AN1435: SiWx917 NCP Firmware Update Application Note

Firmware is a software that executes on an SiWx917 chip. Firmware update to a new version upgrades your device with newly added features, bug fixes and provides enhanced user as well as developer experience.

This application note describes when and how to update the Firmware of SiWx917. It also explains different methods to update firmware using the firmware update examples in the WiSeConnect™ SDK v3.x (hereafter referred to as SDK v3.x).

KEY POINTS

- Notes on SiWx917 firmware.
- When to update Firmware?
- How to update Firmware?
- Firmware Update methods
- Refer to Firmware Update Examples in the WiSeConnect™ v3.x SDK.

1. About SiWx917's Firmware

The SiWx917 employs the Network Wireless Processor (NWP) processor using the Firmware present in the flash. The SiWx917's Firmware is an RPS formatted file. The RPS format is a binary executable format understood by the NWP Bootloader to verify the integrity and authenticity of the Firmware Image. If the Firmware Image is valid, the Bootloader executes the Firmware Image.

The Firmware Image includes an RPS header, Boot descriptor, Applications binary image, and an optional trailer (Digital Signature).

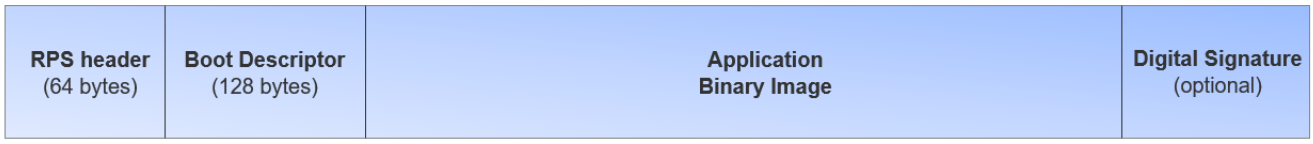


Figure 1.1. SiWx917 Firmware File Format

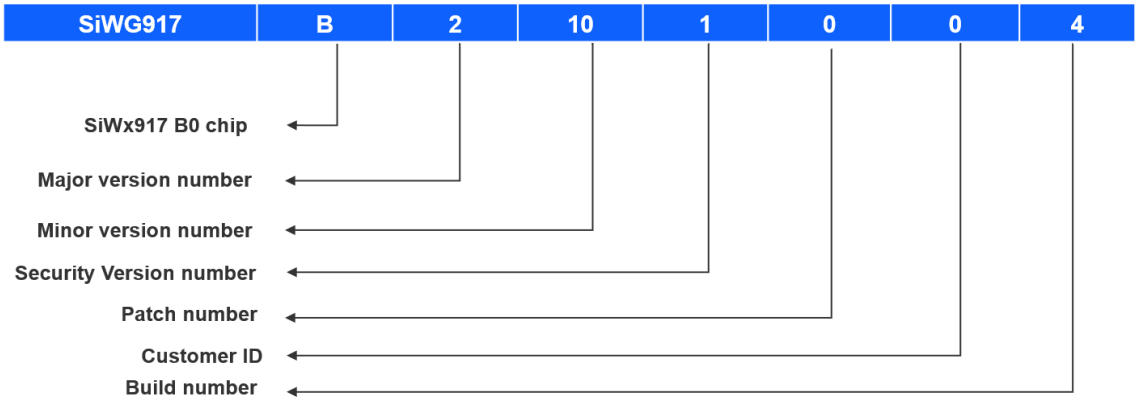
Note:

The phrases "firmware image" and "firmware" mean the same throughout the document.

1.1 Firmware File Versioning

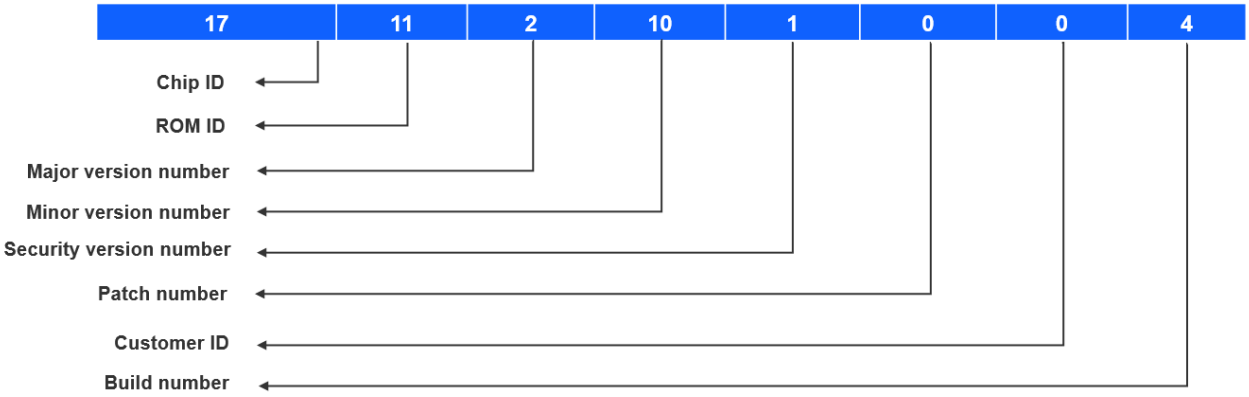
The NWP Connectivity Firmware file - SiWG917-B.u.v.w.x.y.z.rps is located at <WiSeConnect v3.x SDK> → connectivity_firmware path of v3.x SDK. The maximum size of SiWx917's Firmware File is about 1.8 MB.

The following figure explains what each field in the Firmware File name specifies.



1.2 Firmware Versioning

This following figure explains what each field in the Firmware version specifies when sl_wifi_get_firmware_version() API is called.



NOTE:

1. To check the SiWx917's current NWP Firmware version, call the sl_wifi_get_firmware_version() API after initializing the device (after sl_net_init() API call). If the Firmware loaded in the SiWx917's flash is SiWG917-B.u.v.w.x.y.z.rps, the API returns the Firmware version as 1711.u.v.w.x.y.z.
2. Currently, in the v3.1.1 SDK, the Firmware file version is SiWG917-B.2.10.0.0.4.rps and it does not include the Security version number. This is corrected in the later releases of SDK v3.x. The firmware version obtained using the sl_wifi_get_firmware_version() API call shall include a Security version number field (for example, 1711.2.10.1.0.0.4). Whenever the Security-related vulnerabilities in the SiWx917 firmware are found and fixed, the Security version number shall be incremented.

2. When to Update Firmware?

The SiWx917 ICs, modules, or development (expansion) kits, when received for the first time, do not contain any firmware by default.

The SiWx917 connectivity firmware version is tightly coupled with the WiSeConnect 3 extension version. You must update the SiWx917 connectivity firmware when:

- you first receive an SiWx917 EXP expansion kit,
- you first receive an SiWx917 co-processor radio board, or
- you upgrade or downgrade your WiSeConnect 3 extension.

When the SiWx917 NCP does not contain any Firmware or any valid Firmware, the only possible method to update the firmware is Firmware Update via Bootloader. When the SiWx917 has a valid firmware, you many use either Firmware Update via Bootloader or Over The Air (OTA Firmware Update based on your choice. The details of these methods and their relevant examples are explained in Firmware Update Methods and Firmware Update Examples sections of the document.

3. Firmware Update Methods

There are two ways to update the SiWx917's NWP firmware.

- Firmware Update via Bootloader
- Over The Air (OTA) Firmware Update

Before going through the methods, consider the following figure which illustrates where in NWP Flash memory map, the Firmware Image and Firmware Image Backup areas reside. The Firmware Image location is the target location from where the Firmware executes, and the Firmware Image backup location is used for storing the new Firmware Image temporarily during Firmware update. The Bootloader resides in the ROM.

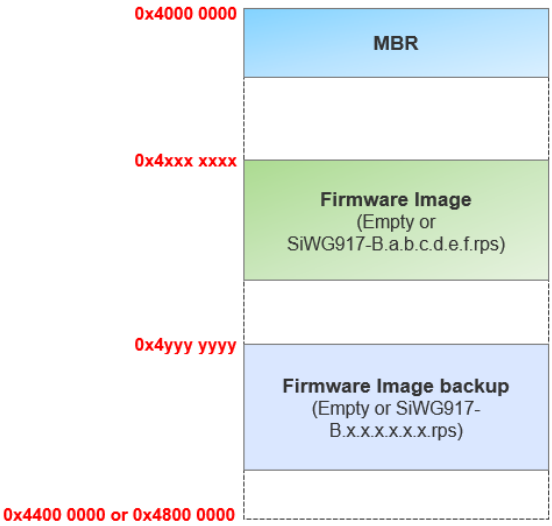


Figure 3.1. 4 MB/8 MB NWP Flash Memory map

3.1 Firmware Update via Bootloader

The Bootloader is the first piece of code that executes when the SiWx917 is powered up or reset. It controls the initial operation of the device. In this method, upon SiWx917's power-up, the firmware update process is carried out by the Bootloader through message exchanges between SiWx917 and host over SPI/UART/SDIO interface. The host can be Tera Term or an external host MCU application.

In this method, the NWP Bootloader can update Firmware Image in one of the two modes:

- Non-safe or Fast Update mode
- Safe Update mode

The Fast or Safe Update mode can be configured using an SDK v3.x API. This API is not yet implemented.

3.1.1 Non-Safe or Fast Update Mode

The Non-safe or Fast Update mode is a single-stage Firmware update process that allows the Firmware Image to be placed into the target location of flash memory, overwriting the existing Firmware Image.

Procedure:

1. Initially, the SiWx917 can be without any firmware (when received for the first time) or can be with some existing firmware say SiWG917-B.a.b.c.d.e.f.rps.
2. Upon Power up, the SiWx917 boots into Bootloader.
3. The SiWG917-B.u.v.w.x.y.z.rps is transferred in chunks from host to Bootloader.
4. As soon as the Bootloader receives a chunk, it transfers the chunk to the Firmware Image location.
5. After the complete Firmware Image is transferred, the Bootloader verifies the Integrity and Authenticity of the new Firmware Image.
6. If the Image is valid, the Bootloader executes the new Firmware Image - SiWG917-B.u.v.w.x.y.z.rps, else if the Image is invalid, the Bootloader does not execute the Image.

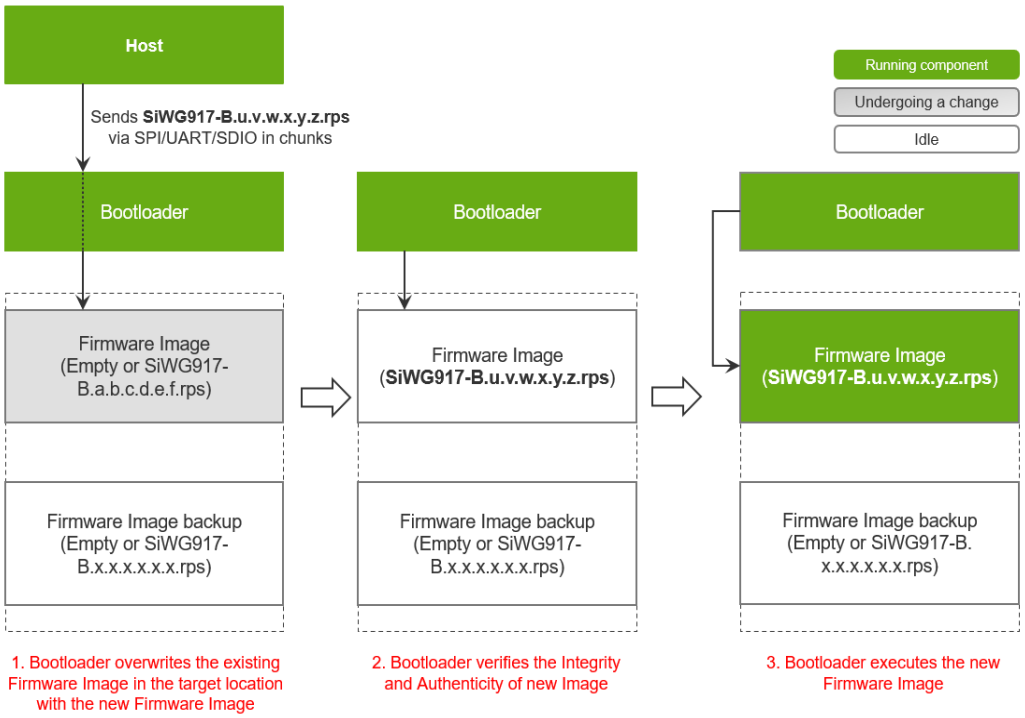


Figure 3.2. Non-Safe Firmware Update Process

Note:

In case of the Non-safe Update mode, if the Firmware Image Update is interrupted for any reason, the device shall be left without a functional and valid firmware. During the next power up, the bootloader detects that the existing Firmware Image is not valid anymore and the SiWx917 enters the firmware update mode.

3.1.2 Safe Update Mode

The Safe Update mode is a two-stage Firmware Update process where the Bootloader places the Firmware Image in a separate download location called “Firmware Image backup” temporarily and then does the integrity and authenticity check of the received image. If the image is valid, the Bootloader replaces the current Firmware Image with the newly verified Firmware Image.

Procedure:

1. Initially, the SiWx917 can be without any firmware (when received for the first time) or can be with some existing firmware say SiWG917-B.a.b.c.d.e.f.rps.
2. Upon Power up, the SiWx917 boots into Bootloader.
3. The SiWG917-B.u.v.w.x.y.z.rps is transferred in chunks from host to Bootloader.
4. The Bootloader transfers these chunks to Firmware Image backup location of flash in parallel.
5. After the complete Firmware Image is transferred, the Bootloader verifies the Integrity and Authenticity of the new Firmware Image.
6. If the Image is valid, the Bootloader will move the new Firmware Image to the target location from where it would run and erase the old firmware data.
7. The Bootloader loads and executes the new Firmware Image - SiWG917-B.u.v.w.x.y.z.rps, else if the Image is invalid, the Bootloader does not execute the Image.

Note:

In case of Safe Update mode, if the firmware update process is interrupted for any reason, the existing Firmware Image is not affected but the whole process of updating to a new image will have to be repeated from the beginning.

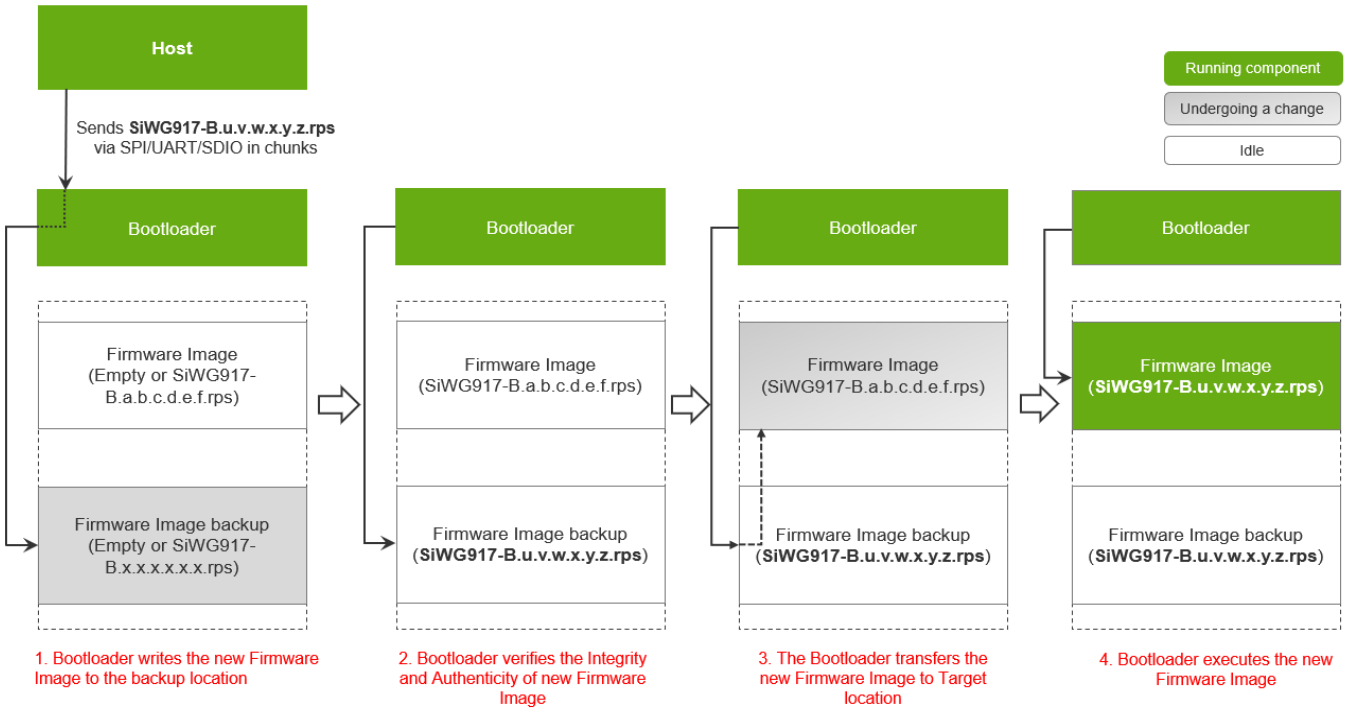


Figure 3.3. Safe Firmware Update Process

3.2 Over The Air (OTA) Firmware Update

OTA Firmware Update is always performed in a Safe Update mode. In this method, the Firmware file is hosted on the remote Server/ Cloud and SiWx917 downloads the Firmware Image wirelessly over a network or Internet.

Procedure:

1. The firmware file is hosted at a Remote Server/Cloud.
2. The SiWx917 should have a valid Firmware Image to perform OTA-based Firmware Update.
3. The host application initializes the SiWx917, drives it to connect to a Wi-Fi Network and indeed connect to the Remote Server/ Cloud via TCP or HTTP.
4. The host application initiates a file download and SiWx917 receives the Firmware Image in chunks.
5. As soon as it receives a chunk, the SiWx917 firmware writes it to the Firmware Image backup location of flash.
6. After receiving all the chunks, the Firmware verifies the new Firmware Image by verifying the Integrity and Authenticity of the new Firmware Image based on the RPS Header configuration.
7. If the image is valid, the current firmware informs the same to host and the host does a soft reset to allow SiWx917 to boot into the bootloader.
8. The Bootloader finds that a new firmware is available in the Firmware Image backup location and again does an Integrity and Authenticity verification of the Firmware Image.
9. After the image is verified, if the image is valid, the Bootloader transfers the Firmware Image from the Firmware Image backup location to the target location.

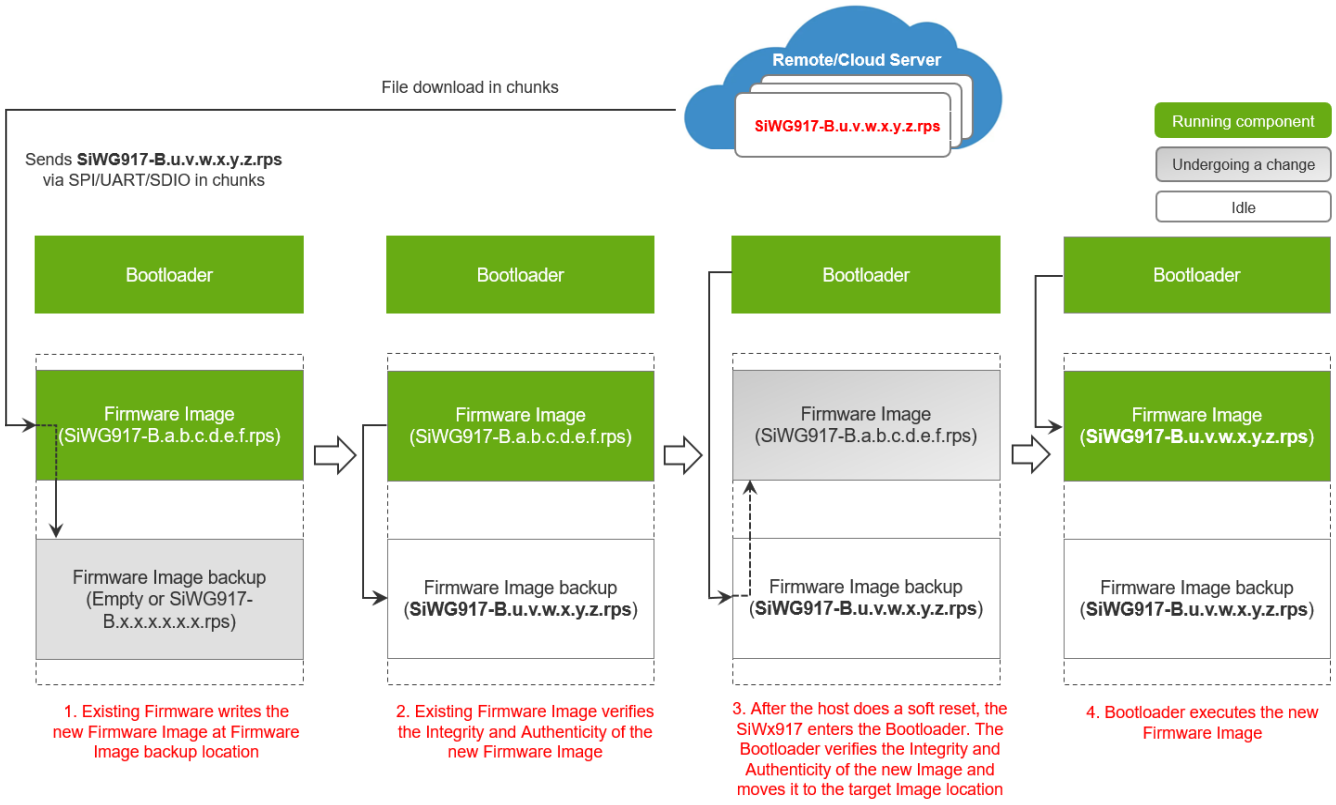


Figure 3.4. OTA Firmware Update Process

4. Firmware Update Examples

4.1 Examples for Firmware Update via Bootloader

The SiWx917's Bootloader and host interact with each other to perform firmware update over SPI/UART/SDIO interface. The host can be Tera Term or a host MCU application.

4.1.1 Using Simplicity Commander as Host

In this method, the SiWx917's Bootloader and Simplicity Commander host interact with each other via UART Interface. Simplicity Commander transfers the firmware file present in your local drive to SiWx917's Bootloader via UART. To update the Firmware using Simplicity Commander, follow the steps mentioned in Getting Started with EFR32 in NCP mode.

4.1.2 Using a Host MCU Application

In the method, the host MCU application and SiWx917 NCP interact with each other through SPI/SDIO/UART. The reference example is Firmware Flashing from Host UART. Please check SiWx917 NCP release notes to know the working status of the example.

4.2 Examples for OTA Firmware Update

This section details on the OTA Firmware Update examples available in the v3.x SDK. Listed below are the examples:

- OTA-based Firmware Update from the host
- OTA-based Firmware Update

4.2.1 OTA-Based Firmware Update from the Host

In this method, the SiWx917 downloads the Firmware file in chunks and forwards the chunks to host application. The host application then forwards the chunks to SiWx917 firmware, which in turn writes them to Firmware Image backup location. The reference example in the SDK v3.x is Firmware Update via a TCP server. In this example application, the SiWx917, as a TCP client receives the Firmware file in chunks from a remote TCP server application. Each chunk is 1024 bytes long. When a Firmware file of 1.6 MB is considered, the Firmware file would be transferred in about 1638 chunks.

The detailed application flow is as follows:

- The host application initializes the SiWx917's Network client Interface by calling sl_net_init() API.
- The host application drives the SiWx917 to connect to a Wi-Fi network. The SiWx917 connects to a Wi-Fi network and gets an IP address.
- The host application gets the current firmware version of SiWx917 by calling sl_wifi_get_firmware_version() API.
- The SiWx917 establishes a TCP connection with the remote TCP server application that hosts the SiWx917's firmware File.
- The host application makes a request to the server to send the RPS header of the Firmware file (referred to as SL_FWUP_RPS_HEADER in the application), which is usually 64 bytes long.
- The TCP server application sends the RPS header of the firmware file.
- Upon receiving the RPS header, the host application sends it to the SiWx917's firmware by calling sl_si91x_fwup_start() API.
- The host application now makes a request to the TCP server to send the RPS content (SL_FWUP_RPS_CONTENT) or Firmware Image in chunks.
- The TCP server application sends Firmware chunk, which is 1024 bytes long.
- For receiving a Firmware Image chunk, the host application makes a request to the TCP server to send a Firmware Image chunk.
- As soon as the host application receives a Firmware Image chunk, it sends it to SiWx917's firmware by calling sl_si91x_fwup_load() API. The SiWx917 then writes it to the Firmware Image backup location.
- The process continues until the host application receives all the firmware chunks.
- After receiving the new Firmware Image, the existing Firmware verifies the Integrity and Authenticity of new Firmware Image based on the RPS header. If the firmware is valid, the sl_si91x_fwup_load() returns SL_STATUS_FW_UPDATE_DONE (0x10003) for the last chunk, else the sl_si91x_fwup_load() returns an error code (0x10004).
- Upon receipt of SL_STATUS_FW_UPDATE_DONE from SiWx917, the host application closes the TCP socket connection and calls sl_net_deinit() API to de-initialize the Network Interface and enable the SiWx917 to boot into the Bootloader.
- The Bootloader finds a new Firmware Image in the Firmware Image backup and does Integrity and Authenticity verification of new Firmware Image based on the RPS header and Master Boot Record (MBR) configuration. If the Firmware Image is valid, the Bootloader transfers it to the target Firmware location, else it doesn't.
- The host application then calls sl_net_init() API to enable the Bootloader to execute the new Firmware Image, if it is valid or executes the previous firmware, if the new Firmware Image is invalid.
- The host application gets the current firmware version of SiWx917 by calling sl_wifi_get_firmware_version() API. This shall give the new Firmware Image's version number if its valid or else gives the previous Firmware Image's version number.

For more details on the example application, refer to Firmware Update example in the v3.x SDK.

The following flow chart illustrates the Firmware Update via TCP Server application flow:

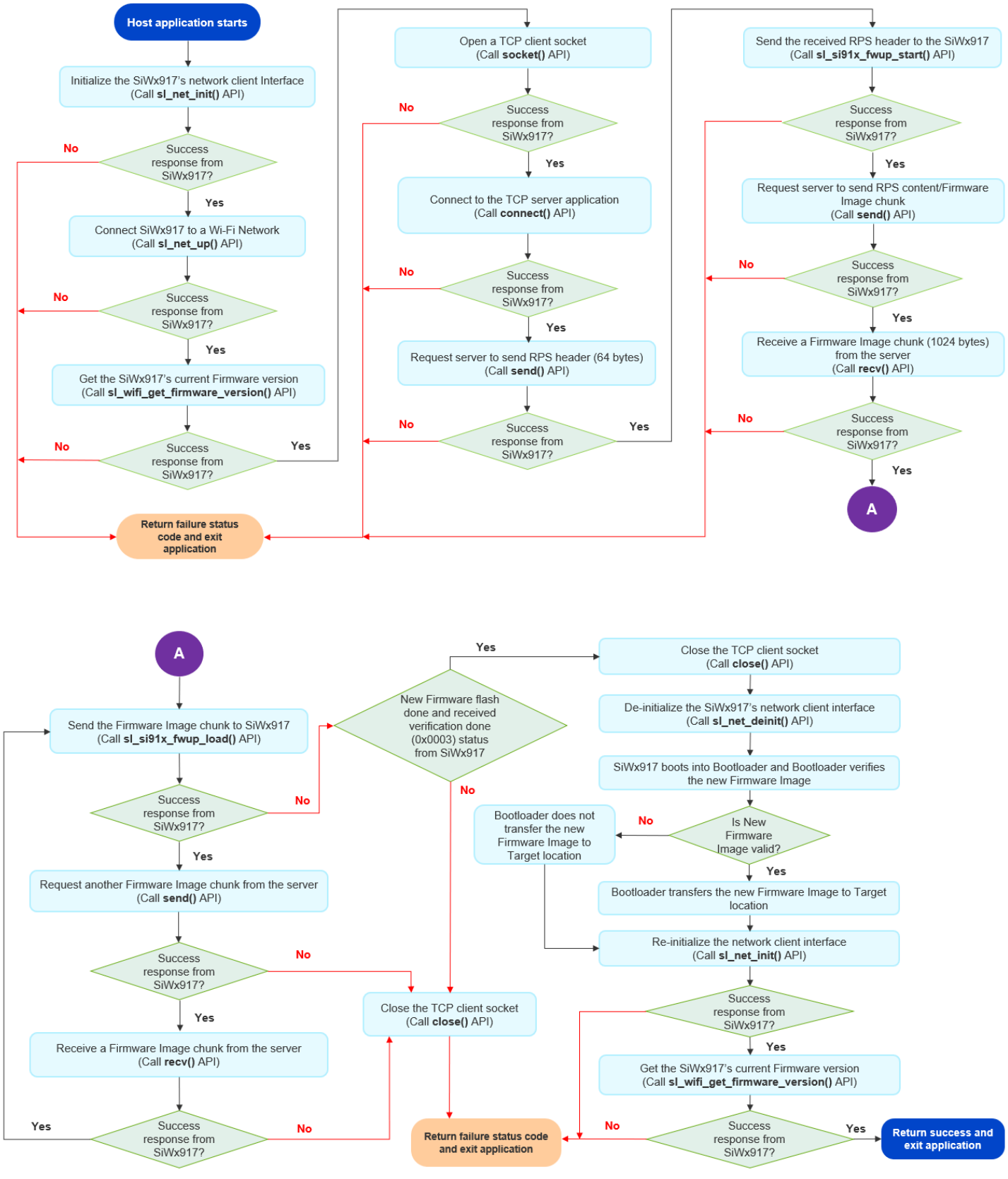


Figure 4.1. Firmware Update via TCP Server Application Flowchart

4.2.2 OTA-Based Firmware Update

In this method, the host application initiates a Firmware download and SiWx917 downloads the Firmware file in chunks and directly writes them to Firmware Image backup location without forwarding to the host. The reference example in the v3.x SDK is Firmware Update via HTTP/HTTPS server. In this example application, the SiWx917, as an HTTP client initiates a Firmware file download from a remote HTTP or HTTPS (secures HTTP Request and Responses via TLS) server that hosts the Firmware Image file with which the SiWx917 is to be updated. This application uses Apache server as an HTTP or HTTPS server or AWS cloud storage services such as AWS S3 bucket or Azure Blob Storage as an HTTPS server.

The detailed application flow is as follows:

- The host application initializes the SiWx917's Network client Interface by calling sl_net_init() API.
- In case of HTTPS-based connection, the host application loads the certificates required for establishing a secure connection with the HTTP server by calling sl_net_set_credential() API.
- The host application drives the SiWx917 to connect to a Wi-Fi network. The SiWx917 connects to a Wi-Fi network and gets an IP address.
- The host application gets the current firmware version of SiWx917 by calling sl_wifi_get_firmware_version() API.
- It then registers a callback function for SL_WIFI_HTTP_OTA_FW_UPDATE_EVENTS by calling sl_wifi_set_callback() API. Whenever the SiWx917's current firmware completes the verification of new Firmware Image, it triggers this callback function to notify Firmware Update success or failure status.
- In case of AWS S3 bucket or Azure Blob Storage, it is required to resolve the AWS Server/Azure Server's host names to get the Server IP address. The host application resolves the host names by calling sl_net_host_get_by_name() API.
- Next, the host application initiates a Firmware file download by calling sl_si91x_http_otaf() API.
- The SiWx917 starts downloading the Firmware File in chunks. As soon as the SiWx917 receives a Firmware Image chunk, it writes it to the Firmware Image backup location. This process continues until the entire Firmware file is downloaded to the backup location.
- Once the download is complete, the current Firmware verifies the Integrity and Authenticity verification of new Firmware Image. The SiWx917 notifies the Firmware Image validity as a response to sl_si91x_http_otaf() API.
- The host application then calls sl_net_deinit() API to de-initialize the Network Interface and enable the SiWx917 boot into the Bootloader.
- The Bootloader finds a new Firmware Image in the Firmware Image backup and verifies Integrity and Authenticity verification of new Firmware Image. If the Firmware Image is valid, the Bootloader transfers it to the target Firmware location, else it does not.
- The host application then calls sl_net_init() API to enable the Bootloader to execute the new Firmware Image, if it is valid or executes the previous firmware, if the new Firmware Image is invalid.
- The host application gets the current firmware version of SiWx917 by calling sl_wifi_get_firmware_version() API. This shall give the new Firmware Image's version number if its valid or else gives the previous Firmware Image's version number.

The following flow chart illustrates the Firmware Update via HTTP/HTTPS Server application flow:

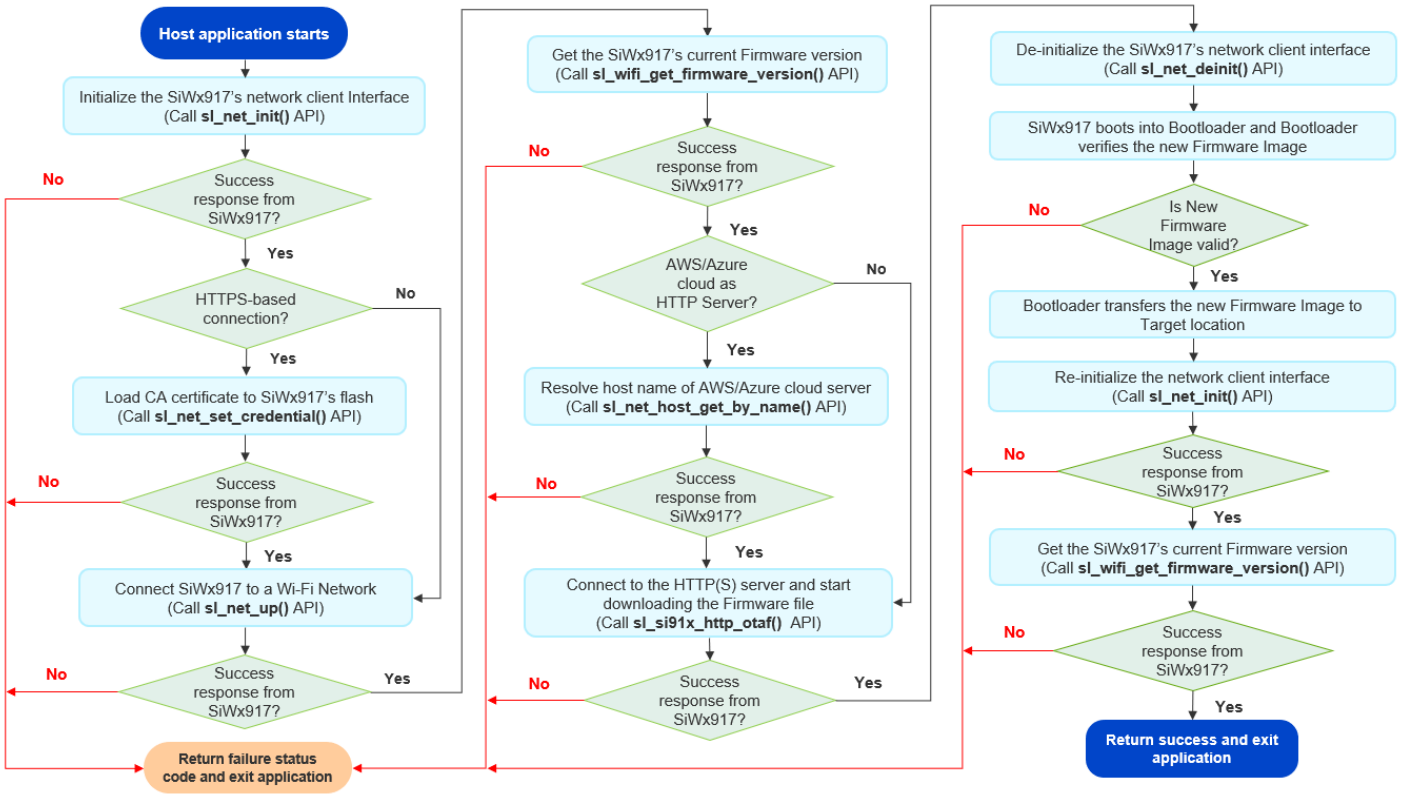


Figure 4.2. Firmware Update via HTTP/HTTPS Server Application Flowchart

Note:

1. When downloading the Firmware file via HTTP over TLS/SSL, the server (in this case AWS cloud/Azure cloud) might send 16K TLS records. Hence, it is highly recommended to enable SL_SI91X_EXT_TCP_IP_SSL_16K_RECORD feature in sl_wifi_device_configuration_t. sl_si91x_boot_configuration_t. ext_tcp_ip_feature_bit_map while calling sl_net_init() API.
2. For firmware update via HTTPS, the SL_SI91X_EXT_TCP_IP_FEAT_SSL_HIGH_PERFORMANCE feature should be enabled in sl_wifi_device_configuration_t.sl_si91x_boot_configuration_t.ext_tcp_ip_feature_bit_map while calling sl_net_init() API.
3. If you encounter SL_STATUS_SI91X_SSL_TLS_HANDSHAKE_FAIL (0x100D2) error while connecting to the Cloud Servers, even though certificates and credentials required to connect to the server securely are correctly provided, enable SL_SI91X_EXT_TCP_IP_FEAT_SSL_MEMORY_CLOUD feature in sl_wifi_device_configuration_t. sl_si91x_boot_configuration_t. ext_tcp_ip_feature_bit_map while calling sl_net_init() API.

5. Revision History

Revision 1.1

August, 2024

- Updated Section 1.2 Firmware Versioning.
- Update Section 2. When to Update Firmware?
- Updated Section 4.1.1 Using Simplicity Commander as Host (Replaced "Tera Term" with "Simplicity Studio").
- Added Section 4.1.2 Using a Host MCU Application.
- Initial Release

Revision 1.0

August, 2024

- Initial Release

Smart. Connected. Energy-Friendly.

IoT Portfolio
www.silabs.com/products

Quality
www.silabs.com/quality

Support & Community
www.silabs.com/community

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

www.silabs.com