Renesas Synergy™ Platform

Wi-Fi Framework Module Guide

R11AN0252EU0100
Rev.1.00
Aug 20, 2018

Introduction

This Module Guide will enable you to effectively use a module in their own design. On completion of this guide, you will be able to add this module to your own design, configure it correctly for the target application and write code, using the included Application Project code as a reference and an efficient starting point. References to more detailed API descriptions and suggestions of other Application Projects that illustrate more advanced uses of the module are available in the Renesas Synergy Knowledge Base, as described in the References section at the end of this document, and should be valuable resources for creating more complex designs.

Contents

1. Wi-Fi Framework Module Overview

The Wi-Fi Framework provides high-level APIs for configuring and provisioning Wi-Fi modules as well as performing data transfers with or without on-chip networking capability. Currently, only the Qualcomm GT202 module is supported. The Wi-Fi Framework communicates through SPI with the underlying GT202 module.

1.1 Wi-Fi Framework Module Features

The Wi-Fi Framework module has the following features:

- Provides high-level APIs to configure and provision a Wi-Fi module
- Provides four different implementations for:
  - A Wi-Fi device driver stack using the sf_wifi_gt202 framework
  - An on-chip stack using the sf_wifi_onchip_stack framework
  - A BSD socket stack using the sf_wifi_onchip_stack framework
  - A NetX and NetX Duo port using the sf_wifi_nsal_nx framework
- Using NetX and NSAL
  - Allows the same application code to be used across different Wi-Fi modules
  - Allows easy migration of the Ethernet-based application to a Wi-Fi based application
  - Allows for debugging and fine-tuning the application and TCP/IP stack as required by the application
  - The current NSAL implementation only provides NetX NSAL. Adding support for a new network stack requires implementing the appropriate NSAL.
- Using the on-chip networking stack
  - Is beneficial when using MCUs with a small memory footprint
  - Provides a BSD sockets interface to create socket-based applications with the on-chip TCP/UDP
  - Provides an option to integrate 3rd-party application protocols on top of TCP/IP such as MQTT and COAP without using the NetX stack.

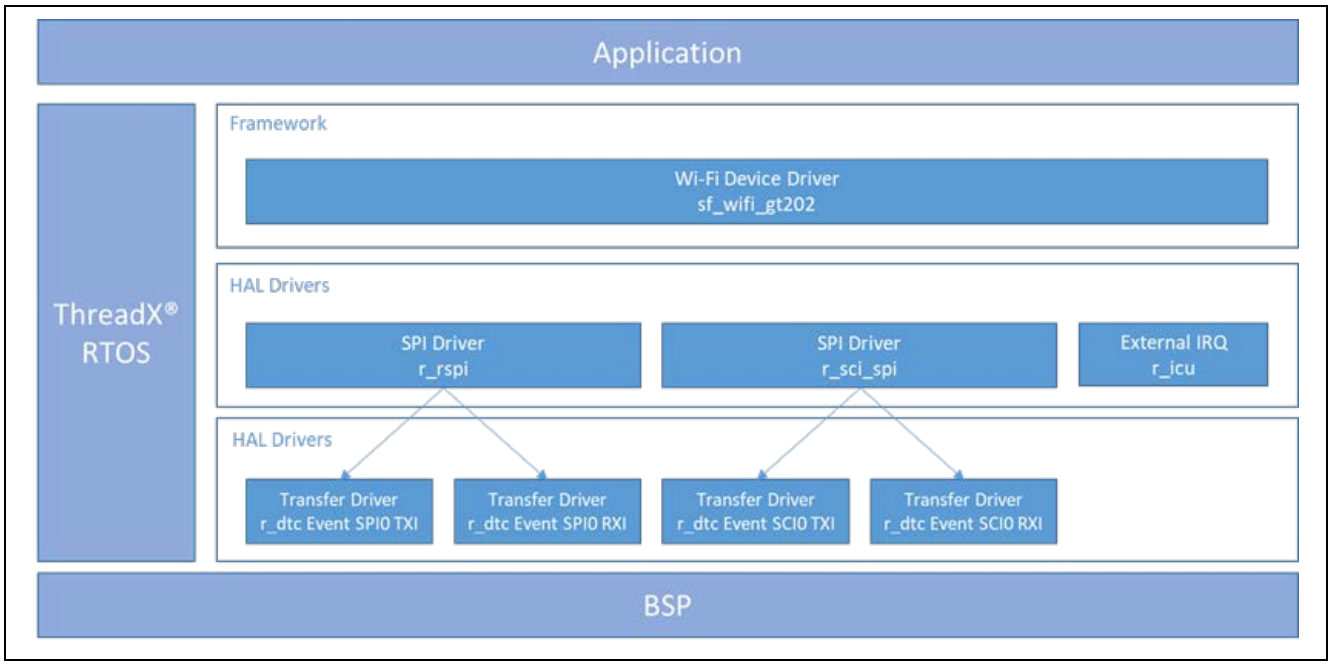

Figure 1 Wi-Fi Device Driver Implementation

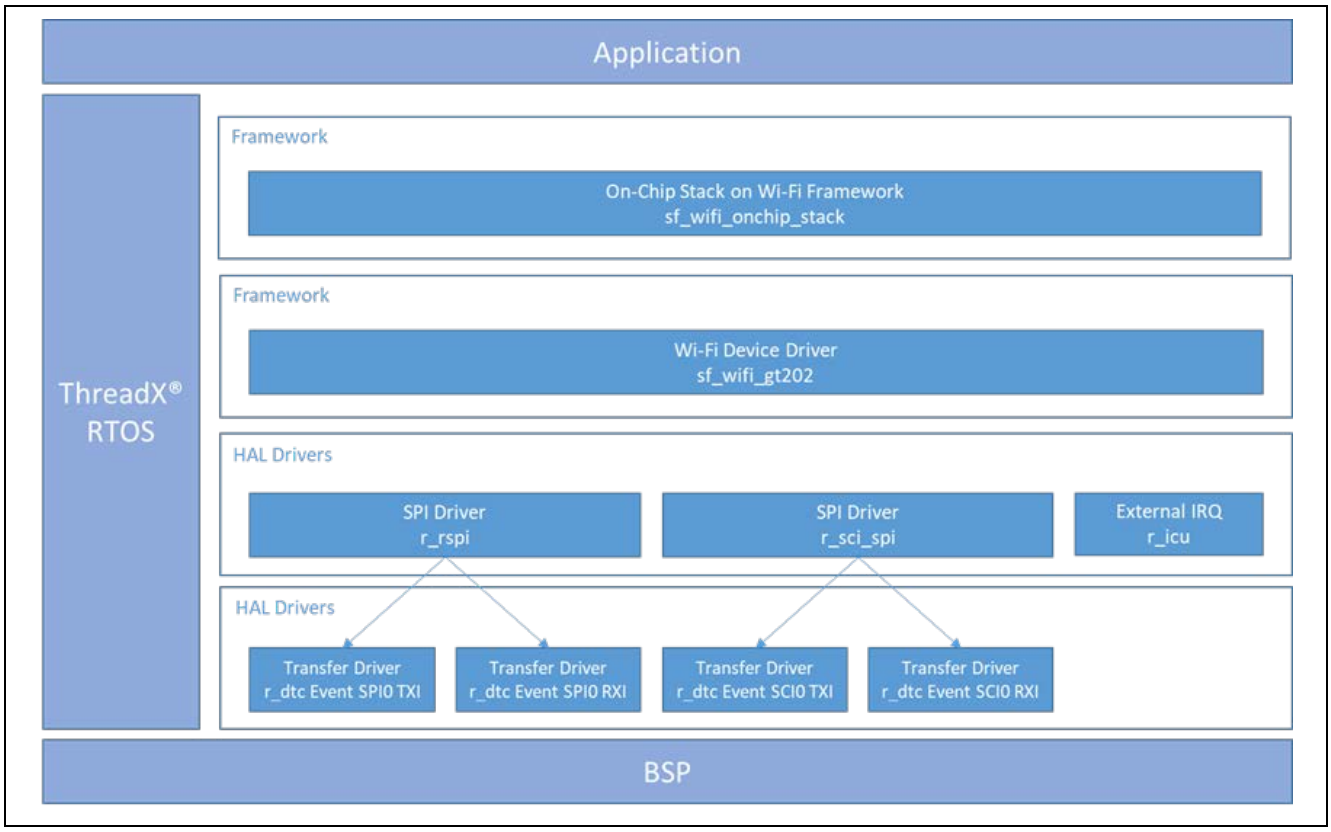

Figure 2 Wi-Fi On-Chip Stack Implementation

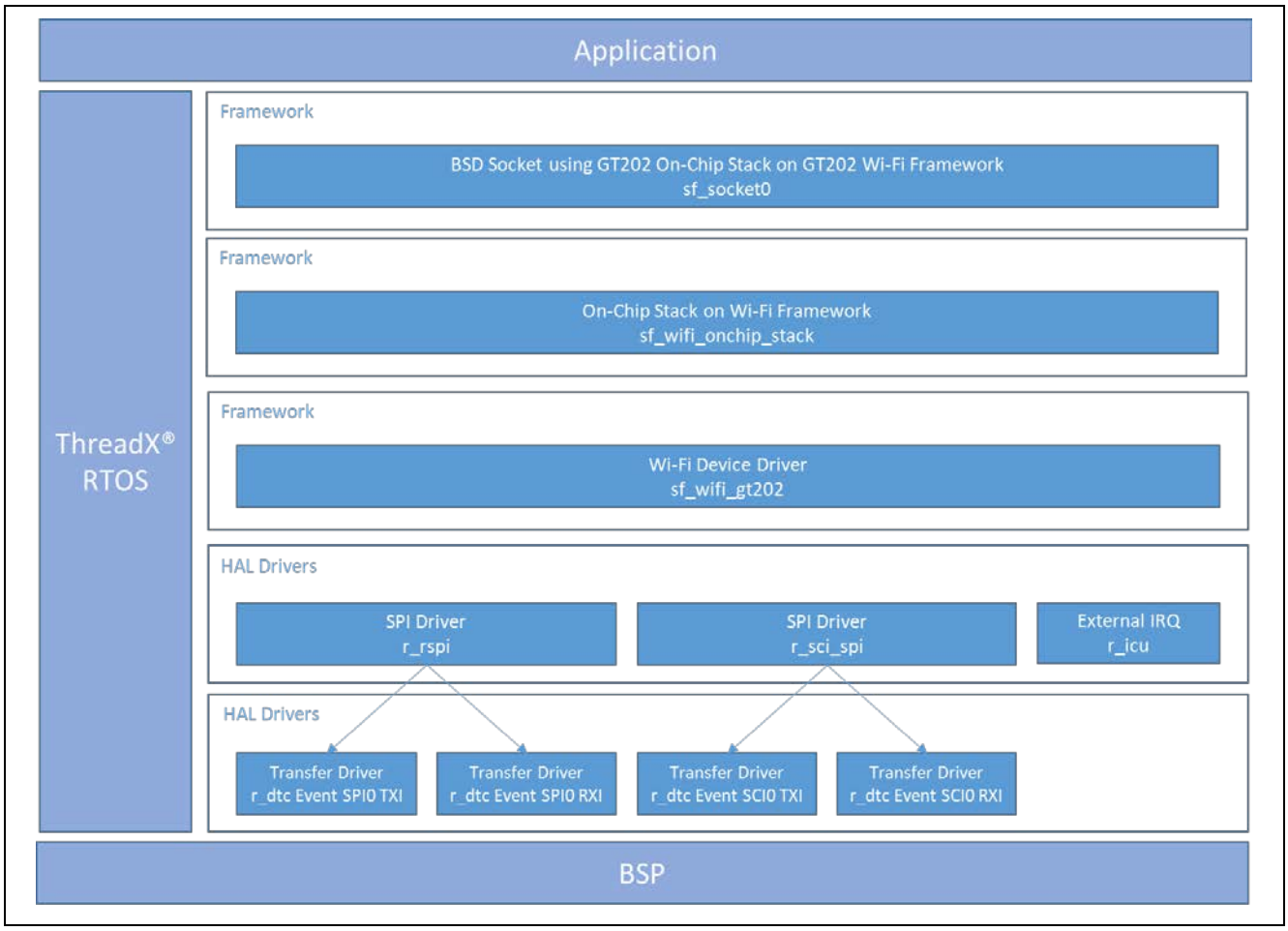

Figure 3 Wi-Fi BSD Socket Implementation

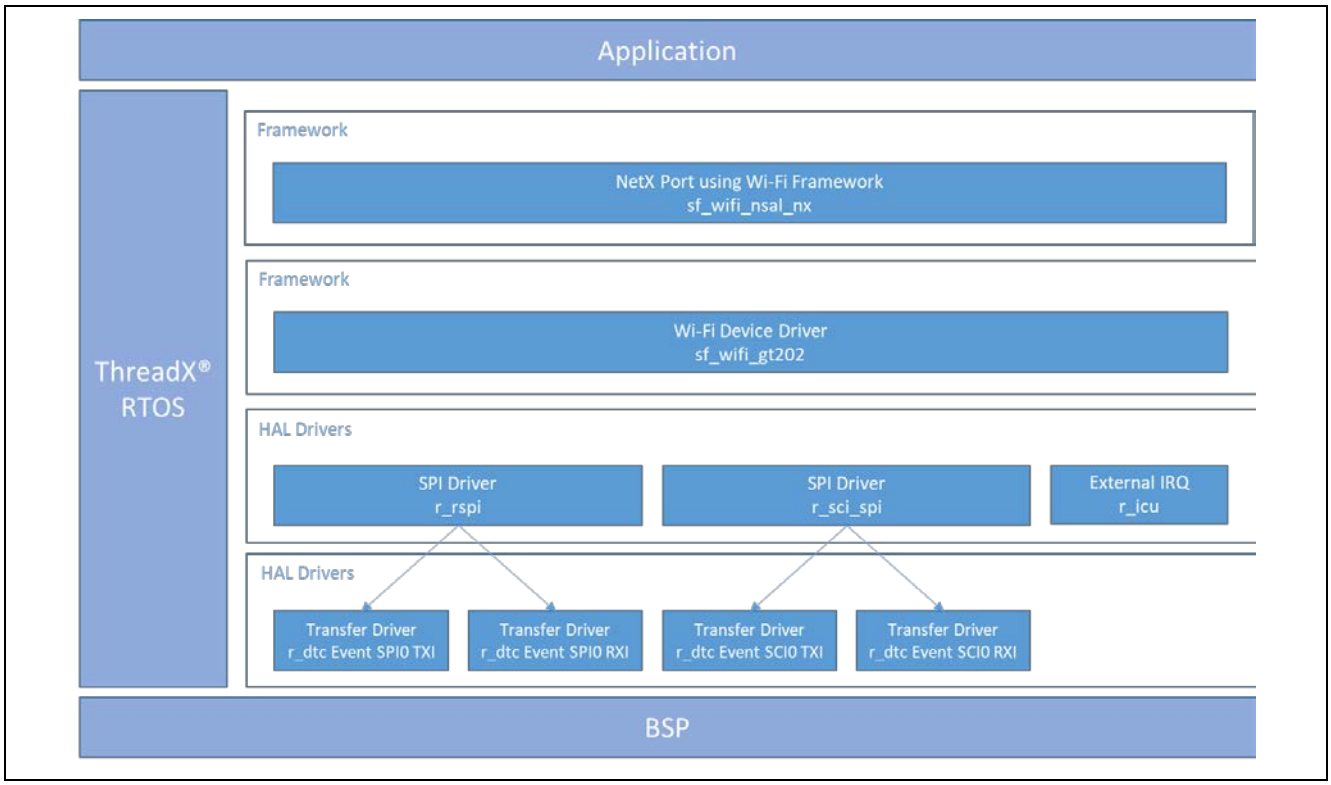

Figure 4 Wi-Fi NetX Port Implementation with NSAL

2. Wi-Fi Framework Module APIs Overview

The Wi-Fi Framework defines APIs for each of the related modules. The following descriptions explain the operation of each API. A table of common return codes follows at the end of the section. Refer to the API reference section for the associated module for additional details.

Additionally, a more detailed API description, along with a working example application (which is too lengthy to include in this document), is available on the Renesas web site. Search for the associated application note document number, r11an0226eu, in search bar on www.Renesas.com. It is highly recommended that you use the application note to augment the summary descriptions found in this document.

2.1 Wi-Fi Framework Module APIs

The Wi-Fi Framework module provides a high-level interface for the network stack and applications irrespective of the Wi-Fi module. Below are the APIs exposed by the framework.

Table 1 Wi-Fi Framework Module API Summary

Function Name	Example API Call and Description
`.open`	`g_sf_wifi0.p_api->open (g_sf_wifi0.p_ctrl, g_sf_wifi0.p_cfg);` This API initializes and enables the Wi-Fi module. The open function returns the Wi-Fi control structure, uniquely identifying the instance of the Wi-Fi Framework.
`.close`	`g_sf_wifi0.p_api->close(g_sf_wifi0.p_ctrl);` This API un-initializes the Wi-Fi module and powers it off.
`.infoGet`	`g_sf_wifi0.p_api->infoGet (g_sf_wifi0.p_ctrl, wifi_info);` This API takes the Wi-Fi control structure as an argument and returns the following information obtained from the Wi-Fi module: • Chipset/driver information string • RSSI value (unsigned integer 16 bits) • Noise level (unsigned integer 16 bits) • Link Quality (unsigned integer 16 bits)

`.statisticsGet`	`g_sf_wifi0.p_api->statisticsGet (g_sf_wifi0.p_ctrl, p_stats);` This API gets the data statistics from the Wi-Fi module. It takes the Wi-Fi control structure as an argument and returns the following statistics: • Received packets (unsigned integer 32 bits) • Transmitted packets (unsigned integer 32 bits) • Transmit packet errors (unsigned integer 32 bits)
`.transmit`	`g_sf_wifi0.p_api->transmit (g_sf_wifi0.p_ctrl, p_buffer, length);` This API sends the data/packet out. This function takes the Wi-Fi control structure as an argument. It takes the network packet buffer, and the network packet buffer length as arguments. The Wi-Fi Framework transmit function passes the packet buffer to the Wi-Fi driver for transmission.
`.provisioningGet`	`g_sf_wifi0.p_api->provisioningGet (g_sf_wifi0.p_ctrl, &sf_wifi_provisioninfo);` This API takes the Wi-Fi control structure as an argument and returns the following parameters: • Mode (enumeration, that is, AP or client) • Channel (unsigned integer 8 bits) • SSID (string) • Security type (enumeration) • Encryption type (enumeration) • Security key (string)
`.provisioningSet`	`g_sf_wifi0.p_api->provisioningSet (g_sf_wifi0.p_ctrl, &g_sf_wifi_provisioninfo);` This API sets the Wi-Fi module in the given mode AP/Station. The provisioningSet function uses the following parameters to provision the Wi-Fi module: • Mode (enumeration, that is, AP or client) • Channel (unsigned integer 8 bits), only used in AP mode • SSID (string) • Security type (enumeration) • Encryption type (enumeration) • Security key (sting) • Connection Status Notification Callback function: Used to get connection status change notification *See note at the end of this table.
`.scan`	`g_sf_wifi0.p_api->scan (g_sf_wifi0.p_ctrl, p_scan, p_count);` This API scans the available SSIDs (that is, access points) in range. This function takes the Wi-Fi control structure as an argument and returns a list of SSIDs scanned by the Wi-Fi module with the following parameters: • HW mode (enumeration a/b/g/n) • RSSI (unsigned integer 16 bits) • SSID (string) • BSSID (byte array of size 6 bytes) • Channel (unsigned integer 8 bits) • Security type (enumeration) • Encryption type(enumeration) • BSS type (enumeration) The Wi-Fi Framework scan function takes the SSID count as an argument, which acts as an in/out parameter. It specifies the size of the scan result array and the Wi-Fi Framework sets it to count the indicating number of scan results stored in the array.

.ACLAdd	g_sf_wifi0.p_api->ACLAdd (g_sf_wifi0.p_ctrl, peer_device_mac); This API adds the given MAC address to the access control list. This function takes the Wi-Fi control structure and the MAC address as arguments.
.ACLDelete	g_sf_wifi0.p_api->ACLDelete (g_sf_wifi0.p_ctrl, peer_device_mac); This API deletes the given MAC address from the access control list. This function takes the Wi-Fi control structure and MAC address as arguments.
.multicastListAdd	g_sf_wifi0.p_api->multicastListAdd (g_sf_wifi0.p_ctrl, p_mac_addr); This API adds the given Multicast IP address to the multicast filer list. This function takes the Wi-Fi control structure and MAC address as arguments.
.multicastListDelete	g_sf_wifi0.p_api->multicastListDelete (g_sf_wifi0.p_ctrl, p_mac_addr); This API deletes the given Multicast IP address from the multicast filer list. This function takes the Wi-Fi control structure and MAC address as arguments.
.macAddressGet	g_sf_wifi0.p_api->macAddressGet (g_sf_wifi0.p_ctrl, p_mac); This API reads MAC address from Wi-Fi module. This function takes the Wi-Fi control structure as argument and returns MAC address read from Wi-Fi module.
.macAddressSet	g_sf_wifi0.p_api->macAddressSet (g_sf_wifi0.p_ctrl, p_mac); This API sets Wi-Fi module's MAC address. This function takes the Wi-Fi control structure and MAC address as arguments.
.versionGet	g_sf_wifi0.p_api->versionGet (&version); Retrieves the API version with the version pointer.

When the device is provisioned in client mode, callback will be called when connection with AP is lost and established back. When the device is provisioned in AP mode, this callback is called when any client connects/disconnects with AP. When the device is provisioned in client mode, arguments passed to callback will have only the following valid field,

- event = SF_WIFI_EVENT_AP_CONNECT or SF_WIFI_AP_DISCONNECT

When the device is provisioned in AP mode, then arguments passed to callback will have only the following valid fields:

- event = SF_WIFI_EVENT_CLIENT_CONNECT or SF_WIFI_CLIENT_DISCONNECT
- mac_addr = MAC address of client.

Note: While calling the `provisioningSet()` API to provision the device in client mode, the framework will not call the callback function on successful association with AP or on failure. When the device is provisioned in AP mode, if the client tries to connect with the AP using wrong password then callback will be called twice, first with connected event and then immediately after this with disconnected event.

2.2 On-chip Networking Stack Support APIs

These APIs can be used to configure the Wi-Fi module when using an on-chip networking stack, which helps to configure the IP address for the interface, and start/stop DHCP server (when configured in the AP mode).

Table 2 On Chip Networking Stack Support Wi-Fi Framework Module API Summary

Function Name	Example API Call and Description
.open	`g_sf_wifi_onchip_stack0.p_api->open (g_sf_wifi_onchip_stack0.p_ctrl, g_sf_wifi_onchip_stack0.p_cfg);` This API calls the Wi-Fi Framework open API which initializes the Wi-Fi module.
.close	`g_sf_wifi_onchip_stack0.p_api->close(g_sf_wifi_onchip_stack0.p_ctrl);` This API calls the Wi-Fi Framework close API which un-initializes the Wi-Fi module.
.ipAddressCfg	`g_sf_wifi_onchip_stack0.p_api->ipAddressCfg (g_sf_wifi_onchip_stack0.p_ctrl, p_cfg);` This API configures the IP address of the interface using an on-chip networking stack. It provides facility to configure static IP address or using DHCP.
.dhcpServerStart	`g_sf_wifi_onchip_stack0.p_api->dhcpServerStart (g_sf_wifi_onchip_stack0.p_ctrl, p_start_ip, p_end_ip);` This API starts the DHCP server on the interface (when configured in AP mode) using on-chip networking stack. It takes the range of IP addresses to be used by DHCP server.
.dhcpServerStop	`g_sf_wifi_onchip_stack0.p_api->dhcpServerStop (g_sf_wifi_onchip_stack0.p_ctrl);` This API stops the DHCP server.
.versionGet	`g_sf_wifi_onchip_stack0.p_api->versionGet (&version);` Retrieves the API version with the version pointer.

2.3 BSD Socket APIs

These APIs can be used for BSD socket support using the GT202 on-chip stack implementation.

Table 3 BSD Socket using GT202 On-Chip Stack Wi-Fi Framework Module API Summary

Function Name	Example API Call and Description
.open	`g_sf_socket0.p_api->open (g_sf_socket0.p_ctrl, g_sf_socket0.p_cfg);` This API initializes the networking interface.
.close	`g_sf_socket0.p_api->close(g_sf_socket0.p_ctrl);` This API closes the network interface.
.versionGet	`g_sf_socket0.p_api->versionGet (&version);` Retrieves the API version with the version pointer.

Additionally, this implementation includes socket APIs which are compliant with BSD APIs. These APIs can be used by the application to perform data transfer using sockets. The following APIs are available:

- socket
- close
- bind
- listen
- accept
- connect
- send
- recv
- recvfrom
- sendto
- setsockopt
- getsockopt
- select

Note: While using on chip networking stack, the application can use all BSD Socket APIs, all on chip Networking Stack support APIs and few Synergy Wi-Fi Framework APIs. The Synergy Wi-Fi Framework APIs which the application can use are `provisioningSet()`, `provisioningGet()`, `scan()`, `macAddressGet()`, `macAddressSet()` and `infoGet()`.

More information is available for these APIs as described in the *NetX BSD 4.3 Sockets API Compliancy Wrapper for NetX User Guide*, which can be found on the Synergy Gallery, on the SSP page, under the documentation tab, in the X-Ware ™ Component Documents for Renesas Synergy zip file.

2.4 Wi-Fi NSAL

The Synergy Wi-Fi Framework supports the NetX/NetX-Duo Network Services Abstraction Layer. This includes the NetX/NetX-Duo driver, packet transmit and receive callback functions implementation.

NetX/NetX-Duo Driver Function

The NetX/NetX-Duo driver function takes the NetX IP instance, Wi-Fi Framework instance and NSAL configuration as arguments. The NSAL configuration controls the behavior of transmit and receive callback functions. The NSAL configuration includes flags which indicate that zero-copy support is enabled or disabled in transmit and receive path. The NetX/NetX-Duo driver functions implement various IP driver commands used by NetX/NetX-Duo. The interface attach command calls the Wi-Fi Framework `open` API to initialize the Wi-Fi module. The initialize command calls the Wi-Fi Framework `macAddressGet` API to read MAC address from the Wi-Fi module. The un-initialize command calls the Wi-Fi Framework `close` API which de-initializes the Wi-Fi module. The multicast-join command calls the Wi-Fi Framework `multicastListAdd` API to add the given MAC address to multicast list. The multicast-leave command calls the Wi-Fi Framework `multicastListDelete` API to delete the given MAC address from the multicast list. The Send/Broadcast command calls the Wi-Fi Framework `transmit` API to transmit a packet.

NSAL Transmit Function

The NSAL transmit function takes the NetX IP instance, the NetX packet, the Wi-Fi Framework instance, and the NSAL configuration as arguments. If zero-copy support is enabled, then the same NetX packet is transferred from NetX to the Wi-Fi driver. If zero-copy is not supported, then it copies data from the NetX packet to the driver buffer. It calls the Wi-Fi Framework `transmit` API, which passes the buffer/packet to the Wi-Fi driver for further transmission.

NSAL Receive Callback

The NSAL receive callback function takes the NetX IP instance, the packet buffer, the packet buffer length, and the NSAL configuration as arguments. This callback is called from the Wi-Fi device driver. If zero-copy support is enabled, then the same NetX packet is transferred from the Wi-Fi driver to NetX. If zero-copy is not supported, then it copies data from the driver buffer to the NetX packet and then passes the NetX stack for further processing. It calls the Wi-Fi Framework `transmit` API, which passes the buffer to the Wi-Fi driver for further transmission.

More information is available for these APIs as described in the *NetX User Guide*, which can be found on the Synergy Gallery, on the SSP page, under the documentation tab, in the X-Ware ™ Component Documents for Renesas Synergy zip file.

2.5 Wi-Fi Framework Error Codes

The following table lists the Wi-Fi Framework specific error codes. These error codes are part of ssp_err_t.

Table 4 Wi-Fi Framework Error Codes

Error Codes	Description
SSP_ERR_WIFI_CONFIG_FAILED	Configuration failed
SSP_ERR_WIFI_INIT_FAILED	Initialization failed
SSP_ERR_WIFI_TRANSMIT_FAILED	Transmission failed
SSP_ERR_WIFI_INVALID_MODE	Invalid mode specified
SSP_ERR_WIFI_FAILED	Wi-Fi failed

3. Wi-Fi Framework Module Operational Overview

The following operational overview is a summary of the operation of the Wi-Fi module. A more detailed description, along with a working example application (which is too lengthy to include in this document), is available on the Renesas web site. Search for the associated application note document number, r11an0226eu, in the top page search bar on www.Renesas.com. It is highly recommended that you use the application note to augment the summary descriptions found in this document.

3.1 Wi-Fi Framework Module Operational Introduction

The Wi-Fi Framework provides a high-level interface for the application to configure the Wi-Fi module, provision the Wi-Fi module, and perform data transfers. This simplifies application development and allows the same application code to be used across different Wi-Fi modules.

The following figure provides an overview of the Synergy Wi-Fi Framework layered architecture:

- The Wi-Fi Framework includes the enclosed 5 blocks in the middle of the architecture graph: NSAL, Wi-Fi Framework API, Wi-Fi on-chip stack API, BSD Socket API, and the Wi-Fi Device Driver Interface.
- The vendor-provided Wi-Fi device drivers are included in the SSP package under SSP Supplemental.

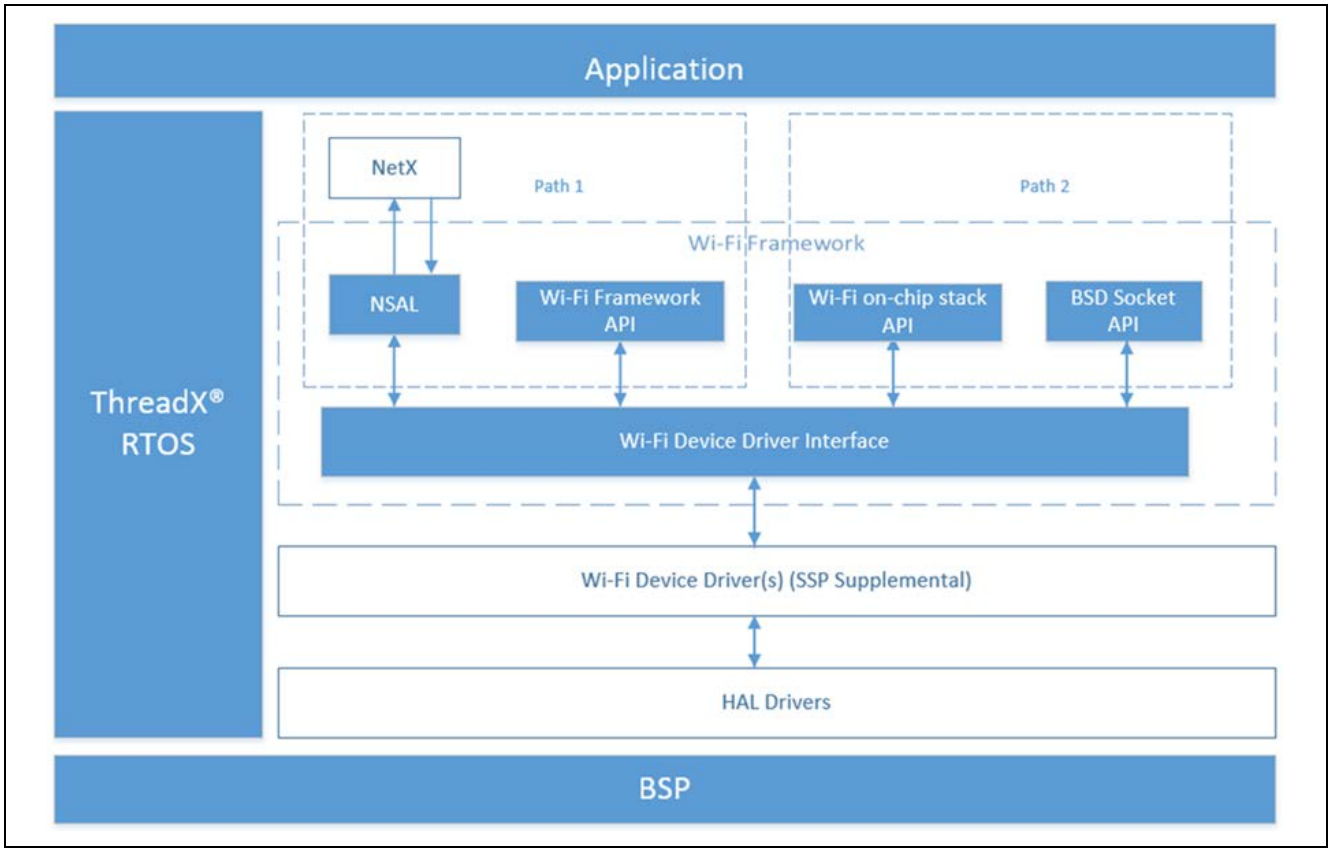

Figure 5 Wi-Fi Framework organization, options and stack implementation

The Wi-Fi Framework implementation allows Wi-Fi modules with or without on-chip networking stack support to be integrated with the SSP low-level support blocks.

- Path 1: Using NetX™/NetX Duo™, NSAL in addition to the Wi-Fi Framework APIs blocks as shown in the previous figure. To simplify the description, throughout this document, NetX refers to both NetX and NetX-Duo when the comment applies to both.
- Path 2: Using on-chip networking stack support API and the BSD Socket APIs as shown in the previous figure.

3.2 Wi-Fi Framework Module Operational Notes and Limitations

3.2.1 Wi-Fi Framework Module Operational Notes

- The Wi-Fi module has various parameters as specified by 802.11 standards. It is possible that individual device drivers and Wi-Fi chipsets might not support the configuration of all the functions.
- For the Wi-Fi interface to become active, at least the channel, Service Set Identifier (SSID), security scheme, and security credentials must be configured.
- The current NSAL implementation includes support for NetX (IPv4) and NetX-Duo (IPv6). NetX and NetX Duo support IPv4, however NetX Duo also supports IPv6. Adding support for a new network stack requires implementing the appropriate NSAL.
- For the security setting, WEP keys can be entered in either ASCII or Hex format and can be configured to use either 40- or 104-bit keys. A WEP key has a 24-bit initialization vector in addition to the secret key. Because of this and depending on the vendor, 64-bit WEP keys can be referred to as 40-bit keys and 128-bit WEP keys can be referred to as 104-bit keys. The Wi-Fi Framework accepts 1 to 4 WEP keys of a specific format and type. In the provisioning structure, you must fill in the security type as SF_WIFI_SECURITY_TYPE_WEP and at least one (maximum is four) WEP key in the key buffer.

3.2.2 Wi-Fi Framework Module API Use Notes

Wi-Fi Framework Module APIs

- Open

  When using the Wi-Fi Framework module with NSAL, that is, with NetX/NeX-Duo, the application should not call the Wi-Fi Framework module `open()` API directly; instead it should call the NetX `nx_ip_create()` API which internally calls `open()` from the NetX driver.

  When using the on-chip networking stack, the application should call the `open()` API from the BSD socket interface, which internally calls the Wi-Fi Framework `open()` API.
- Close

  When using the Wi-Fi Framework module with NSAL that is, with NetX, the application should not call the Wi-Fi Framework module `close()` API directly; instead it should call the NetX `nx_ip_delete()` API which internally calls `close()` from the NetX driver.

  When using the on-chip networking stack, the application should call the `close()` API from the BSD socket interface, which internally calls the Wi-Fi Framework module `close()` API.
- ProvisioningSet

  When the device is provisioned in client mode, the callback will be called when the connection with the AP is lost and re-established. When the device is provisioned in AP mode, this callback is called when any client connects/disconnects with the AP. When the device is provisioned in client mode, the arguments passed to the callback will only have the below valid field:

  — event = SF_WIFI_EVENT_AP_CONNECT or SF_WIFI_AP_DISCONNECT

  When the device is provisioned in AP mode, the arguments passed to the callback will only have the below valid fields:

  — event = SF_WIFI_EVENT_CLIENT_CONNECT or SF_WIFI_CLIENT_DISCONNECT

  — mac_addr = MAC address of client

  While calling the `provisioningSet()` API to provision the device in client mode, the framework will not call the callback function on successful association with the AP or on a failure.

  When the device is provisioned in AP mode, if the client tries to connect with the AP using the wrong password, the callback will be called twice, first with the connected event, and then immediately after this with the disconnected event.

BSD Socket APIs

While using on-chip networking stack, the application can use all BSD Socket APIs, all on-chip Networking Stack support APIs and few Synergy Wi-Fi Framework APIs. The Wi-Fi Framework APIs which the application can use are `provisioningSet()`, `provisioningGet()`, `scan()`, `macAddressGet()`, `macAddressSet()`, and `infoGet()`.

3.2.3 Wi-Fi Framework Module Limitations

- The Wi-Fi Framework does not support the Synergy S1 MCU Series due to memory constraints
- Due to memory constraints, S3A6 and S128 MCUs will support only on-chip networking stack (that is, Path 2 for Wi-Fi use, where networking stack runs on Wi-Fi chipset)
- When RSPI is used for the Wi-Fi module driver, the DTC components that are auto-filled for the dependencies, must be removed because of the limitation in the Wi-Fi module SPI driver. The limitation is that when DTC is used with RSPI, 32-bit transfers are required but GT202 vendor code (that is, Wi-Fi module SPI driver) uses 8-bit transfers only.
- Synergy Wi-Fi Framework APIs implemented for GT202 are not re-entrant. All these APIs call the driver APIs to do the requested operation. If GT202 driver is working on behalf of any Wi-Fi Framework API and another Wi-Fi Framework API is called, it will return SPP_ERR_IN_USE error until the ongoing operation is finished.
- Synergy Wi-Fi Framework `provisioningSet()` API for GT202 may fail if peripheral and IO pins' drive capacity is not set to medium.
- While configuring GT202 with SPI driver on r_rspi, set drive capacity of slave select pin, reset pin, and SPI pins (that is, MISO, MOSI and RSPCK) to medium. While configuring GT202 with SPI driver on r_sci_spi, set drive capacity of slave select pin, reset pin, and SCI pins (that is, TXD_MOSI and RXD_MISO) to medium. Do not change the drive capacity of the SCK pin when r_sci_spi driver is used.

4. Including the Wi-Fi Framework Module in an Application

This section describes how to include the Wi-Fi Framework module in an application using the SSP configurator.

Note: This section assumes you are familiar with creating a project, adding threads, adding a stack to a thread, and configuring a block within the stack. If you are unfamiliar with any of these items, refer to the first few chapters of the *SSP User's Manual* to learn how to perform these important steps when creating SSP-based applications.

To add the Wi-Fi Framework module to an application, simply add it to a thread using the stacks selection sequence given in the following table. There are four different options, so four sequences are shown.

Table 5 Wi-Fi Framework Selection Sequence

Resource	ISDE Tab	Stacks Selection Sequence
g_sf_wifi0 GT202 Wi-Fi Device Driver on sf_wifi_gt202	Threads	New Stack> Framework> Networking> Wi-Fi> GT202 Wi-Fi Device Driver on sf_wifi_gt202
g_sf_wifi_onchip_stack0 On-Chip Stack on GT202 Wi-Fi Framework	Threads	New Stack> Framework> Networking> Wi-Fi> On-Chip Stack on GT202 Wi-Fi Framework
g_sf_socket0 BSD Socket using GT202 On-Chip Stack on GT202 Wi-Fi Framework	Threads	New Stack> Framework> Networking> Wi-Fi> BSD Socket using GT202 On-Chip Stack on GT202 Wi-Fi Framework
g_sf_el_nx0 NetX Port using Wi-Fi Framework on sf_wifi_nsal_nx	Threads	New Stack> Framework> Networking> Wi-Fi> NetX Port using Wi-Fi Framework on sf_wifi_nsal_nx

When the Wi-Fi Framework module is added to the thread stack as shown in the following figure, the configurator automatically adds the needed lower-level drivers. Any drivers that need additional configuration information will be highlighted in Red. Modules with a Gray band are individual modules that stand alone. Modules with a Pink band require the selection of an implementation option.

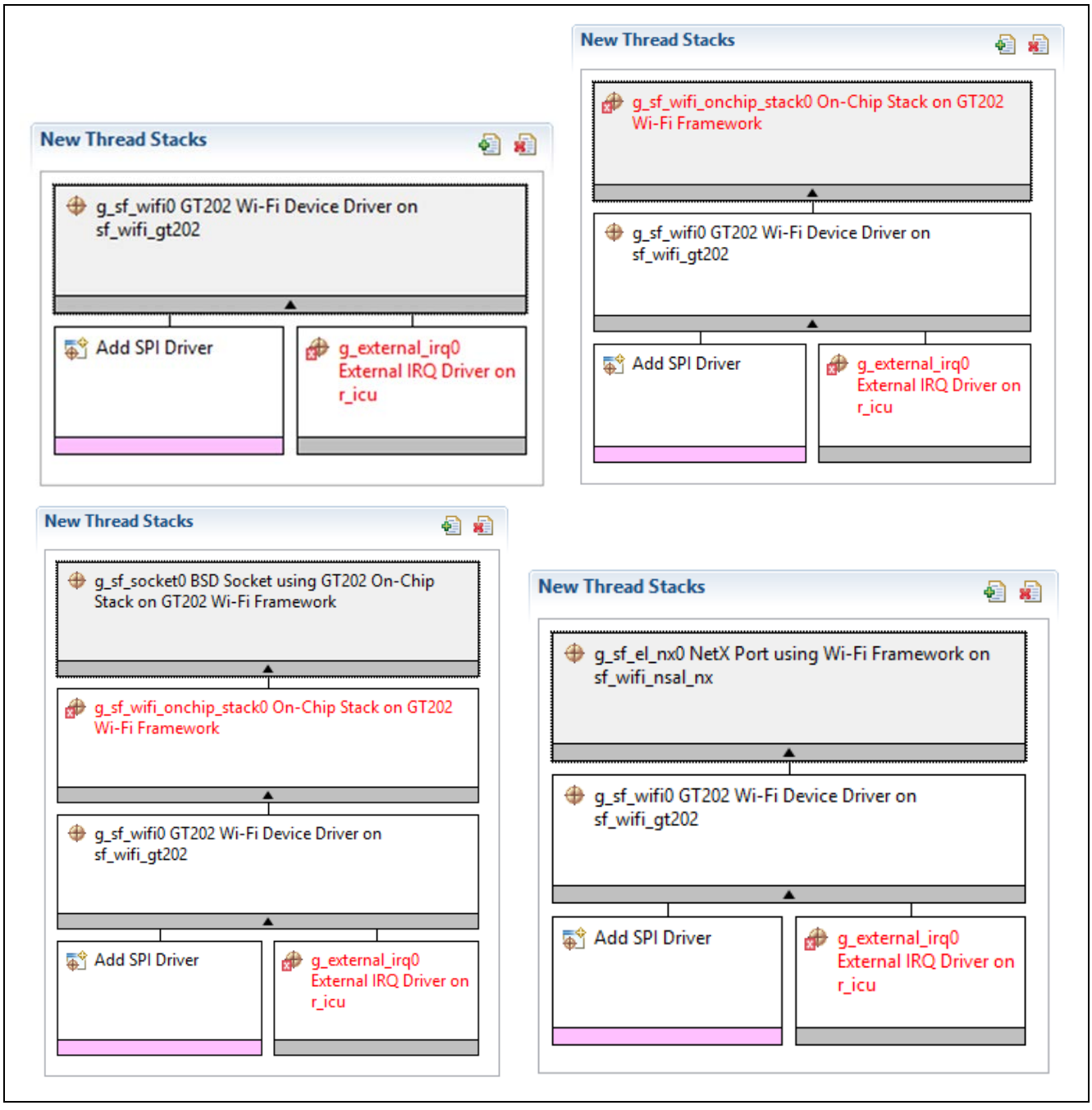

Figure 6 The Four Wi-Fi Framework Module Stack Options

Selecting SF_WIFI_NSAL_NX

The sf_wifi_nsal_nx implementation can be used in conjunction with a NetX IP instance. In this case, it is added as an option underneath the IP instance stack. Add the NetX IP instance using the selection steps shown in the following table.

Table 6 Selection Sequence for NetX IP instance

Resource	ISDE Tab	Stacks Selection Sequence
g_sf_wifi0 GT202 Wi-Fi Device Driver on sf_wifi_gt202	Threads	New Stack> X-Ware> NetX> NetX IP Instance

Use the available option to add sf_wifi_nsal_nx under the NetX IP Instance as a NetX Network Driver.

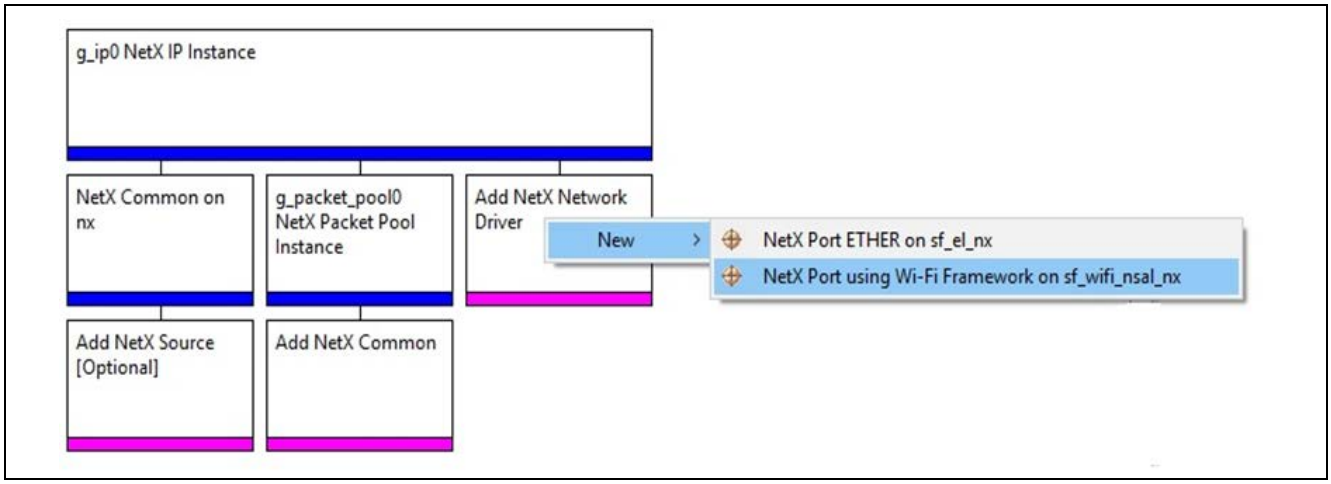

Figure 7 NetX IP Instance use of sf_wifi_nsal_nx

The lower level modules are filled in automatically as shown in the following figure.

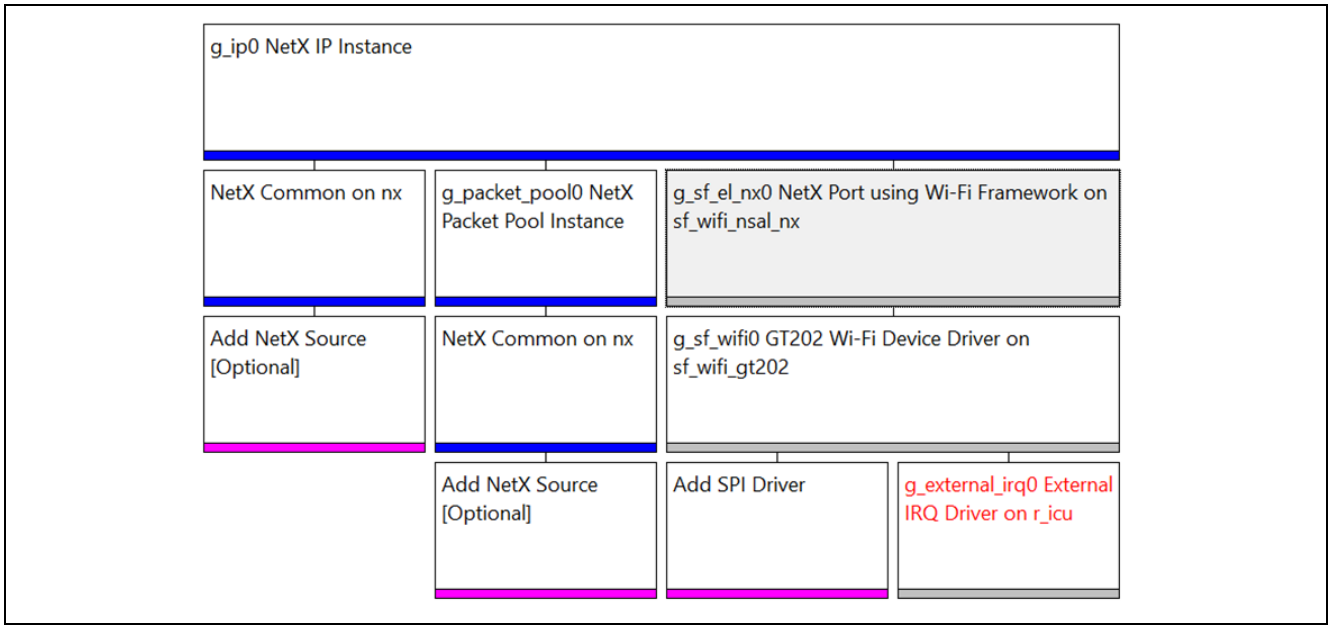

Figure 8 NetX Port using Wi-Fi Framework on sf_wifi_nsal_nx

5. Configuring the Wi-Fi Framework Modules

You must configure the Wi-Fi Framework module for the desired operation. The SSP configuration window will automatically identify, by highlighting the block in red, any required configuration selections, such as interrupts or operating modes, which must be configured for lower level modules, for successful operation. Furthermore, only those properties that can be changed without causing conflicts are available for modification. Other properties are 'locked' and not available for changes and are identified with a lock icon for the 'locked' property in the Property window in the ISDE. This approach simplifies the configuration process and makes it much less error prone than previous 'manual' approaches to configuration. The available configuration settings and defaults for all the user accessible properties are given in the properties tab within the SSP Configurator, and are shown in the following tables for easy reference.

One of the properties most often identified as requiring a change is the Interrupt Priority. This configuration setting is available with the Properties window of the associated module. Simply select the indicated module and then view the properties window. The Interrupt settings are often toward the bottom of the properties list, so scroll down until they become available. Also note that the Interrupt Priorities listed in the properties window in the ISDE will include an indication as to the validity of the setting based on the MCU targeted (CM4 or CM0+). This level of detail is not included in the below configuration properties tables, but is easily visible with the ISDE when configuring Interrupt Priority levels.

Note: You may want to open your ISDE and create the Wi-Fi Driver and explore the property settings in parallel with looking over the Configuration Table Settings given below. This will help orient you and can be a useful 'hands-on' approach to learning the ins and outs of developing with SSP.

Configuration properties for each of the four Wi-Fi implementations supported by the Wi-Fi Framework is provided in the following four sections.

5.1 Configuring the GT202 Wi-Fi Device Driver

This section shows the configuration settings for the Wi-Fi Device Driver on the GT202 implementation of the Wi-Fi Framework.

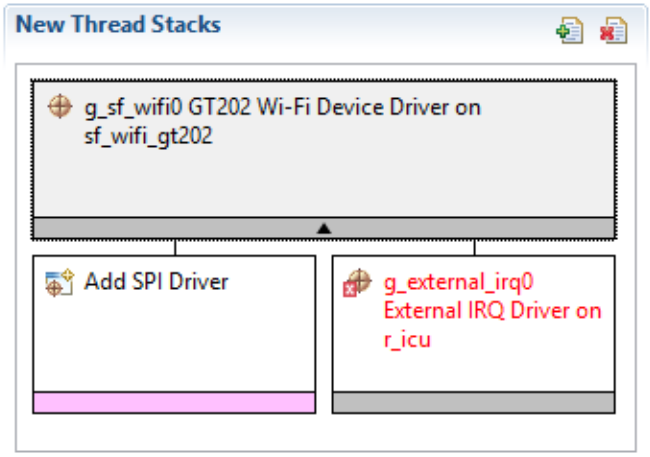

Figure 9 GT202 Wi-Fi Device Driver Thread Stack

Table 7 Configuration Settings for the Wi-Fi Device Driver on sf_wifi_gt202

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enable or disable the parameter checking
On-Chip Stack Support	Enabled, Disabled Default: Disabled	On-chip stack support selection
Driver Heap Size in bytes (Minimum 8192 bytes)	8192	Driver heap size selection
Name (Must be a valid C symbol)	g_sf_wifi0	Module name
Hardware Mode	802.11a, 802.11b, 802.11g, 802.11n Default: 802.11n	Hardware mode selection
Transmit (TX) Power (Valid Range 1-17)	10	Transmit power selection
Ready/Clear To Send (RTS/CTS) Flag	Enabled, Disabled Default: Enabled	Ready/Clear to send selection
Delivery Traffic Indication Message (DTIM) Interval (Valid Range: 1-255)	3	Delivery traffic indication message interval selection
Broadcast SSID (AP mode only)	Enabled, Disabled Default: Enabled	Broadcast SSID selection
Beacon Interval in Microseconds (AP mode only and must be greater than 1023)	1024	Beacon interval in microseconds selection
Station Inactivity Time out in Seconds (AP mode only and must be greater than 0)	100	Station inactivity timeout selection

Requested High Throughput	Enabled, Disabled Default: Disabled	Requested high throughput selection
Reset Pin (Must be a valid C symbol)	IOPORT_PORT_06_PIN_00	Reset pin selection
Slave Select Pin (SSL) (Must be a valid C symbol)	IOPORT_PORT_01_PIN_03	Slave select pin selection
GT202 Driver Task Thread Priority (Modifying Task Thread Priority may cause Driver to malfunction)	5	GT202 driver task thread priority selection
Callback	NULL	Callback selection
Support NetX Packet Chaining	Enabled, Disabled Default: Enabled	Support NetX packet chaining selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 8 Configuration Settings for the SPI HAL Driver on r_rspi

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enable or disable the parameter error checking
Name	g_spi0	Module name
Channel	0	SCI or SPI Channel number to which the device has been connected
Operating Mode	Master	Configure as a Master or Slave device. Current version of SSP supports only SPI Master mode.
Clock Phase	Data sampling on even edge, data variation on odd edge	Data sampling on odd or even clock edge
Clock Polarity	High when idle	Clock level when idle
Mode Fault Error	Disable	Indicates mode fault error (master/slave conflict) flag
Bit Order	MSB First	Select transmit order MSB/LSB first
Bitrate	500000	Transmission or reception rate. Bits per second.
Callback	NULL	Optional Callback function pointer
SPI Mode	Clock Synchronous operation	Select SPI or clock syn mode operation
Slave Select Polarity(SSL0)	Active Low	Select SSL0 signal polarity
Slave Select Polarity(SSL1)	Active Low	Select SSL1 signal polarity
Slave Select Polarity(SSL2)	Active Low	Select SSL2 signal polarity
Slave Select Polarity(SSL3)	Active Low	Select SSL3 signal polarity
Select Loopback1	Normal	Select loopback1
Select Loopback2	Normal	Select loopback2
Enable MOSI Idle	Disable	Select MOSI idle fixed value and selection
MOSI Idle State	MOSI Low	Select MOSI idle fixed value and selection
Enable Parity	Disable	Enable/disable parity
Parity Mode	Parity Even	Select parity
Select SSL (Slave Select)	SSL0	Select which slave to use; 0-SSL0; 1-SSL1; 2-SSL2; 3-SSL3
Select SSL Level After Transfer	SSL Level Do Not Keep	Select SSL level after transfer completion; 0-negate; 1-keep
Clock Delay Enable	Clock Delay Disable	Clock delay enable selection
Clock Delay Count	Clock Delay 1 RSPCK	Clock delay count selection

SSL Negation Delay Enable	Negation Delay Disable	SSL negation delay enable selection
Negation Delay Count	Negation Delay 1 RSPCK	Negation delay count selection
Next Access Delay Enable	Next Access Delay Disable	Next access delay enable selection
Next Access Delay Count	Next Access Delay 1 RSPCK	Next access delay count selection
Receive Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Receive interrupt priority selection
Transmit Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Transmit interrupt priority selection
Transmit End Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Transmit end interrupt priority selection
Error Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Error interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 9 Configuration Settings for the Transfer Driver on r_dtc Event SPI0 TXI

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Selects if code for parameter checking is to be included in the build
Software Start	Enabled, Disabled Default: Disabled	Software start selection
Linker section to keep DTC vector table	.ssp_dtc_vector_table	Linker section to keep DTC vector table selection
Name	g_transfer0	Module name
Mode	Normal	Mode selection
Transfer Size	2 Bytes	Transfer size selection

Destination Address Mode	Fixed	Destination address mode selection
Source Address Mode	Incremented	Source address mode selection
Repeat Area (Unused in Normal Mode)	Source	Repeat area selection
Interrupt Frequency	After all transfers have completed	Interrupt frequency selection
Destination Pointer	NULL	Destination pointer selection
Number of Transfers	0	Number of transfers selection
Number of Blocks (Valid only in Block Mode)	0	Number of blocks selection
Activation Source (Must enable IRQ)	Event SPI0 TXI	Activation source selection
Auto Enable	FALSE	Auto enable selection
Callback (Only valid with Software start)	NULL	Callback selection
ELC Software Event Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Disabled	ELC Software Event interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 10 Configuration Settings for the Transfer Driver on r_dtc Event SPI0 RXI

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Selects if code for parameter checking is to be included in the build
Software Start	Enabled, Disabled Default: Disabled	Software start selection
Linker section to keep DTC vector table	.ssp_dtc_vector_table	Linker section to keep DTC vector table selection
Name	g_transfer1	Module name
Mode	Normal	Mode selection
Transfer Size	2 Bytes	Transfer size selection
Destination Address Mode	Incremented	Destination address mode selection
Source Address Mode	Fixed	Source address mode selection
Repeat Area (Unused in Normal Mode)	Destination	Repeat area selection
Interrupt Frequency	After all transfers have completed	Interrupt frequency selection
Destination Pointer	NULL	Destination pointer selection
Source Pointer	NULL	Source pointer selection
Number of Transfers	0	Number of transfers selection
Number of Blocks (Valid only in Block Mode)	0	Number of blocks selection
Activation Source (Must enable IRQ)	Event SPI0 RXI	Activation source selection
Auto Enable	FALSE	Auto enable selection
Callback (Only valid with Software start)	NULL	Callback selection

ELC Software Event Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Disabled	ELC Software Event interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 11 Configuration Settings for the SPI HAL Module on r_sci_spi

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enable or disable the parameter error checking
Name	g_spi0	Module name
Channel	0	SCI or SPI Channel number to which the device has been connected
Operating Mode	Master	Configure as a Master or Slave device. Current version of SSP supports only SPI Master mode.
Clock Phase	Data sampling on even edge, data variation on odd edge	Data sampling on odd or even clock edge
Clock Polarity	High when idle	Clock level when idle
Mode Fault Error	Disable	Indicates mode fault error (master/slave conflict) flag
Bit Order	MSB First	Select transmit order MSB/LSB first
Bitrate	100000	Transmission or reception rate. Bits per second.
Bit Rate Modulation Enable	Enable, Disable Default: Enable	Bit Rate Modulation Function enable or disable. This is applicable only for SCI SPI.
Callback	NULL	Optional Call back function pointer
Receive Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Receive interrupt priority selection
Transmit Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Transmit interrupt priority selection

Transmit End Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Transmit end interrupt priority selection
Error Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Error interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 12 Configuration Settings for the Transfer Driver on r_dtc Event SCI0 TXI

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Selects if code for parameter checking is to be included in the build
Software Start	Enabled, Disabled Default: Disabled	Software start selection
Linker section to keep DTC vector table	.ssp_dtc_vector_table	Linker section to keep DTC vector table selection
Name	g_transfer0	Module name
Mode	Normal	Mode selection
Transfer Size	1 Byte	Transfer size selection
Destination Address Mode	Fixed	Destination address mode selection
Source Address Mode	Incremented	Source address mode selection
Repeat Area (Unused in Normal Mode	Source	Repeat area selection
Interrupt Frequency	After all transfers have completed	Interrupt frequency selection
Destination Pointer	NULL	Destination pointer selection
Source Pointer	NULL	Source pointer selection
Number of Transfers	0	Number of transfers selection
Number of Blocks (Valid only in Block Mode)	0	Number of blocks selection
Activation Source (Must enable IRQ)	Event SCI0 TXI	Activation source selection
Auto Enable	FALSE	Auto enable selection
Callback (Only valid with Software start)	NULL	Callback selection

ELC Software Event Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Disabled	ELC Software Event interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 13 Configuration Settings for the Transfer Driver on r_dtc Event SCI0 RXI

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Selects if code for parameter checking is to be included in the build
Software Start	Enabled, Disabled Default: Disabled	Software start selection
Linker section to keep DTC vector table	.ssp_dtc_vector_table	Linker section to keep DTC vector table selection
Name	g_transfer1	Module name
Mode	Normal	Mode selection
Transfer Size	1 Byte	Transfer size selection
Destination Address Mode	Incremented	Destination address mode selection
Source Address Mode	Fixed	Source address mode selection
Repeat Area (Unused in Normal Mode	Destination	Repeat area selection
Interrupt Frequency	After all transfers have completed	Interrupt frequency selection
Destination Pointer	NULL	Destination pointer selection
Source Pointer	NULL	Source pointer selection
Number of Transfers	0	Number of transfers selection
Number of Blocks (Valid only in Block Mode)	0	Number of blocks selection
Activation Source (Must enable IRQ)	Event SCI0 RXI	Activation source selection
Auto Enable	FALSE	Auto enable selection
Callback (Only valid with Software start)	NULL	Callback selection
ELC Software Event Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Disabled	ELC Software Event interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 14 Configuration Settings for the External IRQ HAL Module on r_icu

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Parameter checking setting enables or disables the addition of parameter checking code
Name	g_external_irq0	Module name
Channel	0	Specifies the hardware IRQ channel used
Trigger	Falling	Selection for trigger event mode
Digital Filtering	Disabled	Digital filter enable/disable
Digital Filtering Sample Clock (Only valid when Digital Filtering is Enabled)	PCLK/64	Sets noise filter sampling period
Interrupt enabled after initialization	TRUE	Determines if the interrupt is enabled immediately after initialization
Callback	custom_hw_irq_isr	A user callback function can be registered in external_irq_api_t::open. If this callback function is provided, it is called from the interrupt service routine (ISR) each time the IRQn triggers. **Warning:** Since the callback is called from an ISR, care should be taken not to use blocking calls or lengthy processing. Spending excessive time in an ISR can affect the responsiveness of the system.
Pin Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Disabled	Pin interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

5.2 Configuring the On-Chip Stack on GT202 Wi-Fi Framework

This section shows the configuration settings for the on-chip stack on the GT202 implementation of the Wi-Fi Framework.

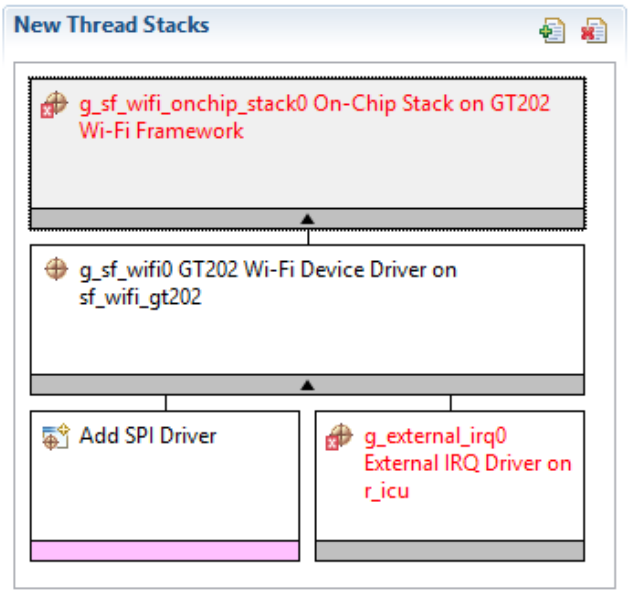

Figure 10 On-Chip Stack on GT202 Wi-Fi Framework Thread Stack

Table 15 Configuration Settings for the On-Chip Stack on GT202 Wi-Fi Framework

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enable or disable the parameter checking.
Name (Must be a valid C Symbol)	g_sf_wifi_onchip_stack0	Module name

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 16 Configuration Settings for the Wi-Fi Device Driver

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enable or disable the parameter checking
On-Chip Stack Support	Enabled, Disabled Default: Disabled	On-chip stack support selection
Driver Heap Size in bytes (Minimum 8192 bytes)	8192	Driver heap size selection
Name (Must be a valid C symbol)	g_sf_wifi0	Module name
Hardware Mode	802.11a, 802.11b, 802.11g, 802.11n Default: 802.11n	Hardware mode selection
Transmit (TX) Power (Valid Range 1-17)	10	Transmit power selection
Ready/Clear to Send (RTS/CTS) Flag	Enabled, Disabled Default: Enabled	Ready/Clear to send selection
Delivery Traffic Indication Message (DTIM) Interval (Valid Range: 1-255)	3	Delivery traffic indication message interval selection

Broadcast SSID (AP mode only)	Enabled, Disabled Default: Enabled	Broadcast SSID selection
Beacon Interval in Microseconds (AP mode only and must be greater than 1023)	1024	Beacon interval in microseconds selection
Station Inactivity Time out in Seconds (AP mode only and must be greater than 0)	100	Station inactivity timeout selection
Requested High Throughput	Enabled, Disabled Default: Disabled	Requested high throughput selection
Reset Pin (Must be a valid C symbol)	IOPORT_PORT_06_PIN_00	Reset pin selection
Slave Select Pin (SSL) (Must be a valid C symbol)	IOPORT_PORT_01_PIN_03	Slave select pin selection
GT202 Driver Task Thread Priority (Modifying Task Thread Priority may cause Driver to malfunction)	5	GT202 driver task thread priority selection
Callback	NULL	Callback selection
Support NetX Packet Chaining	Enabled, Disabled Default: Enabled	Support NetX packet chaining selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 17 Configuration Settings for the SPI HAL Driver on r_rspi

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enable or disable the parameter error checking
Name	g_spi0	Module name
Channel	0	SCI or SPI Channel number to which the device has been connected
Operating Mode	Master	Configure as a Master or Slave device. Current version of SSP supports only SPI Master mode.
Clock Phase	Data sampling on even edge, data variation on odd edge	Data sampling on odd or even clock edge
Clock Polarity	High when idle	Clock level when idle
Mode Fault Error	Disable	Indicates Mode fault error (master/slave conflict) flag
Bit Order	MSB First	Select transmit order MSB/LSB first
Bitrate	500000	Transmission or reception rate. Bits per second.
Callback	NULL	Optional Callback function pointer
SPI Mode	Clock Synchronous operation	Select SPI or clock syn mode operation
Slave Select Polarity (SSL0)	Active Low	Select SSL0 signal polarity
Slave Select Polarity (SSL1)	Active Low	Select SSL1 signal polarity
Slave Select Polarity (SSL2)	Active Low	Select SSL2 signal polarity

Slave Select Polarity (SSL3)	Active Low	Select SSL3 signal polarity
Select Loopback1	Normal	Select loopback1
Select Loopback2	Normal	Select loopback2
Enable MOSI Idle	Disable	Select MOSI idle fixed value and selection
MOSI Idle State	MOSI Low	Select MOSI idle fixed value and selection
Enable Parity	Disable	Enable/disable parity
Parity Mode	Parity Even	Select parity
Select SSL (Slave Select)	SSL0	Select which slave to use; 0-SSL0; 1-SSL1; 2-SSL2; 3-SSL3
Select SSL Level After Transfer	SSL Level Do Not Keep	Select SSL level after transfer completion; 0-negate; 1-keep
Clock Delay Enable	Clock Delay Disable	Clock delay enable selection
Clock Delay Count	Clock Delay 1 RSPCK	Clock delay count selection
SSL Negation Delay Enable	Negation Delay Disable	SSL negation delay enable selection
Negation Delay Count	Negation Delay 1 RSPCK	Negation delay count selection
Next Access Delay Enable	Next Access Delay Disable	Next access delay enable selection
Next Access Delay Count	Next Access Delay 1 RSPCK	Next access delay count selection
Receive Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Receive interrupt priority selection
Transmit Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Transmit interrupt priority selection
Transmit End Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Transmit end interrupt priority selection
Error Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Error interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 18 Configuration Settings for the Transfer Driver on r_dtc Event SPI0 TXI

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Selects if code for parameter checking is to be included in the build
Software Start	Enabled, Disabled Default: Disabled	Software start selection
Linker section to keep DTC vector table	.ssp_dtc_vector_table	Linker section to keep DTC vector table selection
Name	g_transfer0	Module name
Mode	Normal	Mode selection
Transfer Size	2 Bytes	Transfer size selection
Destination Address Mode	Fixed	Destination address mode selection
Source Address Mode	Incremented	Source address mode selection
Repeat Area (Unused in Normal Mode)	Source	Repeat area selection
Interrupt Frequency	After all transfers have completed	Interrupt frequency selection
Destination Pointer	NULL	Destination pointer selection
Source Pointer	NULL	Source pointer selection
Number of Transfers	0	Number of transfers selection
Number of Blocks (Valid only in Block Mode)	0	Number of blocks selection
Activation Source (Must enable IRQ)	Event SPI0 TXI	Activation source selection
Auto Enable	FALSE	Auto enable selection
Callback (Only valid with Software start)	NULL	Callback selection
ELC Software Event Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Disabled	ELC Software Event interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 19 Configuration Settings for the Transfer Driver on r_dtc Event SPI0 RXI

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Selects if code for parameter checking is to be included in the build
Software Start	Enabled, Disabled Default: Disabled	Software start selection
Linker section to keep DTC vector table	.ssp_dtc_vector_table	Linker section to keep DTC vector table selection
Name	g_transfer1	Module name
Mode	Normal	Mode selection
Transfer Size	2 Bytes	Transfer size selection
Destination Address Mode	Incremented	Destination address mode selection

Source Address Mode	Fixed	Source address mode selection
Repeat Area (Unused in Normal Mode)	Destination	Repeat area selection
Interrupt Frequency	After all transfers have completed	Interrupt frequency selection
Destination Pointer	NULL	Destination pointer selection
Source Pointer	NULL	Source pointer selection
Number of Transfers	0	Number of transfers selection
Number of Blocks (Valid only in Block Mode)	0	Number of blocks selection
Activation Source (Must enable IRQ)	Event SPI0 RXI	Activation source selection
Auto Enable	FALSE	Auto enable selection
Callback (Only valid with Software start)	NULL	Callback selection
ELC Software Event Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Disabled	ELC Software Event interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 20 Configuration Settings for the SPI HAL Module on r_sci_spi

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enable or disable the parameter error checking
Name	g_spi0	Module name
Channel	0	SCI or SPI Channel number to which the device has been connected
Operating Mode	Master	Configure as a Master or Slave device. Current version of SSP supports only SPI Master mode.
Clock Phase	Data sampling on even edge, data variation on odd edge	Data sampling on odd or even clock edge
Clock Polarity	High when idle	Clock level when idle
Mode Fault Error	Disable	Indicates mode fault error (master/slave conflict) flag
Bit Order	MSB First	Select transmit order MSB/LSB first
Bitrate	100000	Transmission or reception rate. Bits per second.
Bit Rate Modulation Enable	Enable, Disable Default: Enable	Bit Rate Modulation Function enable or disable. This is applicable only for SCI SPI.
Callback	NULL	Optional Call back function pointer

Receive Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Receive interrupt priority selection
Transmit Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Transmit interrupt priority selection
Transmit End Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Transmit end interrupt priority selection
Error Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Error interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 21 Configuration Settings for the Transfer Driver on r_dtc Event SCI0 TXI

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Selects if code for parameter checking is to be included in the build
Software Start	Enabled, Disabled Default: Disabled	Software start selection
Linker section to keep DTC vector table	.ssp_dtc_vector_table	Linker section to keep DTC vector table selection
Name	g_transfer0	Module name
Mode	Normal	Mode selection
Transfer Size	1 Byte	Transfer size selection
Destination Address Mode	Fixed	Destination address mode selection
Source Address Mode	Incremented	Source address mode selection
Repeat Area (Unused in Normal Mode	Source	Repeat area selection
Interrupt Frequency	After all transfers have completed	Interrupt frequency selection
Destination Pointer	NULL	Destination pointer selection
Source Pointer	NULL	Source pointer selection
Number of Transfers	0	Number of transfers selection

Number of Blocks (Valid only in Block Mode)	0	Number of blocks selection
Activation Source (Must enable IRQ)	Event SCI0 TXI	Activation source selection
Auto Enable	FALSE	Auto enable selection
Callback (Only valid with Software start)	NULL	Callback selection
ELC Software Event Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Disabled	ELC Software Event interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 22 Configuration Settings for the Transfer Driver on r_dtc Event SCI0 RXI

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Selects if code for parameter checking is to be included in the build
Software Start	Enabled, Disabled Default: Disabled	Software start selection
Linker section to keep DTC vector table	.ssp_dtc_vector_table	Linker section to keep DTC vector table selection
Name	g_transfer1	Module name
Mode	Normal	Mode selection
Transfer Size	1 Byte	Transfer size selection
Destination Address Mode	Incremented	Destination address mode selection
Source Address Mode	Fixed	Source address mode selection
Repeat Area (Unused in Normal Mode	Destination	Repeat area selection
Interrupt Frequency	After all transfers have completed	Interrupt frequency selection
Destination Pointer	NULL	Destination pointer selection
Source Pointer	NULL	Source pointer selection
Number of Transfers	0	Number of transfers selection
Number of Blocks (Valid only in Block Mode)	0	Number of blocks selection
Activation Source (Must enable IRQ)	Event SCI0 RXI	Activation source selection
Auto Enable	FALSE	Auto enable selection
Callback (Only valid with Software start)	NULL	Callback selection
ELC Software Event Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Disabled	ELC Software Event interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 23 Configuration Settings for the External IRQ HAL Module on r_icu

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Parameter checking setting enables or disables the addition of parameter checking code
Name	g_external_irq0	Module name
Channel	0	Specifies the hardware IRQ channel used
Trigger	Falling	Selection for trigger event mode
Digital Filtering	Disabled	Digital filter enable/disable
Digital Filtering Sample Clock (Only valid when Digital Filtering is Enabled)	PCLK/64	Sets noise filter sampling period
Interrupt enabled after initialization	TRUE	Determines if the interrupt is enabled immediately after initialization
Callback	custom_hw_irq_isr	A user callback function can be registered in external_irq_api_t::open. If this callback function is provided, it is called from the interrupt service routine (ISR) each time the IRQn triggers. Warning: Since the callback is called from an ISR, care should be taken not to use blocking calls or lengthy processing. Spending excessive time in an ISR can affect the responsiveness of the system.
Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Disabled	Interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

5.3 Configuring the BSD Socket using GT202 On-Chip Stack on GT202 Wi-Fi Framework

This section shows the configuration settings for the BSD socket using the GT202 on-chip stack implementation of the Wi-Fi Framework.

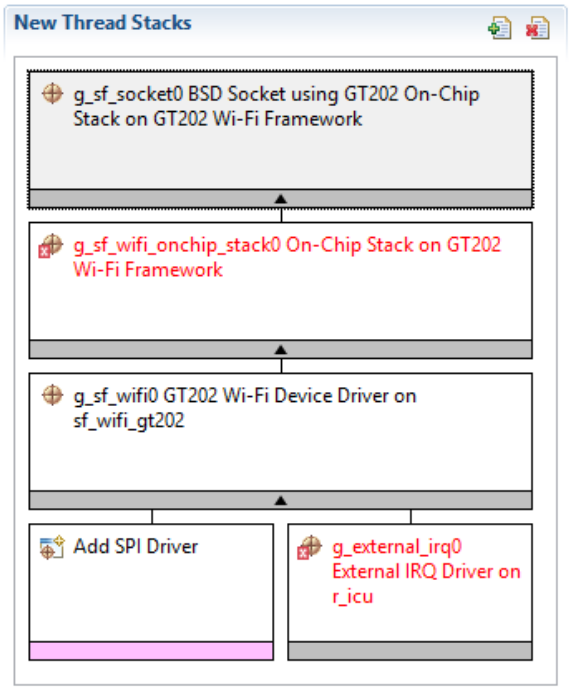

Figure 11 BSD Socket using GT202 On-Chip Stack on GT202 Wi-Fi Framework

Table 24 Configuration Settings for the BSD Socket using GT202 On-Chip Stack on GT202 Wi-Fi Framework

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enable or disable the parameter checking
Name (Must be a valid C Symbol)	g_sf_socket0	Module name

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 25 Configuration Settings for the On-Chip Stack on GT202 Wi-Fi Framework

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enable or disable the parameter checking
Name (Must be a valid C Symbol)	g_sf_wifi_onchip_stack0	Module name

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 26 Configuration Settings for the Wi-Fi Device Driver

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enable or disable the parameter checking
On-Chip Stack Support	Enabled, Disabled Default: Disabled	On-chip stack support selection

Driver Heap Size in bytes (Minimum 8192 bytes)	8192	Driver heap size selection
Name (Must be a valid C symbol)	g_sf_wifi0	Module name
Hardware Mode	802.11a, 802.11b, 802.11g, 802.11n Default: 802.11n	Hardware mode selection
Transmit (TX) Power (Valid Range 1-17)	10	Transmit power selection
Ready/Clear to Send (RTS/CTS) Flag	Enabled, Disabled Default: Enabled	Ready/Clear to send selection
Delivery Traffic Indication Message (DTIM) Interval (Valid range: 1-255)	3	Delivery traffic indication message interval selection
Broadcast SSID (AP mode only)	Enabled, Disabled Default: Enabled	Broadcast SSID selection
Beacon Interval in Microseconds (AP mode only and must be greater than 1023)	1024	Beacon interval in microseconds selection
Station Inactivity Time out in Seconds (AP mode only and must be greater than 0)	100	Station inactivity timeout selection
Requested High Throughput	Enabled, Disabled Default: Disabled	Requested high throughput selection
Reset Pin (Must be a valid C symbol)	IOPORT_PORT_06_PIN_00	Reset pin selection
Slave Select Pin (SSL) (Must be a valid C symbol)	IOPORT_PORT_01_PIN_03	Slave select pin selection
GT202 Driver Task Thread Priority (Modifying Task Thread Priority may cause Driver to malfunction)	5	GT202 driver task thread priority selection
Callback	NULL	Callback selection
Support NetX Packet Chaining	Enabled, Disabled Default: Enabled	Support NetX packet chaining selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 27 Configuration Settings for the SPI HAL Driver on r_rspi

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enable or disable the parameter error checking
Name	g_spi0	Module name
Channel	0	SCI or SPI Channel number to which the device has been connected
Operating Mode	Master	Configure as a Master or Slave device. Current version of SSP supports only SPI Master mode.
Clock Phase	Data sampling on even edge, data variation on odd edge	Data sampling on odd or even clock edge
Clock Polarity	High when idle	Clock level when idle

Mode Fault Error	Disable	Indicates Mode fault error (master/slave conflict) flag
Bit Order	MSB First	Select transmit order MSB/LSB first
Bitrate	500000	Transmission or reception rate. Bits per second.
Callback	NULL	Optional Callback function pointer
SPI Mode	Clock Synchronous operation	Select SPI or clock syn mode operation
Slave Select Polarity (SSL0)	Active Low	Select SSL0 signal polarity
Slave Select Polarity (SSL1)	Active Low	Select SSL1 signal polarity
Slave Select Polarity (SSL2)	Active Low	Select SSL2 signal polarity
Slave Select Polarity (SSL3)	Active Low	Select SSL3 signal polarity
Select Loopback1	Normal	Select loopback1
Select Loopback2	Normal	Select loopback2
Enable MOSI Idle	Disable	Select MOSI idle fixed value and selection
MOSI Idle State	MOSI Low	Select MOSI idle fixed value and selection
Enable Parity	Disable	Enable/disable parity
Parity Mode	Parity Even	Select parity
Select SSL (Slave Select)	SSL0	Select which slave to use; 0-SSL0; 1-SSL1; 2-SSL2; 3-SSL3
Select SSL Level After Transfer	SSL Level Do Not Keep	Select SSL level after transfer completion; 0-negate; 1-keep
Clock Delay Enable	Clock Delay Disable	Clock delay enable selection
Clock Delay Count	Clock Delay 1 RSPCK	Clock delay count selection
SSL Negation Delay Enable	Negation Delay Disable	SSL negation delay enable selection
Negation Delay Count	Negation Delay 1 RSPCK	Negation delay count selection
Next Access Delay Enable	Next Access Delay Disable	Next access delay enable selection
Next Access Delay Count	Next Access Delay 1 RSPCK	Next access delay count selection
Receive Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Receive interrupt priority selection
Transmit Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Transmit interrupt priority selection
Transmit End Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid),	Transmit end interrupt priority selection

	Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	
Error Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Error interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 28 Configuration Settings for the Transfer Driver on r_dtc Event SPI0 TXI

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Selects if code for parameter checking is to be included in the build
Software Start	Enabled, Disabled Default: Disabled	Software start selection
Linker section to keep DTC vector table	.ssp_dtc_vector_table	Linker section to keep DTC vector table selection
Name	g_transfer0	Module name
Mode	Normal	Mode selection
Transfer Size	2 Bytes	Transfer size selection
Destination Address Mode	Fixed	Destination address mode selection
Source Address Mode	Incremented	Source address mode selection
Repeat Area (Unused in Normal Mode)	Source	Repeat area selection
Interrupt Frequency	After all transfers have completed	Interrupt frequency selection
Destination Pointer	NULL	Destination pointer selection
Source Pointer	NULL	Source pointer selection
Number of Transfers	0	Number of transfers selection
Number of Blocks (Valid only in Block Mode)	0	Number of blocks selection
Activation Source (Must enable IRQ)	Event SPI0 TXI	Activation source selection
Auto Enable	FALSE	Auto enable selection
Callback (Only valid with Software start)	NULL	Callback selection
ELC Software Event Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Disabled	ELC Software Event interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 29 Configuration Settings for the Transfer Driver on r_dtc Event SPI0 RXI

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Selects if code for parameter checking is to be included in the build
Software Start	Enabled, Disabled Default: Disabled	Software start selection
Linker section to keep DTC vector table	.ssp_dtc_vector_table	Linker section to keep DTC vector table selection
Name	g_transfer1	Module name
Mode	Normal	Mode selection
Transfer Size	2 Bytes	Transfer size selection
Destination Address Mode	Incremented	Destination address mode selection
Source Address Mode	Fixed	Source address mode selection
Repeat Area (Unused in Normal Mode	Destination	Repeat area selection
Interrupt Frequency	After all transfers have completed	Interrupt frequency selection
Destination Pointer	NULL	Destination pointer selection
Source Pointer	NULL	Source pointer selection
Number of Transfers	0	Number of transfers selection
Number of Blocks (Valid only in Block Mode)	0	Number of blocks selection
Activation Source (Must enable IRQ)	Event SPI0 RXI	Activation source selection
Auto Enable	FALSE	Auto enable selection
Callback (Only valid with Software start)	NULL	Callback selection
ELC Software Event Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Disabled	ELC Software Event interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 30 Configuration Settings for the SPI HAL Module on r_sci_spi

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enable or disable the parameter error checking
Name	g_spi0	Module name
Channel	0	SCI or SPI Channel number to which the device has been connected
Operating Mode	Master	Configure as a Master or Slave device. Current version of SSP supports only SPI Master mode.
Clock Phase	Data sampling on even edge, data variation on odd edge	Data sampling on odd or even clock edge
Clock Polarity	High when idle	Clock level when idle

Mode Fault Error	Disable	Indicates Mode fault error (master/slave conflict) flag
Bit Order	MSB First	Select transmit order MSB/LSB first
Bitrate	100000	Transmission or reception rate. Bits per second.
Bit Rate Modulation Enable	Enable, Disable Default: Enable	Bitrate Modulation Function enable or disable. This is applicable only for SCI SPI.
Callback	NULL	Optional Call back function pointer
Receive Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Receive interrupt priority selection
Transmit Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Transmit interrupt priority selection
Transmit End Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Transmit end interrupt priority selection
Error Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Error interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 31 Configuration Settings for the Transfer Driver on r_dtc Event SCI0 TXI

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Selects if code for parameter checking is to be included in the build
Software Start	Enabled, Disabled Default: Disabled	Software start selection
Linker section to keep DTC vector table	.ssp_dtc_vector_table	Linker section to keep DTC vector table selection

Name	g_transfer0	Module name
Mode	Normal	Mode selection
Transfer Size	1 Byte	Transfer size selection
Destination Address Mode	Fixed	Destination address mode selection
Source Address Mode	Incremented	Source address mode selection
Repeat Area (Unused in Normal Mode	Source	Repeat area selection
Interrupt Frequency	After all transfers have completed	Interrupt frequency selection
Destination Pointer	NULL	Destination pointer selection
Source Pointer	NULL	Source pointer selection
Number of Transfers	0	Number of transfers selection
Number of Blocks (Valid only in Block Mode)	0	Number of blocks selection
Activation Source (Must enable IRQ)	Event SCI0 TXI	Activation source selection
Auto Enable	FALSE	Auto enable selection
Callback (Only valid with Software start)	NULL	Callback selection
ELC Software Event Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Disabled	ELC Software Event interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 32 Configuration Settings for the Transfer Driver on r_dtc Event SCI0 RXI

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Selects if code for parameter checking is to be included in the build
Software Start	Enabled, Disabled Default: Disabled	Software start selection
Linker section to keep DTC vector table	.ssp_dtc_vector_table	Linker section to keep DTC vector table selection
Name	g_transfer1	Module name
Mode	Normal	Mode selection
Transfer Size	1 Byte	Transfer size selection
Destination Address Mode	Incremented	Destination address mode selection
Source Address Mode	Fixed	Source address mode selection
Repeat Area (Unused in Normal Mode	Destination	Repeat area selection
Interrupt Frequency	After all transfers have completed	Interrupt frequency selection
Destination Pointer	NULL	Destination pointer selection
Source Pointer	NULL	Source pointer selection
Number of Transfers	0	Number of transfers selection
Number of Blocks (Valid only in Block Mode)	0	Number of blocks selection

Activation Source (Must enable IRQ)	Event SCI0 RXI	Activation source selection
Auto Enable	FALSE	Auto enable selection
Callback (Only valid with Software start)	NULL	Callback selection
ELC Software Event Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Disabled	ELC Software Event interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 33 Configuration Settings for the External IRQ HAL Module on r_icu

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Parameter checking setting enables or disables the addition of parameter checking code
Name	g_external_irq0	Module name
Channel	0	Specifies the hardware IRQ channel used
Trigger	Falling	Selection for trigger event mode
Digital Filtering	Disabled	Digital filter enable/disable
Digital Filtering Sample Clock (Only valid when Digital Filtering is Enabled)	PCLK/64	Sets noise filter sampling period
Interrupt enabled after initialization	TRUE	Determines if the interrupt is enabled immediately after initialization
Callback	custom_hw_irq_isr	A user callback function can be registered in external_irq_api_t::open. If this callback function is provided, it is called from the interrupt service routine (ISR) each time the IRQn triggers. **Warning:** Since the callback is called from an ISR, care should be taken not to use blocking calls or lengthy processing. Spending excessive time in an ISR can affect the responsiveness of the system.
Pin Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Disabled	Pin interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

5.4 Configuring the NetX Port using Wi-Fi Framework

This section shows the configuration settings for the NetX Port implementation of the Wi-Fi Framework.

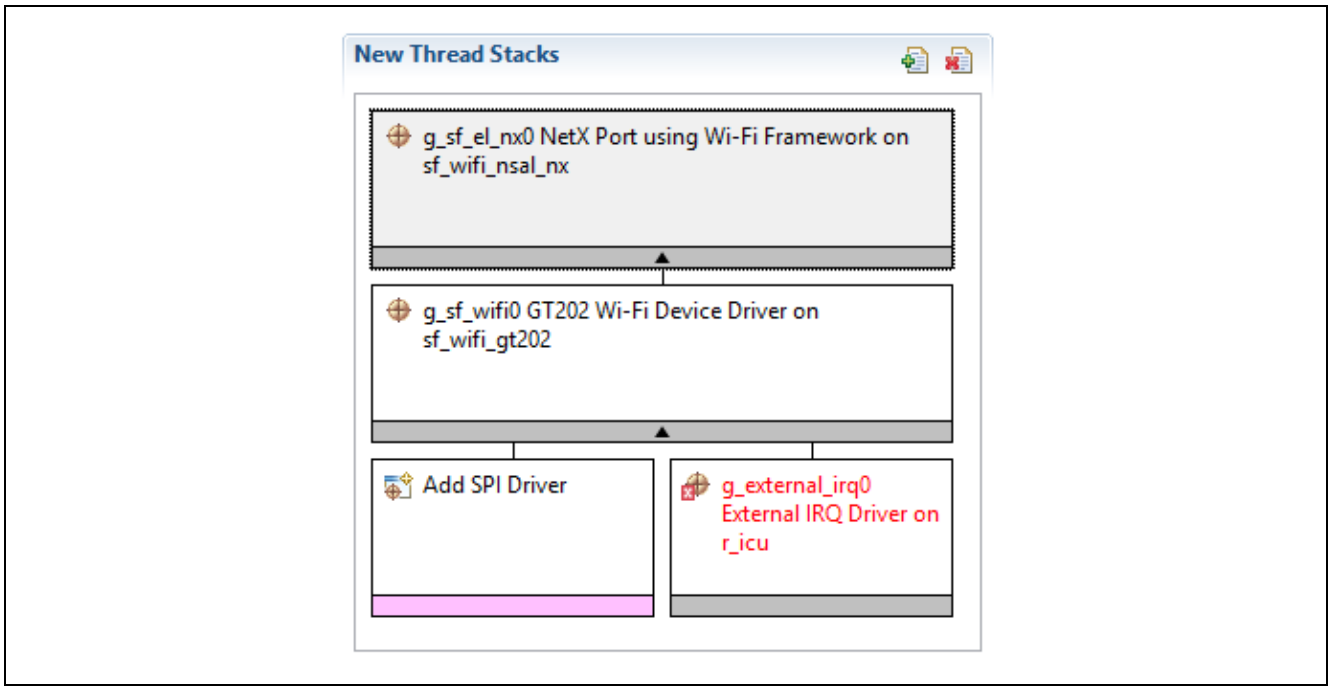

Figure 12 NetX Port using Wi-Fi Framework

Table 34 Configuration Settings for the NetX Port using Wi-Fi Framework on sf_wifi_nsal_nx

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enable or disable the parameter checking
Name (Must be a valid C Symbol)	g_sf_el_nx0	Module name

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 35 Configuration Settings for the Wi-Fi Device Driver

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enable or disable the parameter checking
On-Chip Stack Support	Enabled, Disabled Default: Disabled	On-chip stack support selection
Driver Heap Size in bytes (Minimum 8192 bytes)	8192	Driver heap size selection
Name (Must be a valid C symbol)	g_sf_wifi0	Module name
Hardware Mode	802.11a, 802.11b, 802.11g, 802.11n Default: 802.11n	Hardware mode selection
Transmit (TX) Power (Valid Range 1-17)	10	Transmit power selection
Ready/Clear to Send (RTS/CTS) Flag	Enabled, Disabled Default: Enabled	Ready/Clear to send selection
Delivery Traffic Indication Message (DTIM) Interval (Valid Range: 1-255)	3	Delivery traffic indication message interval selection

Broadcast SSID (AP mode only)	Enabled, Disabled Default: Enabled	Broadcast SSID selection
Beacon Interval in Microseconds (AP mode only and must be greater than 1023)	1024	Beacon interval in microseconds selection
Station Inactivity Time out in Seconds (AP mode only and must be greater than 0)	100	Station inactivity timeout selection
Requested High Throughput	Enabled, Disabled Default: Disabled	Requested high throughput selection
Reset Pin (Must be a valid C symbol)	IOPORT_PORT_06_PIN_00	Reset pin selection
Slave Select Pin (SSL) (Must be a valid C symbol)	IOPORT_PORT_01_PIN_03	Slave select pin selection
GT202 Driver Task Thread Priority (Modifying Task Thread Priority may cause Driver to malfunction)	5	GT202 driver task thread priority selection
Callback	NULL	Callback selection
Support NetX Packet Chaining	Enabled, Disabled Default: Enabled	Support NetX packet chaining selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 36 Configuration Settings for the SPI HAL Driver on r_rspi

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enable or disable the parameter error checking
Name	g_spi0	Module name
Channel	0	SCI or SPI Channel number to which the device has been connected
Operating Mode	Master	Configure as a Master or Slave device. Current version of SSP supports only SPI Master mode.
Clock Phase	Data sampling on even edge, data variation on odd edge	Data sampling on odd or even clock edge
Clock Polarity	High when idle	Clock level when idle
Mode Fault Error	Disable	Indicates Mode fault error (master/slave conflict) flag
Bit Order	MSB First	Select transmit order MSB/LSB first
Bitrate	500000	Transmission or reception rate. Bits per second.
Callback	NULL	Optional Callback function pointer
SPI Mode	Clock Synchronous operation	Select spi or clock syn mode operation
Slave Select Polarity(SSL0)	Active Low	Select SSL0 signal polarity
Slave Select Polarity(SSL1)	Active Low	Select SSL1 signal polarity
Slave Select Polarity(SSL2)	Active Low	Select SSL2 signal polarity
Slave Select Polarity(SSL3)	Active Low	Select SSL3 signal polarity
Select Loopback1	Normal	Select loopback1
Select Loopback2	Normal	Select loopback2
Enable MOSI Idle State	Disable	Select MOSI idle fixed value and selection
MOSI Idle State	MOSI Low	Select mosi idle fixed value and selection
Enable Parity	Disable	Enable/disable parity

Parity Mode	Parity Even	Select parity
Select SSL (Slave Select)	SSL0	Select which slave to use; 0-SSL0; 1-SSL1; 2-SSL2; 3-SSL3
Select SSL Level After Transfer	SSL Level Do Not Keep	Select SSL level after transfer completion; 0-negate; 1-keep
Clock Delay Enable	Clock Delay Disable	Clock delay enable selection
Clock Delay Count	Clock Delay 1 RSPCK	Clock delay count selection
SSL Negation Delay Enable	Negation Delay Disable	SSL negation delay enable selection
Negation Delay Count	Negation Delay 1 RSPCK	Negation delay count selection
Next Access Delay Enable	Next Access Delay Disable	Next access delay enable selection
Next Access Delay Count	Next Access Delay 1 RSPCK	Next access delay count selection
Receive Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Receive interrupt priority selection
Transmit Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Transmit interrupt priority selection
Transmit End Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Transmit end interrupt priority selection
Error Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Error interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 37 Configuration Settings for the Transfer Driver on r_dtc Event SPI0 TXI

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Selects if code for parameter checking is to be included in the build
Software Start	Enabled, Disabled	Software start selection

	Default: Disabled	
Linker section to keep DTC vector table	.ssp_dtc_vector_table	Linker section to keep DTC vector table selection
Name	g_transfer0	Module name
Mode	Normal	Mode selection
Transfer Size	2 Bytes	Transfer size selection
Destination Address Mode	Fixed	Destination address mode selection
Source Address Mode	Incremented	Source address mode selection
Repeat Area (Unused in Normal Mode)	Source	Repeat area selection
Interrupt Frequency	After all transfers have completed	Interrupt frequency selection
Destination Pointer	NULL	Destination pointer selection
Source Pointer	NULL	Source pointer selection
Number of Transfers	0	Number of transfers selection
Number of Blocks (Valid only in Block Mode)	0	Number of blocks selection
Activation Source (Must enable IRQ)	Event SPI0 TXI	Activation source selection
Auto Enable	FALSE	Auto enable selection
Callback (Only valid with Software start)	NULL	Callback selection
ELC Software Event Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Disabled	ELC Software Event interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 38 Configuration Settings for the Transfer Driver on r_dtc Event SPI0 RXI

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Selects if code for parameter checking is to be included in the build
Software Start	Enabled, Disabled Default: Disabled	Software start selection
Linker section to keep DTC vector table	.ssp_dtc_vector_table	Linker section to keep DTC vector table selection
Name	g_transfer1	Module name
Mode	Normal	Mode selection
Transfer Size	2 Bytes	Transfer size selection
Destination Address Mode	Incremented	Destination address mode selection
Source Address Mode	Fixed	Source address mode selection
Repeat Area (Unused in Normal Mode)	Destination	Repeat area selection
Interrupt Frequency	After all transfers have completed	Interrupt frequency selection
Destination Pointer	NULL	Destination pointer selection
Source Pointer	NULL	Source pointer selection
Number of Transfers	0	Number of transfers selection

Number of Blocks (Valid only in Block Mode)	0	Number of blocks selection
Activation Source (Must enable IRQ)	Event SPI0 RXI	Activation source selection
Auto Enable	FALSE	Auto enable selection
Callback (Only valid with Software start)	NULL	Callback selection
ELC Software Event Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Disabled	ELC Software Event interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 39 Configuration Settings for the SPI HAL Module on r_sci_spi

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enable or disable the parameter error checking
Name	g_spi0	Module name
Channel	0	SCI or SPI Channel number to which the device has been connected
Operating Mode	Master	Configure as a Master or Slave device. Current version of SSP supports only SPI Master mode.
Clock Phase	Data sampling on even edge, data variation on odd edge	Data sampling on odd or even clock edge.
Clock Polarity	High when idle	Clock level when idle.
Mode Fault Error	Disable	Indicates Mode fault error (master/slave conflict) flag
Bit Order	MSB First	Select transmit order MSB/LSB first
Bitrate	100000	Transmission or reception rate. Bits per second
Bit Rate Modulation Enable	Enable, Disable Default: Enable	Bitrate Modulation Function enable or disable. This is applicable only for SCI SPI.
Callback	NULL	Optional Call back function pointer
Receive Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Receive interrupt priority selection

Transmit Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Transmit interrupt priority selection
Transmit End Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Transmit end interrupt priority selection
Error Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Priority 2	Error interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 40 Configuration Settings for the Transfer Driver on r_dtc Event SCI0 TXI

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Selects if code for parameter checking is to be included in the build
Software Start	Enabled, Disabled Default: Disabled	Software start selection
Linker section to keep DTC vector table	.ssp_dtc_vector_table	Linker section to keep DTC vector table selection
Name	g_transfer0	Module name
Mode	Normal	Mode selection
Transfer Size	1 Byte	Transfer size selection
Destination Address Mode	Fixed	Destination address mode selection
Source Address Mode	Incremented	Source address mode selection
Repeat Area (Unused in Normal Mode)	Source	Repeat area selection
Interrupt Frequency	After all transfers have completed	Interrupt frequency selection
Destination Pointer	NULL	Destination pointer selection
Source Pointer	NULL	Source pointer selection
Number of Transfers	0	Number of transfers selection
Number of Blocks (Valid only in Block Mode)	0	Number of blocks selection
Activation Source (Must enable IRQ)	Event SCI0 TXI	Activation source selection
Auto Enable	FALSE	Auto enable selection

Callback (Only valid with Software start)	NULL	Callback selection
ELC Software Event Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Disabled	ELC Software Event interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 41 Configuration Settings for the Transfer Driver on r_dtc Event SCI0 RXI

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Selects if code for parameter checking is to be included in the build
Software Start	Enabled, Disabled Default: Disabled	Software start selection
Linker section to keep DTC vector table	.ssp_dtc_vector_table	Linker section to keep DTC vector table selection
Name	g_transfer1	Module name
Mode	Normal	Mode selection
Transfer Size	1 Byte	Transfer size selection
Destination Address Mode	Incremented	Destination address mode selection
Source Address Mode	Fixed	Source address mode selection
Repeat Area (Unused in Normal Mode	Destination	Repeat area selection
Interrupt Frequency	After all transfers have completed	Interrupt frequency selection
Destination Pointer	NULL	Destination pointer selection
Source Pointer	NULL	Source pointer selection
Number of Transfers	0	Number of transfers selection
Number of Blocks (Valid only in Block Mode)	0	Number of blocks selection
Activation Source (Must enable IRQ)	Event SCI0 RXI	Activation source selection
Auto Enable	FALSE	Auto enable selection
Callback (Only valid with Software start)	NULL	Callback selection
ELC Software Event Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Disabled	ELC Software Event interrupt priority selection.

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

Table 42 Configuration Settings for the External IRQ HAL Module on r_icu

ISDE Property	Setting	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Parameter checking setting enables or disables the addition of parameter checking code
Name	g_external_irq0	Module name
Channel	0	Specifies the hardware IRQ channel used
Trigger	Falling	Selection for trigger event mode
Digital Filtering	Disabled	Digital filter enable/disable
Digital Filtering Sample Clock (Only valid when Digital Filtering is Enabled)	PCLK/64	Sets noise filter sampling period
Interrupt enabled after initialization	TRUE	Determines if the interrupt is enabled immediately after initialization
Callback	custom_hw_irq_isr	A user callback function can be registered in external_irq_api_t::open. If this callback function is provided, it is called from the interrupt service routine (ISR) each time the IRQn triggers. **Warning:** Since the callback is called from an ISR, care should be taken not to use blocking calls or lengthy processing. Spending excessive time in an ISR can affect the responsiveness of the system.
Pin interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) Default: Disabled	Pin interrupt priority selection

Note: The example values and defaults are for a project using the Synergy S7G2 MCU. Other MCUs may have different default values and available configuration settings.

5.5 Wi-Fi Framework Module Clock Configuration

The Wi-Fi Framework module uses the clocks required for the specific selections of the low-level modules such as SPI.

5.6 Wi-Fi Framework Module Pin Configuration

The Wi-Fi Framework module uses input and output pins depending on the selections of the low-level modules such as SPI or IRQ.

6. Using the Wi-Fi Framework Module in an Application

The following description is a high-level overview of some typical Wi-Fi use cases. A more detailed description and a working application project (which is too lengthy to include in this document), is available on the Renesas web site. Just search for the associated application note document number, r11an0226eu, in the top page search bar on www.Renesas.com. It is highly recommended that you use the application note to augment the summary descriptions found in this document.

Each of the Wi-Fi Framework implementations are treated differently in a target application. The typical control flow for initialization, packet transmission using NetX/NetX Duo, packet reception using NetX/NetX Duo, and using an on-chip networking stack, are of particular interest. Example flow diagrams for some typical implementations of these functions are shown as follows:

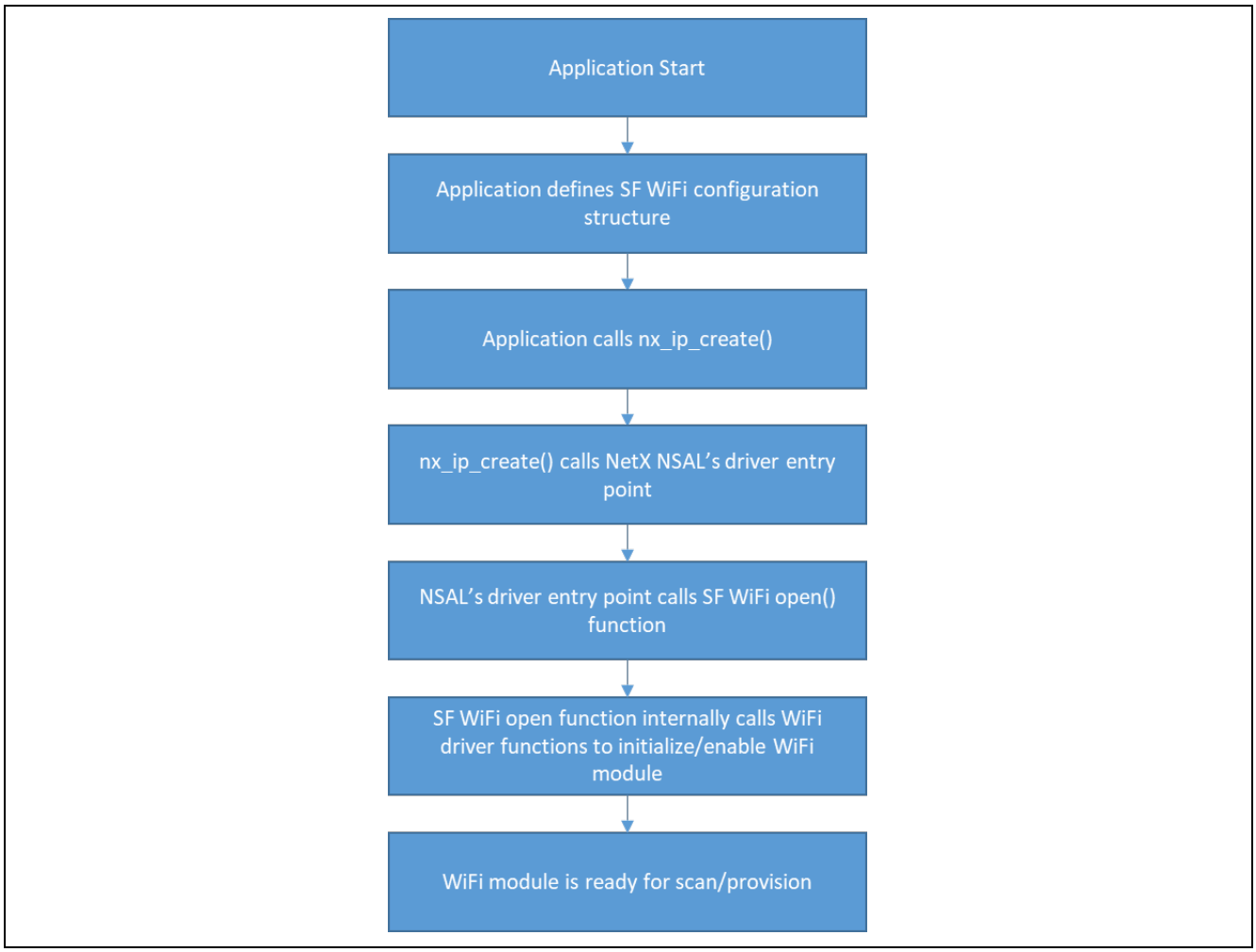

Figure 13 Application Control Flow using Wi-Fi module initialization

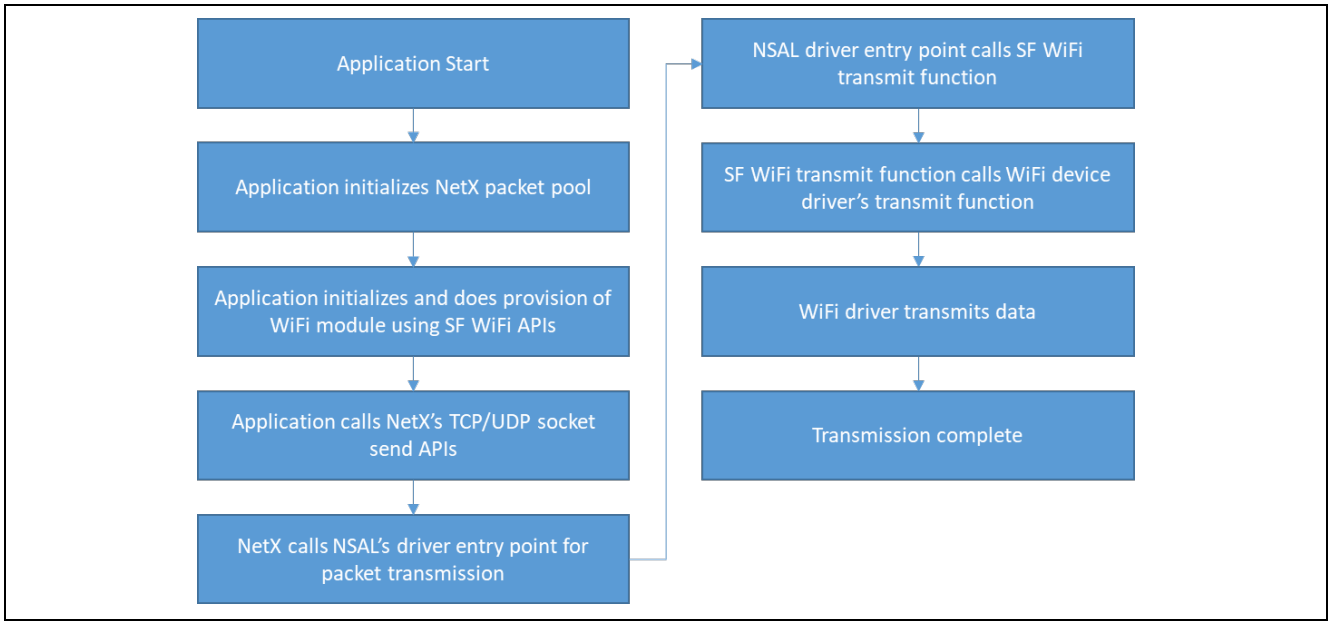

Figure 14 Application Control Flow doing packet transmission using NetX

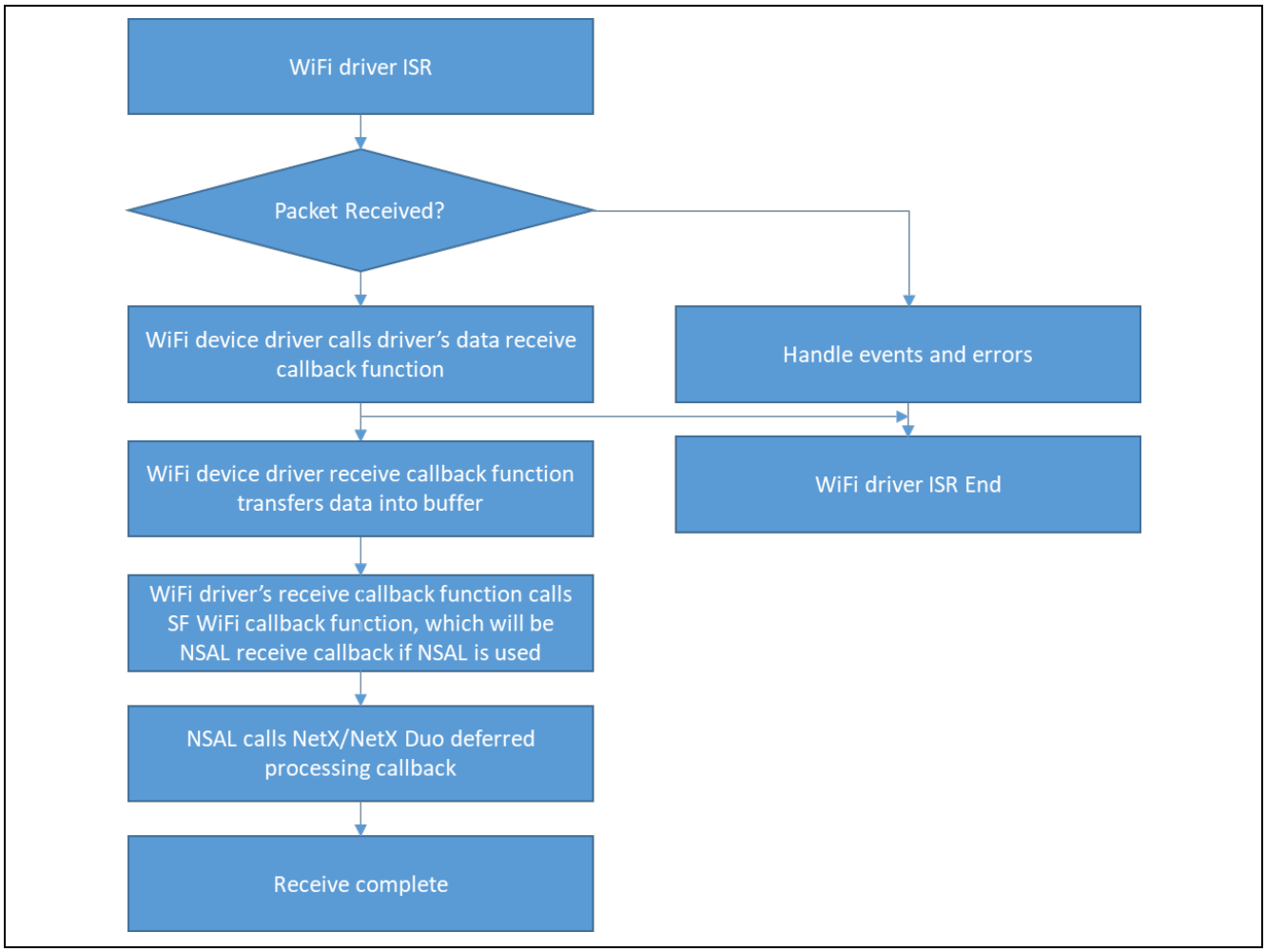

Figure 15 Application control flow receiving packet using NetX

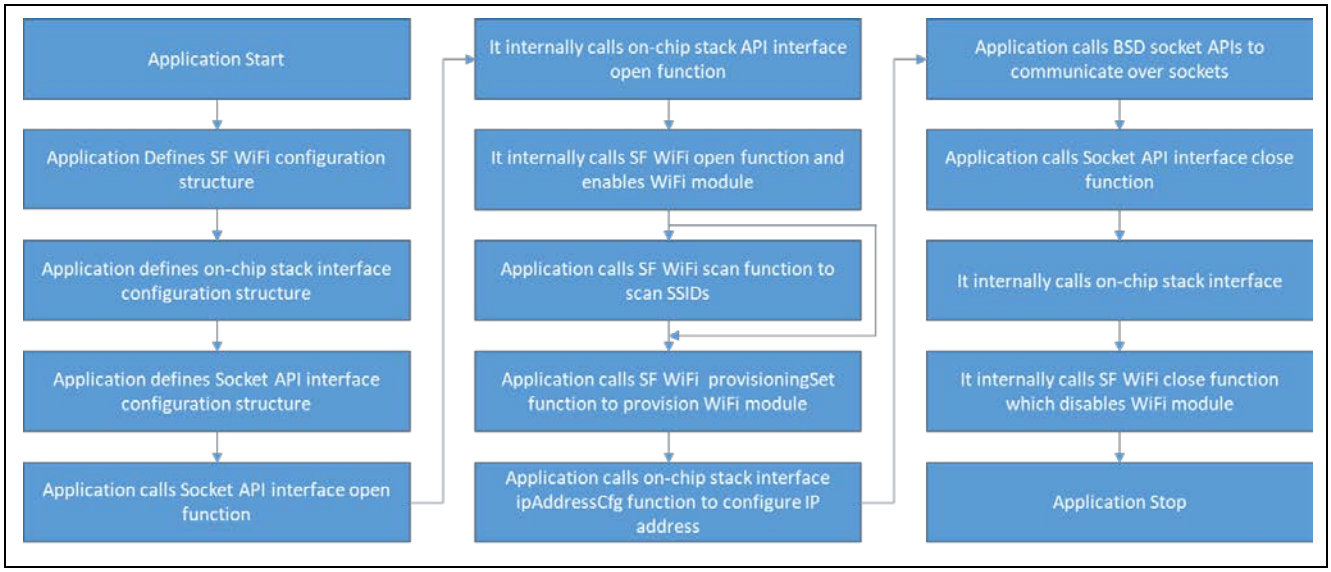

Figure 16 Application Flow Control using On-Chip Networking Stack

7. Wi-Fi Framework Module Application Project

The application project associated with this module guide demonstrates common steps to using the Wi-Fi Framework on the GT202 in an example application. The project can be found using the link provided in the References section at the end of this document. You may want to import and open the application project within the ISDE and view the configuration setting for the Wi-Fi Framework module. You can also read over the code (in wifi_thread_entry.c and wifi_fw_gt202_mg.c), which is used to illustrate the Wi-Fi Framework module APIs in a complete design.

The application project demonstrates the typical use of the Wi-Fi Framework module APIs. To simplify the implementation of Wi-Fi Framework, NetX DHCP Client module is used as the highest-level application protocols which pull in the dependency module such as the NetX IP instance and the NetX Port using Wi-Fi Framework on sf_wifi_nsal_nx to implement Wi-Fi function. By using NetX DHCP Client module, some basic setup operations of UDP will be implemented automatically, such as socket reception and transmission. This high-level protocol usage will significantly reduce customer workload.

This application project is implemented on the SK-S7G2 board with Longsys Wi-Fi module GT202 using the NetX networking stack. Communication with the GT202 device drive is through the NSAL interface. Demo code initializes the NetX TCP/IP and provisions the Wi-Fi module. The DHCP client layer is used to acquire the IP address from the DHCP server. After receiving IP address successfully, you can confirm the networking connectivity by pinging the acquired IP address.

A more detailed API description, along with a working example application of NetX DHCP Client Module, is available on the Renesas web site. Search for the associated application note document number, r11an0138eu, in the search bar on www.Renesas.com. It is highly recommended that you use this module guide to get familiar with NetX DHCP Client module used in this document.

The following table identifies the target versions for the associated software and hardware used by the application project.

Table 43 Software and Hardware Resources Used by the Application Project

Resource	Revision	Description
e2 studio	5.4.0.023 or later	Integrated Solution Development Environment
SSP	1.3.3 or later	Synergy Software Platform
IAR EW for Renesas Synergy	7.71.3 or later	IAR Embedded Workbench for Renesas Synergy
SSC	5.4.0.023 or later	Synergy Standalone Configurator
SK-S7G2	v3.0 to v3.3	Starter Kit
GT202 PMOD board	V1.1	Wi-Fi module

The application project requires connecting a USB micro cable on the J19 (DEBUG_USB) connector to a PC to display the output to the Renesas debug virtual console, and DHCP Client device (GT202 module) that is connected to a network with a DHCP Server through the PMOD A connector (J12).

A simple flow diagram of the application project is given in the following figure:

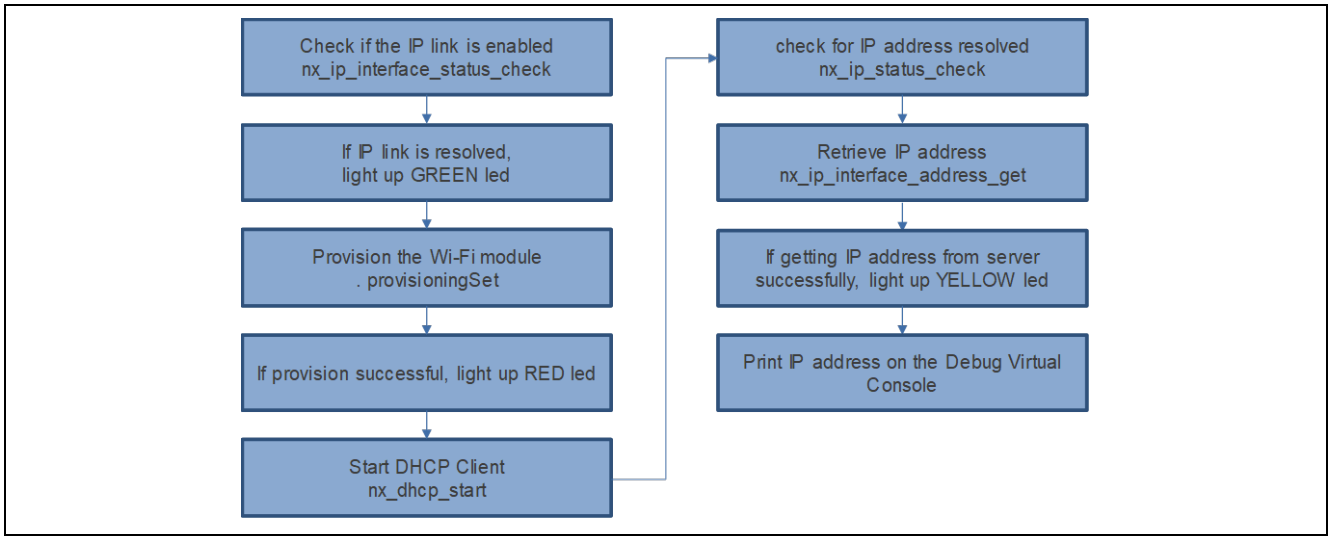

Figure 17 Wi-Fi Framework Module Application Project Flow Diagram

The wifi_fw_gt202_mg.c file is located in the project once it has been imported into the ISDE. Note that this is not the auto-generated code file wifi_thread_entry.c. The wifi_thread_entry calls a function defined in wifi_fw_gt202_mg.c file to run a DHCP Client session and implement Wi-Fi module communication with DHCP server to get dynamic IP address. You can open this file within the ISDE and follow along with the following description to help identify key uses of APIs.

1. Check if the IP link is enabled with **nx_ip_interface_status_check(&g_ip0, 0, NX_IP_LINK_ENABLED, &ip_status, 100).
2. Provision the Wi-Fi module with g_sf_wifi0.p_api->provisioningSet (g_sf_wifi0.p_ctrl, &g_provision_info)
   Notes: 1. This sample application has fixed setting for the following provision configurations:
      - Mode: Client
      - Security: WPA2
      - Encryption: Auto

      Slight adjustment of provision configurations can allow provision for Wi-Fi modules that have different security settings.
   2. SSID and password of DHCP server has been fixed in this project:
      - SSID: Renesas Router
      - Password: 12345678

      Please change the information according to the actual server's configuration.
3. Start the DHCP client with nx_dhcp_start(&g_dhcp_client0).
4. Check the DHCP server to determine if the IP address is resolved with **nx_ip_status_check(&g_ip0, NX_IP_ADDRESS_RESOLVED, (ULONG *) &ip_status, 100).
5. Acquire the leased IP address with nx_ip_interface_address_get(&g_ip0,0,&ip0_ip_address,&ip0_mask).
   Notes: 1. In the above steps, if the operation fails, trap the error and do an infinite loop. Otherwise, print success message on Renesas Debug Virtual Console and light up related led.
   2. This description assumes you are familiar with using printf() with the Debug Console in the Synergy Software Package. If you are unfamiliar with this, refer to the "How do I Use Printf() with the Debug Console in the Synergy Software Package" Knowledge Base article. Alternatively, you can see results via the watch variables in the debug mode.

      ** nx_ip_interface_status_check() has one more parameter than nx_ip_status_check(), interface_index, which is used to specify interface index number. For more information, please refer to the *NetX User Guide*.

A few key properties are configured in this application project to support the required operations and the physical properties of the target board and MCU. The properties with the values set for this specific project are listed in the following tables and figures. You can also open the application project and view these settings in the Properties window as a hands-on exercise.

In this section, only properties that changed from the default configuration are described in detail.

For the following blocks, there are no changes from the default setting:

- NetX DHCP Client
- NetX DHCP Common
- NetX IP Instance
- NetX Common on NetX
- NetX Package Pool Instance
- NetX Port using Wi-Fi Framework on sf_wifi_nasl_nx

Note: Some parameters in above blocks, such as Wi-Fi thread stack size and priority need to be adjusted according to the actual application.

The properties that change from the default value are marked in the red boxes.

- GT202 Wi-Fi Device Drive on sf_wifi_gt202

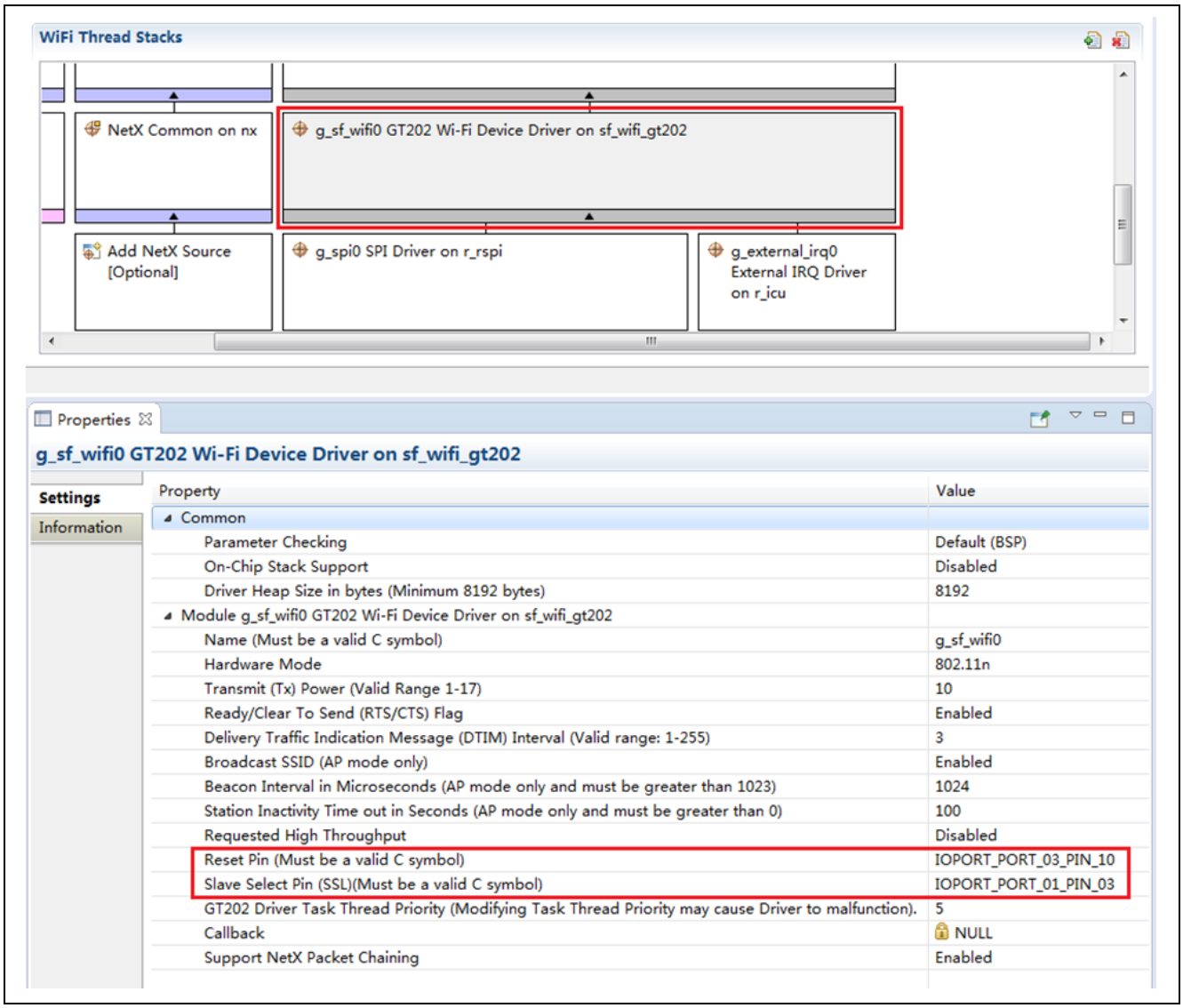

Figure 18 Wi-Fi Device Driver Configuration

The following table provides additional information on the properties that differ from the default values.

Table 44 Wi-Fi Device Driver Configuration in the Application Project

ISDE Property	Value	Description
Reset Pin	IOPORT_PORT_03_PIN_10	Hardware reset pin to GT202 module
Slave Select Pin	IOPORT_PORT_01_PIN_03	Hardware slave select pin to GT202 module

To configure these two pins as reset pin and slave select pin for the Wi-Fi module, go to **Pins** tab and find related pins for reconfiguration. The following tables list the setting of these pins.

Table 45 Pin Selection Sequence for Wi-Fi Framework Module on SPI

Resource	ISDE Tab	Pin Selection Sequence
SPI	Pins	Select Peripherals > Connectivity:SPI > SPI0

Table 46 Pin Configuration Settings for Wi-Fi Framework Module on SPI

Pin Configuration Property	Value
Pin Group Selection	_A only
Operation Mode	Enable
MOSI	P100
MISO	P101
RSPCK	P102
SSL0	None
SSL1	None
SSL2	None
SSL3	None

Table 47 Pin Selection Sequence for Reset Pin and Slave Select Pin

Resource	ISDE Tab	Pin Selection Sequence
GPIO	Pins	Select Ports > P1 > P103 (Slave Select Pin)
GPIO	Pins	Select Ports > P3 > P310 (Reset Pin)

Table 48 Pin Configuration Settings for Reset Pin and Slave Select Pin

Pin Configuration Property	Value
Mode	Output mode (Initial High)
Pull up	None
Drive Capacity	Medium
Output type	CMOS

SPI Driver on r_rspi

The GT202 uses SPI to communicate with the MCU. To understand the configurations of the SPI peripheral, see the *SPI Module Guide*. This application project uses the RSPI interface to communicate with GT202. Add the RSPI block as shown in the following figure. To achieve fast communication, Bitrate in SPI Driver on r_rspi has been increased.

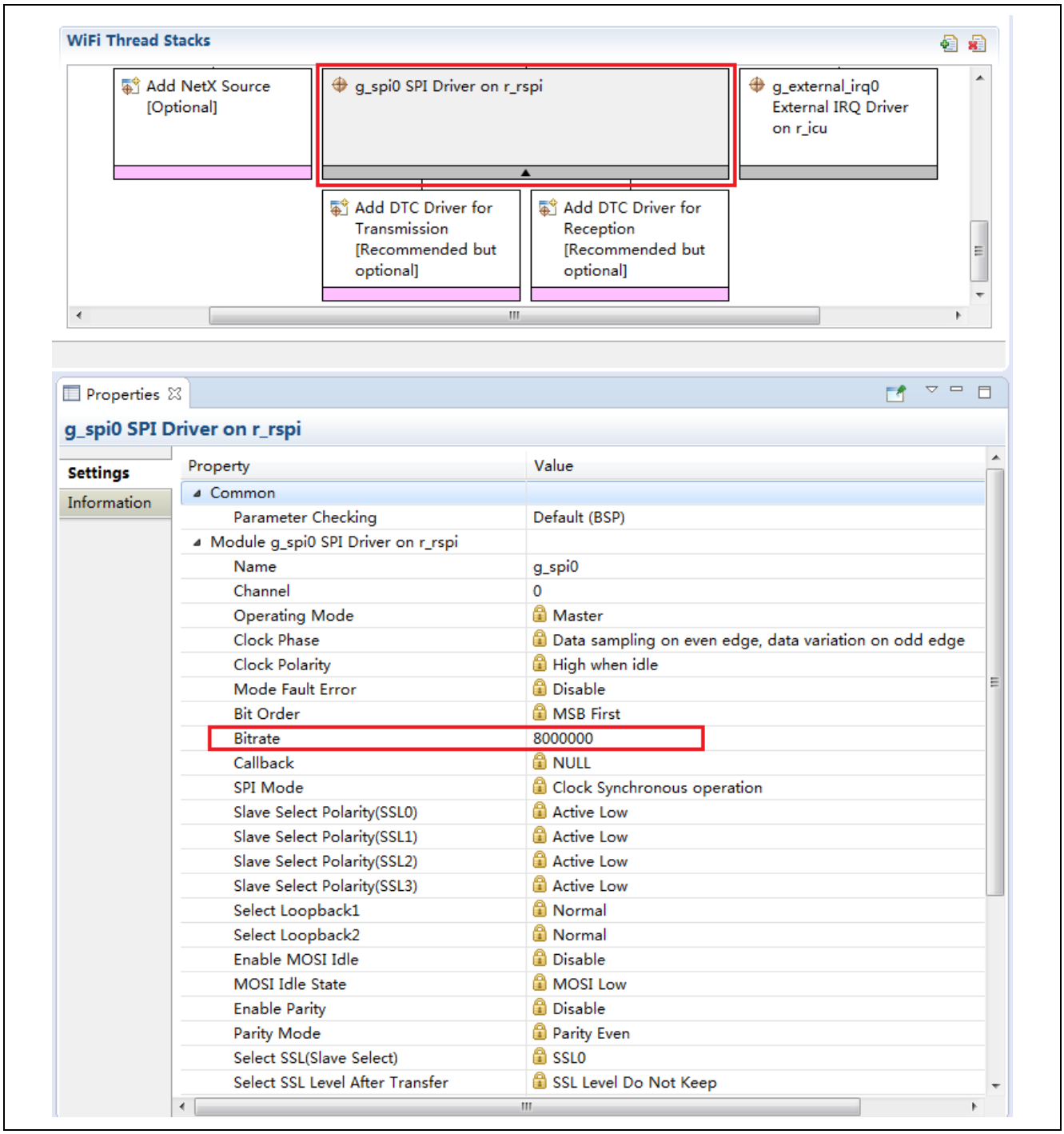

Figure 19 RSPI Configuration

See the *RSPI HAL Module Guide* for detailed description of the R_RSPI configuration properties. When the RSPI is pulled in to complete the GT202 device driver, the Synergy configurator automatically adjusts several settings and has most of the settings locked to avoid misconfigurations.

Note: When using RSPI, delete DTC components that are auto-filled for the dependencies which include Transfer Driver on r_dtc Event SPI0 TXI and Transfer Driver on r_dtc Event SPI0 RXI under SPI Driver on r_rspi module.

External IRQ Driver on r_icu

The following figure shows the configuration for PMOD A interrupt pin setup. Because PMOD A uses IRQ channel 4 as external interrupt input pin, change channel from 0 to 4. Meanwhile, select suitable interrupt level for R_ICU.

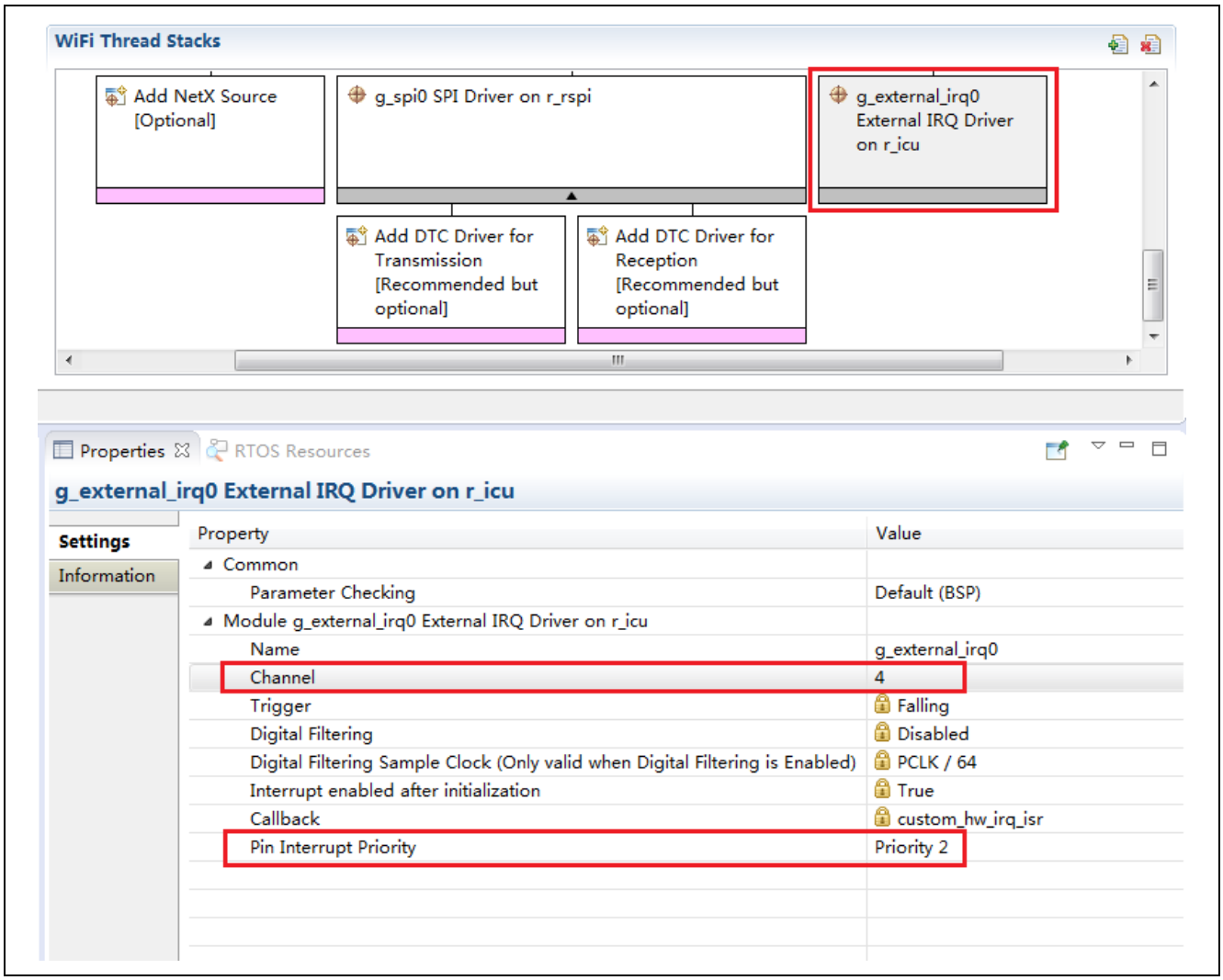

Figure 20 PMOD A Interrupt Configuration

See the *External IRQ HAL Module Guide* for detailed description of the external IRQ configuration properties. When the external IRQ is pulled in to complete the GT202 device driver, the Synergy configurator automatically adjusts several settings and has most of the settings locked to avoid misconfigurations.

8. Customizing the Wi-Fi FW Module for a Target Application

Some configuration settings will normally be changed from those shown in the application project. For example, you can easily change the configuration settings for thread stack size and priority in thread tabs, Wi-Fi module related settings in stacks. You can also change the SPI port pins to select the desired communication interface; this can be done using the **Pins** tab in the configurator.

9. Running the Wi-Fi FW Module Application Project

To run the Wi-Fi Framework module application project and to see it executed on a target kit, you can simply import it into your ISDE, compile, and run debug.

Note: The following steps are described in sufficient detail for someone experienced with the basic flow through the Synergy development process. If these steps are not familiar, refer to the first few chapters of the *SSP User's Manual* for a description of how to accomplish these steps.

To create and run the application project, follow these steps:

1. Import and build the example project included with this module guide according to the *Renesas Synergy™ Project Import Guide*, r11an0023eu0120-synergy-ssp-import-guide.pdf.
2. Connect to the host PC using the USB cable (use J19 DEBUG_USB connector).
3. Start to debug the application.
4. The output can be viewed in the Renesas Debug Virtual Console.

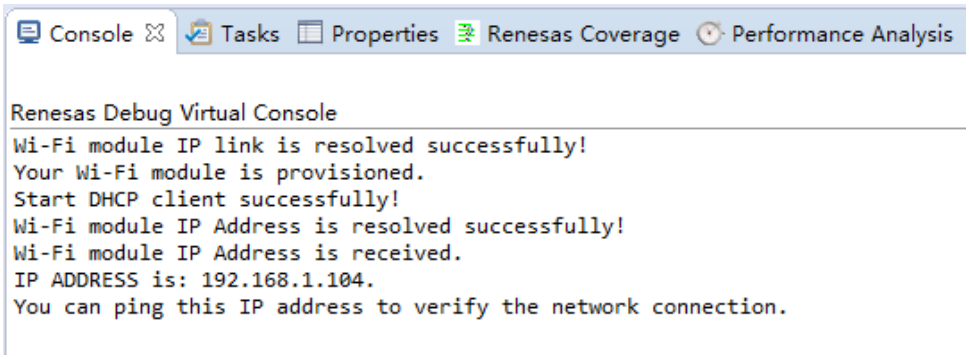

Figure 21 Example Output from Wi-Fi FW Module Application Project

Note: Make sure you have a Wi-Fi access point and have modified SSID and password in this project according to access point's information. The definition of SSID and password locates in the wifi_fw_gt202_mg.h, and you can open it and modify the definition according to actual application. Also confirm that GT202 has been pulled into PMOD A before supplying power to SK-S7G2 board. Make sure that 3.3V is selected for PMOD A using jumper (J13), as shown in the following figure.

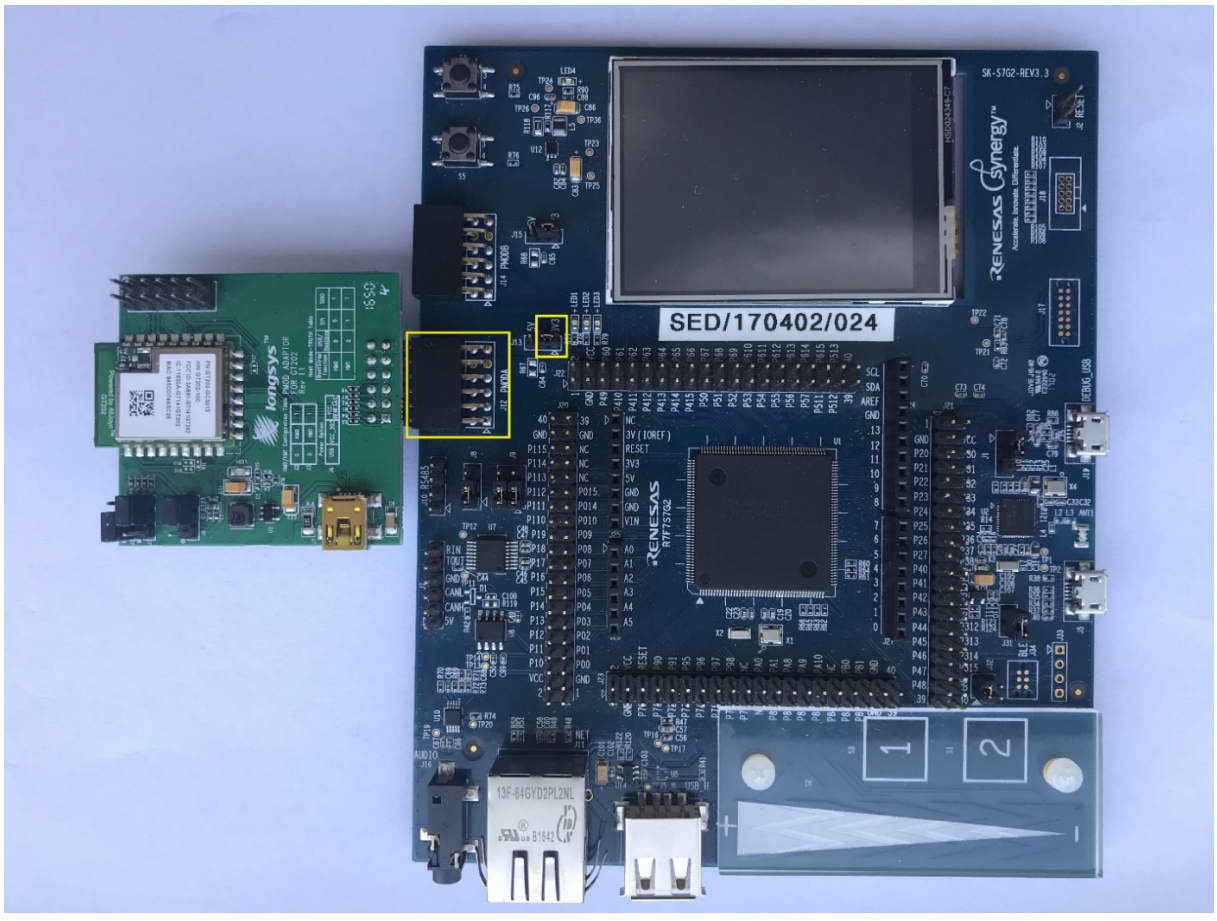

Figure 22 Hardware Setup

After the GT202 is provisioned and you got the IP address from the console window, open a command window and type the following message as an example (you should use the IP address printed on the console from your setup):

>ping 192.168.1.104

The figure below shows an example of a successful ping result. This proves a successful link connection.

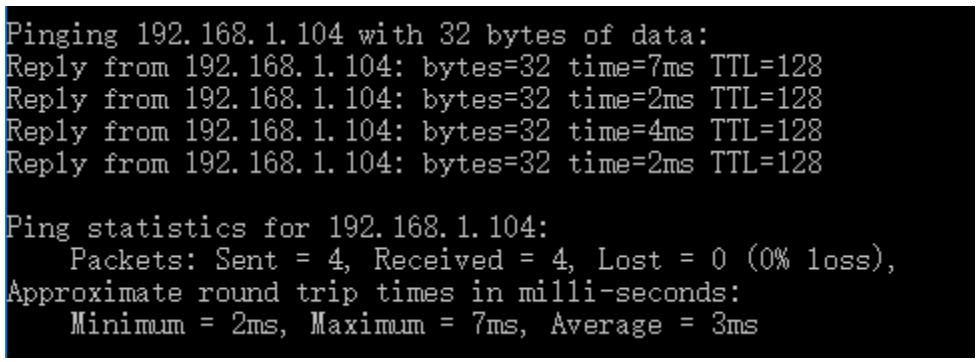

Figure 23 Ping the Wi-Fi Module

10. Wi-Fi FW Module Conclusion

This module guide has provided all the background information needed to select, add, configure, and use the module in an example project. Many of these steps were time consuming and error-prone activities in previous generations of embedded systems. The Renesas Synergy Platform makes these steps much less time consuming and removes the common errors like conflicting configuration settings or incorrect selection of lower-level drivers. The use of high-level APIs (as demonstrated in the application project) illustrates additional development-time savings by allowing work to begin at a high level and avoiding the time required in older development environments to use, or, in some cases, create, lower-level drivers.

11. Wi-Fi FW Module Next Steps

After you have mastered a simple Wi-Fi Framework module project, you may want to review a more complex example. Other application projects and application notes that demonstrate Wi-Fi implementation and DHCP operation can be found as described in the References section at the end of this document.

12. Wi-Fi FW Module Reference Information

SSP User Manual: Available in html format in the SSP distribution package and as a pdf from the Synergy Gallery.

Links to all the most up-to-date Wi-Fi Framework module reference materials and resources are available on the Synergy Knowledge Base: https://en-us.knowledgebase.renesas.com/English_Content/Renesas_Synergy™_Platform/Renesas_Synergy_Knowledge_Base/sf_wifi_Module_Guide_Resources.

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Aug 20, 2018	-	Initial version

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.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 7.1