP client configuration.

Important: The DTC driver is not configured and the configuration is kept optional. The DTC driver is not configured because the DTC uses word 4-bit transfer and the Wi-Fi driver supports 8-bit transfer mode.

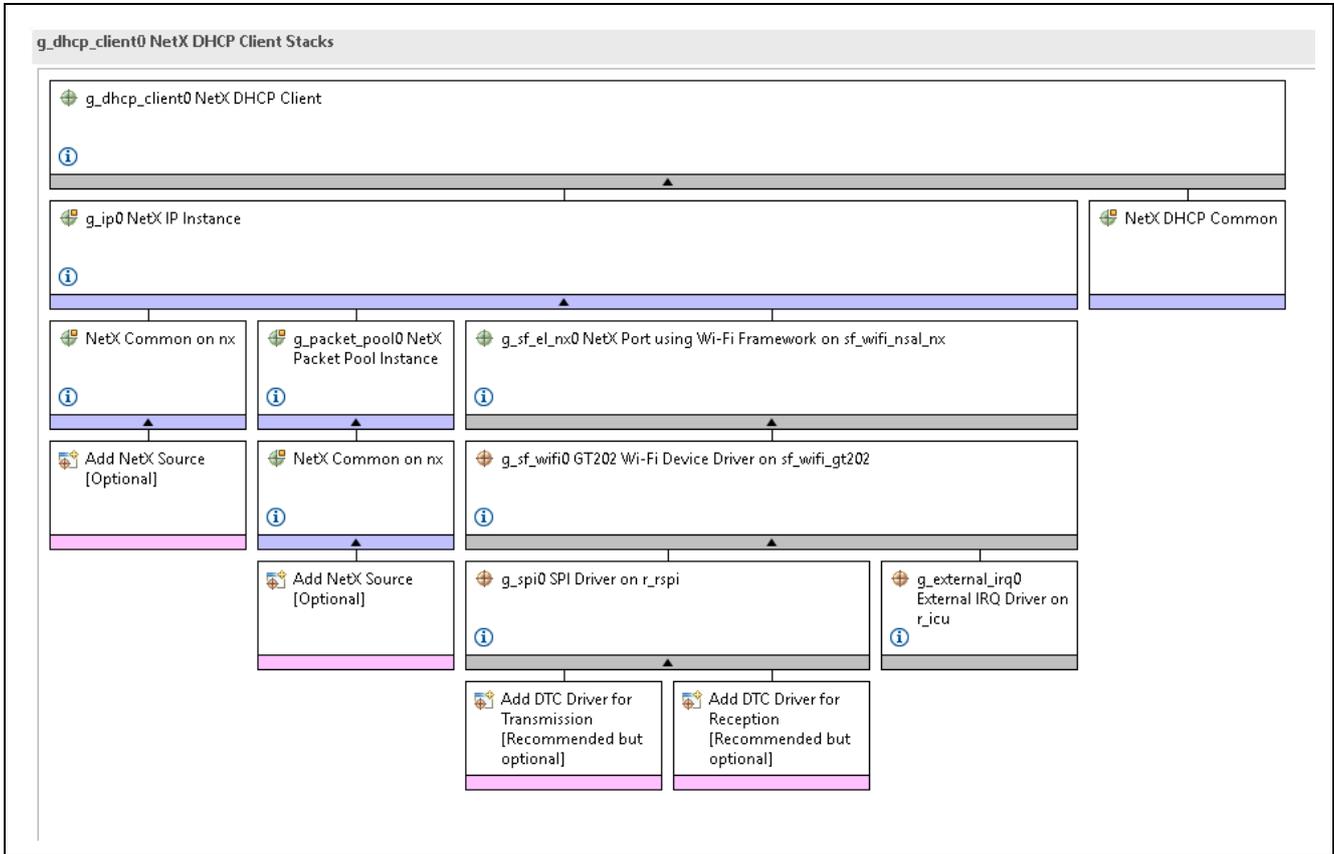


Figure 7. Snapshot of Final DHCP Client Configuration along with Wi-Fi Framework

Wi-Fi Framework and its associated components are added and configured with correct values. You can add the optional NetX source, in case you want to debug the project with NetX.

NetX component comes as a library in the SSP, but when the NetX source is added to the project, it gives the option to fine-tune the NetX stack to your requirements.

In addition to the NetX configuration, the packet pool configuration needs to be changed according to the project requirements. For this application, Figure 8 shows the packet pool size is selected as 4096 x 16.

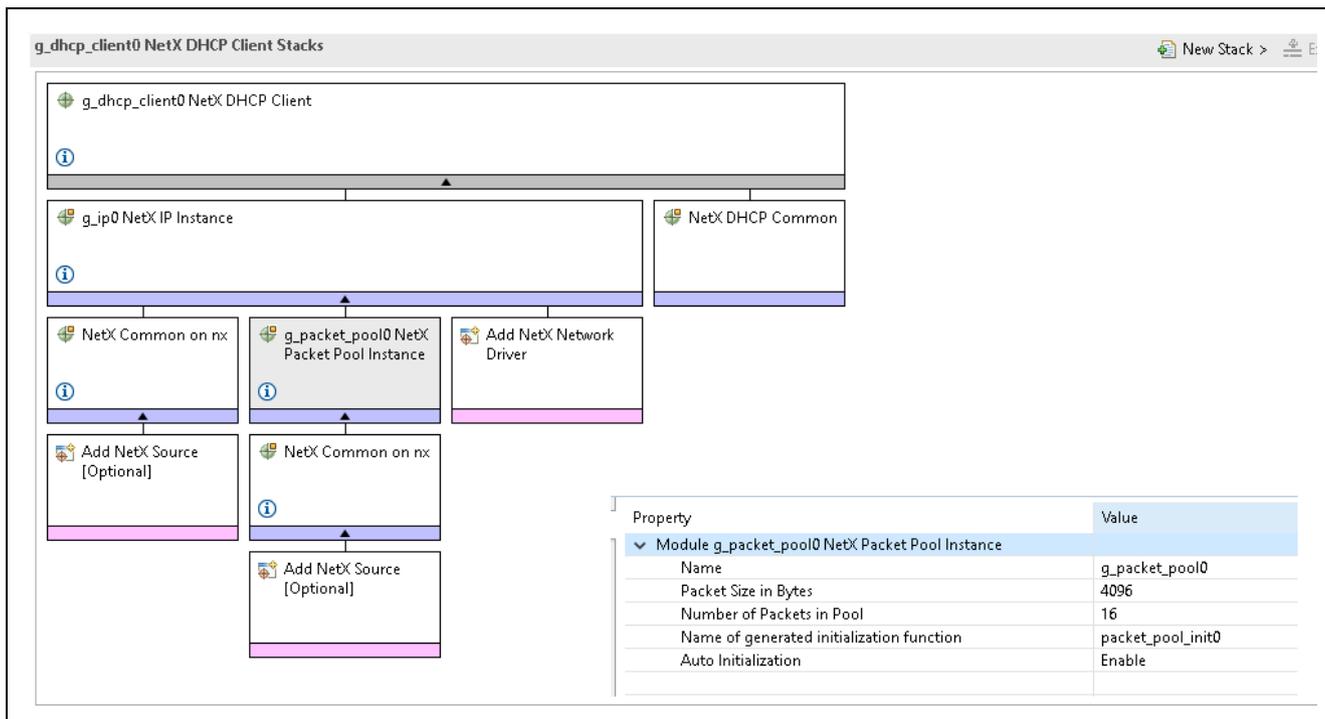


Figure 8. Changing the NetX Properties and Packet Pool Configuration

The actual configuration of the applications can also be viewed in the attached project using the ISDE configurator. Bringing in the NetX and ThreadX source to the project gives you options to debug the code with NetX and ThreadX code.

3.2 NetX HTTP Server

The application needs the HTTP server. Create a new thread and name it **HTTP server thread**. Add the HTTP server to the thread using the **New Stack > X-Ware > NetX > Protocols > NetX HTTP Server**. Figure 9 shows the HTTP server being added using the configurator.

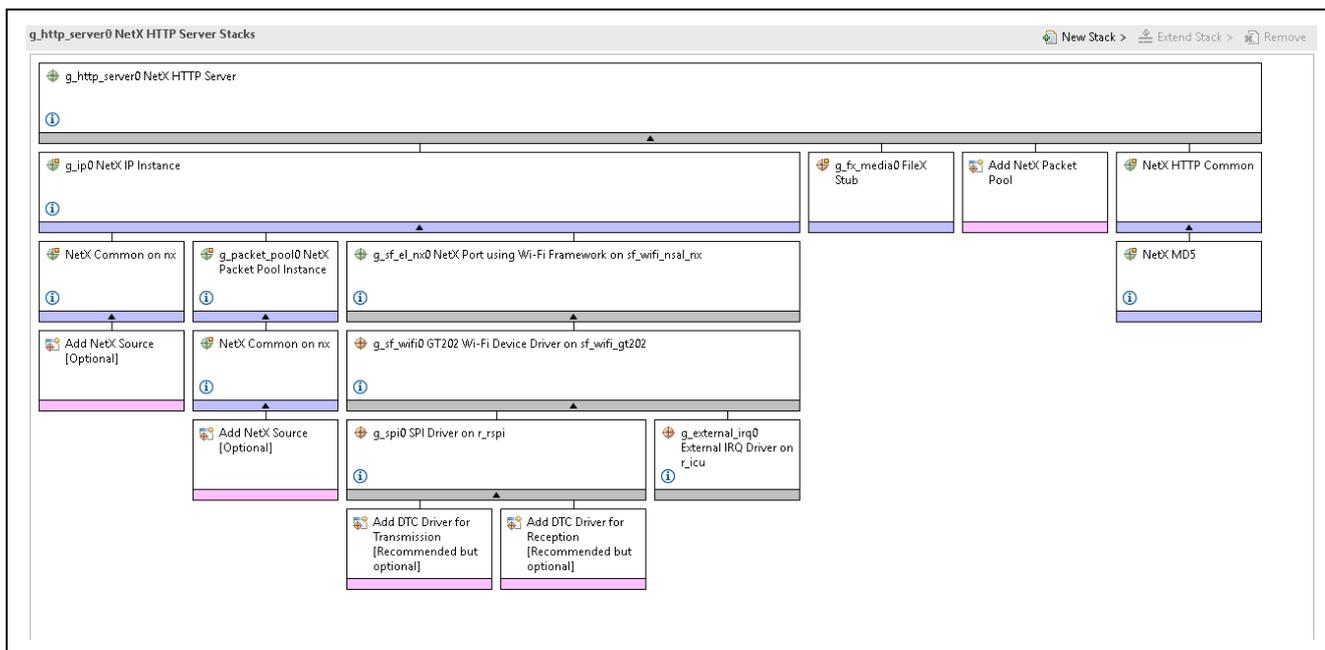


Figure 9. HTTP Server Added to HTTP Server Thread

3.3 LED Control

The application needs LED Blink Control. Create a new thread and name it **blink_thread**. Add event flag **blink_flag** to the thread. This thread is used to control the LED based on your input from the Web page. The associated code can be found in `blink_thread_entry.c`.

4. Wi-Fi Application Implementation Details

The Wi-Fi application is created on top of the Thermostat application (available for the PE-HMI1 Synergy MCU Kit). This section covers the Wi-Fi implementation details. Details on the Thermostat application are covered in a limited manner, as required for the application. For other details on the Thermostat application, see the *Thermostat Application Note* and the related project in the References section.

One of the reasons for choosing the Thermostat application is to showcase the modular approach to adding a new application on top of the existing application. The SSP framework gives a modular and uniform interface to add or remove the Wi-Fi module without changing the application. In addition, the application framework provides an approach to add/delete components/drivers based on the hardware modules. The current Wi-Fi application created for the GT202 module can be changed for other Wi-Fi modules by changing the add-on module in the configurator. This gives customers the flexibility to change the Wi-Fi module during the evaluation or pre-production stages.

For Figure 10, the dotted line shows the Thermostat application and the Wi-Fi application block. On the right side, the Wi-Fi application block is seen, consisting of the Wi-Fi application thread, HTTP server thread, and LED control thread. These three threads, and their thread objects/stacks, form the Wi-Fi application. A few modifications are done on the GUIX side (Wi-Fi Icon and IP Address Prompt) to add the Wi-Fi-specific additions to the GUIX.

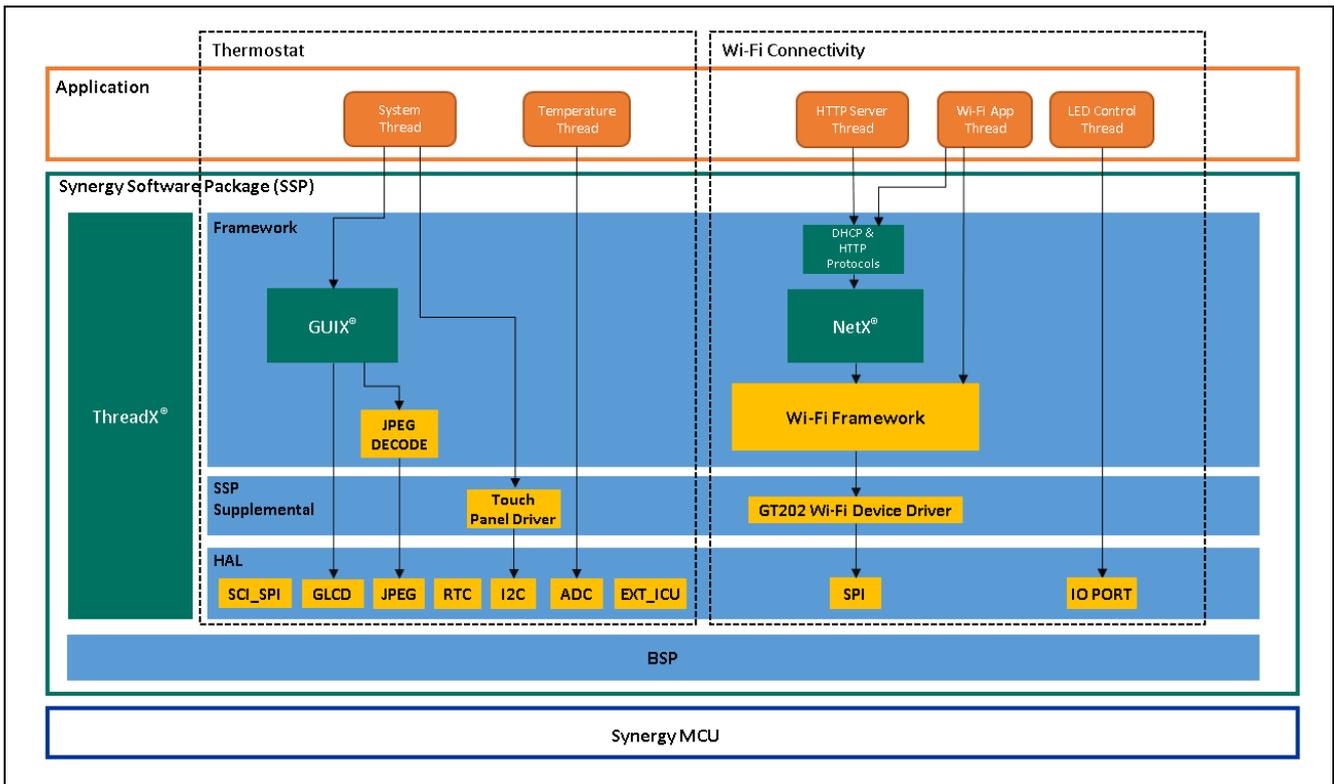


Figure 10. Synergy Wi-Fi Framework Application Architecture

The following sections detail the Wi-Fi application and its thread.

4.1 Wi-Fi Application Thread

This thread module, along with configurator created code, brings in the DHCP client application, with the TCP/IP core stack. In addition, it also brings the Wi-Fi Framework, NSAL, and Wi-Fi driver add-on. This thread is responsible for the DHCP client, Wi-Fi provisioning and connectivity.

The configurator-generated code is part of the `common_data.c/h` and `wifi_app_thread.c/h` in the `src/synergy_gen` folder. The code under these files are common code, specific to the thread's module stack components. In this case, the common code for NetX is available under `g_common_init()` function. Common code has the NetX driver entry function and the packet pool creation.

In your application code (`src/wifi_app_thread_entry.c`) added are, checking the status of the IP link creation, provisioning the Wi-Fi with your credentials, starting the DHCP client, and acquiring the IP address from the DHCP server.

Before provisioning the Wi-Fi, verify the IP link is enabled. The API to check the interface status is as follows:

```
(nx_ip_interface_status_check (&g_ip0, 0, NX_IP_LINK_ENABLED, &ip_status,
NX_WAIT_FOREVER)).
```

Once the IP link is enabled, the application can start provisioning the Wi-Fi using the Framework APIs with the credentials required by the Wi-Fi network to which you are intending to connect. The API to provision the Wi-Fi with your credentials is as follows. The parameter details can be found in the User's Manual or in the sample application code.

```
g_sf_wifi0.p_api->provisioningSet (g_sf_wifi0.p_ctrl, &g_provision_info);
```

After the successful provisioning of the Wi-Fi module, the board is ready to communicate to the DHCP server using the DHCP client. Your code needs to start the DHCP client (the DHCP client is created as part of the configurator generated code). The API for starting the DHCP client is part of your written code and its API is shown as follows:

```
nx_dhcp_start (&g_dhcp_client0);
```

After the DHCP client has started, it communicates with the DHCP server to get the IP address for the board. The following API and code is used to check if the IP address is resolved:

```
while (status != NX_IP_ADDRESS_RESOLVED)
{
    nx_ip_status_check(&g_ip0, NX_IP_ADDRESS_RESOLVED, (ULONG *) &status,
10);
}
```

Once the IP address is resolved, the DHCP client receives the IP address from the DHCP server, and you can get the leased IP address using the API:

```
nx_ip_interface_address_get (&g_ip0, 0, &ip0_ip_address, &ip0_mask);
```

The application is ready for TCP/IP communication with other clients on the Wi-Fi networks using the IP address.

4.2 HTTP Server Thread

This thread module, along with the configurator created code, part of `common_data.c/h` and `http_server_thread.c/h` under `src/synergy_gen` folder, brings in the HTTP server application.

The `authentication_check` and `get_notify` parameters are functions where your page creation and handling of the page specific Get/Set are handled. These are available in `src/http_server.c`.

To start the HTTP server, you can call the API `status = nx_http_server_start (&g_http_server0);`

This thread essentially creates the HTTP server and starts the server where the page you created will be hosted from the board. In addition to the page hosting, it gives the option to Get/Set your data from/on to the page. Additionally, the thread sets the event flag to indicate the LED control thread with your desired operation (ON/OFF/BLINK) for the LEDs on the board.

4.3 LED Control Thread

The thread mainly controls the LEDs based on the event received from you through the HTTP server page.

4.4 Code Organization

Figure 11 shows the sample directory structure of the Wi-Fi application code, the Wi-Fi Framework, its related driver, and code. The Wi-Fi Framework is an add-on listed under the `ssp_supplemental` folder.



Figure 11. Wi-Fi Framework Application, Framework Add-on Code and its Driver Directory Structure

5. Application Setup

Figure 12 shows the overview of the application and its test setup. The application is set up in client mode, connecting it to the Access Point or Router. Clients such as PCs/Smartphones can also be connected to the Access Point/Router. Once they all become part of the Wi-Fi LAN, they can communicate with the board and vice versa. This can be verified by pinging to the board's IP address from the PC. In this example, the web server running on the PE-HMI1 Synergy MCU board can be accessed using a web browser with the IP address displayed on the LCD Screen.

Important: In this example application project, the GT202 module mounted on the PE-HMI1 only supports 2.4 GHz bandwidth. So, the Access Point/Router should also be of the same bandwidth.

To run the application, pick the project **Synergy_WiFi_Framework_App_PE_HMI1_GT202.zip** to import from the included application bundle.

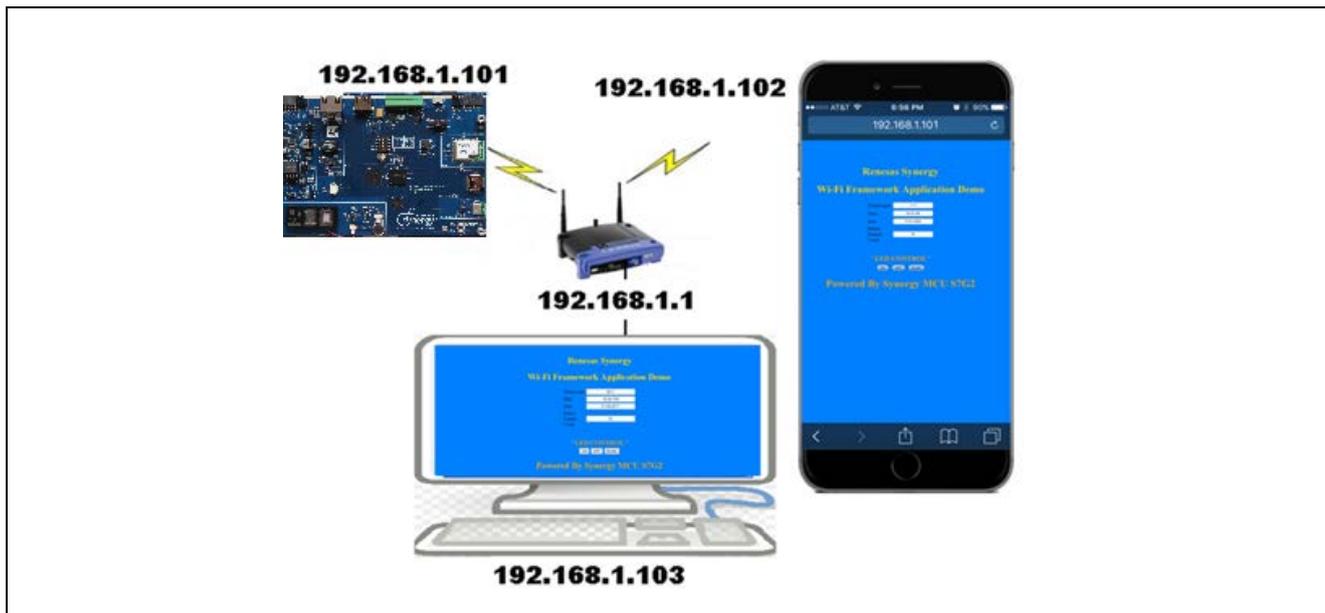


Figure 12. Wi-Fi Application Setup

5.1 Importing the Existing Project

Import the attached project **Synergy_WiFi_Framework_App_PE_HMI1_GT202.zip** for quick testing and demonstration. The procedure in the *Renesas Synergy™ Project Import Guide*, available as part of this application bundle (r11an0023eu0121_synergy_ssp.pdf) should be followed to import the project into e² studio ISDE and build and debug the project.

5.2 Modifying the User Credentials in the Existing Project

Your credentials (login details) required for the Wi-Fi network need to be hardcoded in the imported project to run the application on your Wi-Fi network.

Once the project is imported, make changes to **MY_SSID_FOR_AP** and **MY_PASSWORD_FOR_AP** in the **wifi_app_thread_entry.c**. Remove the default SSID and passcode and hardcode it to the correct SSID and password required by the AP/Router. The macro for the hardcoded SSID and Password are shown below:

```
#define MY_SSID_FOR_AP "Cisco36183" //Change this to your Wi-Fi AP/Router's SSID

#define MY_PASSWORD_FOR_AP "12345678" //Change this to your Wi-Fi AP/Router's Password
```

Note: The security type for the application is selected as WPA2. If you are running different security settings, then the changes need to be done in the code. The reference in the code can be found in the **wifi_app_thread_entry.c** `"g_provision_info.security = SF_WIFI_SECURITY_TYPE_WPA2;"`.

Once the changes are made to the project, build and download the image on to the board through the J-Link debugger. After the board is reset, the LCD display comes up with a splash screen. After a few seconds, you will see the main Thermostat screen.



Figure 13. LCD Screenshots of the Wi-Fi Thermostat Application

The notification bar on top left corner of the screen displays the Wi-Fi logo to indicate that the Wi-Fi is provisioned successfully; it also indicates the IP address of the board. Initially, the IP address of the board will be 0.0.0.0; later it gets updated with the IP address leased from the DHCP server. Figure 14 shows the screen with the updated IP address.

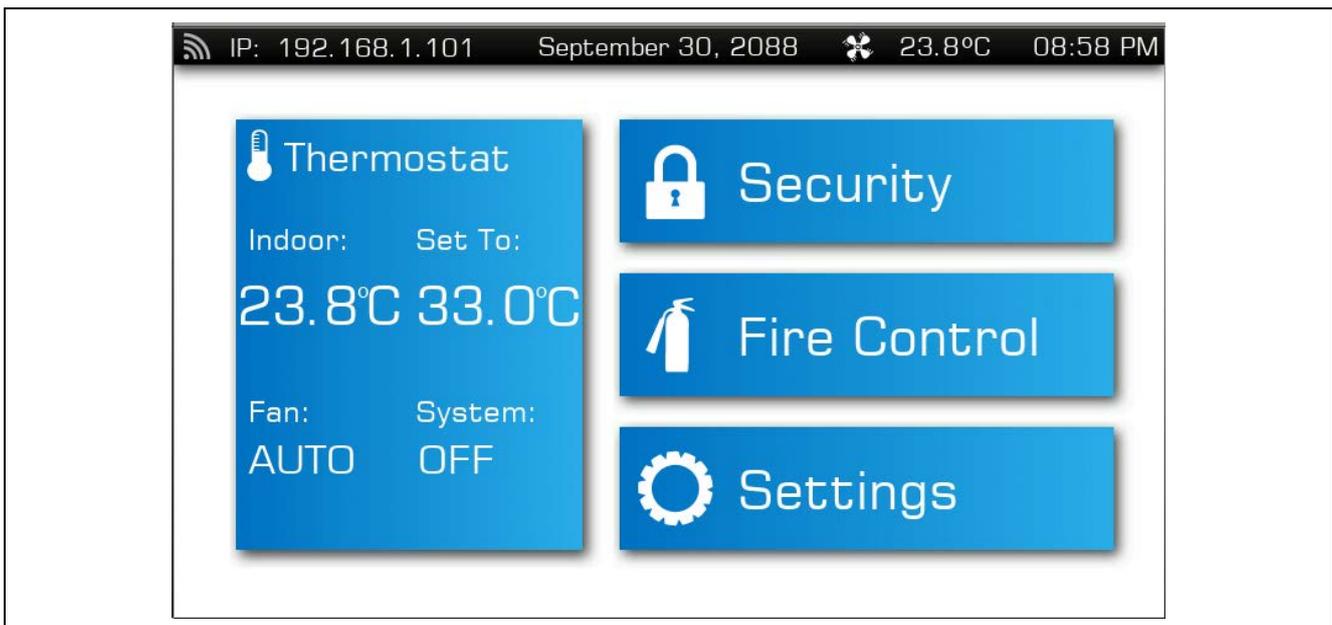


Figure 14. Notification Bar with IP Address

You can configure the time and date, using the settings icon on the thermostat screen.

You can verify the following items:

- Connectivity by pinging the board through a PC connected to the same Wi-Fi LAN (AP/Router Network).
- The Web Server application by typing the board's IP address on the browser of their PC/Phone/Tab connected to same Wi-Fi LAN (AP/Router).

Figure 15 shows the browser of the Wi-Fi application running on the PE-HMI1 board.

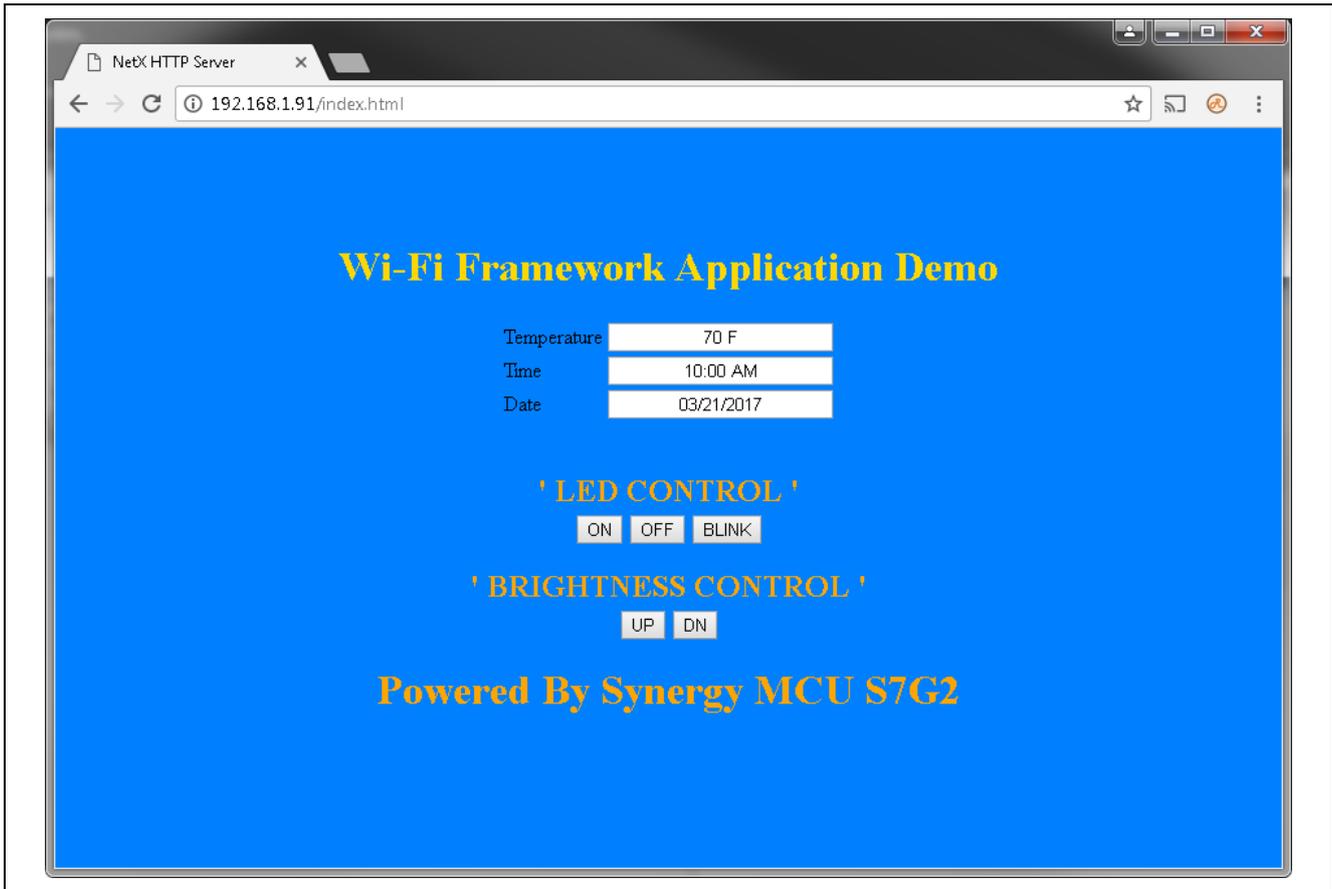


Figure 15. Web Page for PE-HMI1 Wi-Fi Framework Application

From the webpage, you can set:

- Brightness of the LCD Screen using the brightness control UP/DOWN.
- LED to BLINK, OFF, or ON. The user LED is located on the back side of the PE-HMI1 board.

From the webpage, you can get:

- Temperature of the CPU.
- Time and Date information from the RTC running on the board.

The LED is located on the back side of the board. In Figure 3 the LED1 is as shown in the highlighted area of the PE-HMI1 board. Figure 16 shows a close-up view of the LED.



Figure 16 PE-HMI1 User LED1

6. Next Steps

- Visit www.renesas.com/synergy/solutionsgallery to learn more about Development Tools and Utilities for Synergy.
- Visit www.renesas.com/synergy/tools to download development tools & utilities.
- Renesas Synergy Module Guides collateral link: www.renesas.com/synergy/tools-kits.

7. References

1. SSP User's Manual: www.renesas.com/synergy/ssp
2. PE-HMI1 Product Example: www.renesas.com/en-us/products/synergy/hardware/kits/pe-hmi1.html
3. PE-HMI1 User's Manual: www.renesas.com/en-us/doc/products/renesas-synergy/doc/r12um0005eu0100_synergy_pe_hmi1.pdf
4. PE-HMI1 Quick Start Guide: www.renesas.com/en-us/doc/products/renesas-synergy/doc/r12qs0002eu0101-synergy-pe-hmi1.pdf
5. Getting Started with Thermostat Application for S7G2 and S5D9: www.renesas.com/us/en/software/D6003611.html
6. Renesas Synergy Knowledge Base: www.renesas.com/synergy/knowledgebase
7. Renesas Synergy™ Project Import Guide (r11an0023eu0121-synergy-ssp-import-guide.pdf) included in this package)

Website and Support

Visit the following vanity URLs to learn about key elements of the Synergy Platform, download components and related documentation, and get support.

Synergy Software	www.renesas.com/synergy/software
Synergy Software Package	www.renesas.com/synergy/ssp
Software add-ons	www.renesas.com/synergy/addons
Software glossary	www.renesas.com/synergy/softwareglossary
Development tools	www.renesas.com/synergy/tools
Synergy Hardware	www.renesas.com/synergy/hardware
Microcontrollers	www.renesas.com/synergy/mcus
MCU glossary	www.renesas.com/synergy/mcuglossary
Parametric search	www.renesas.com/synergy/parametric
Kits	www.renesas.com/synergy/kits
Synergy Solutions Gallery	www.renesas.com/synergy/solutionsgallery
Partner projects	www.renesas.com/synergy/partnerprojects
Application projects	www.renesas.com/synergy/applicationprojects
Self-service support resources:	
Documentation	www.renesas.com/synergy/docs
Knowledgebase	www.renesas.com/synergy/knowledgebase
Forums	www.renesas.com/synergy/forum
Training	www.renesas.com/synergy/training
Videos	www.renesas.com/synergy/videos
Chat and web ticket	www.renesas.com/synergy/resourcelibrary

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Feb.12.16	—	Initial version
1.01	Jun.17.16	—	Specified SSP v1.0.0
1.02	Nov.18.16	—	Minor formatting changes
1.03	Apr.03.18	—	Update for SSP v1.4.0
1.04	May.07.19	—	Updated for SSP v1.6.0. Added note for 2.4 GHz Router.
1.05	Oct.11.19	—	Updated for SSP v1.7.0

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
 - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
 - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

Corporate Headquarters

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/.