AN1431: SiWx917 SoC Firmware Update Application Note

This application note describes the firmware update procedure for the SiWG917. This document covers the firmware load and update process and mechanisms, as well as the importance of secure over-the-air updates with combined images.

KEY POINTS

- Block Diagram
- Firmware Load, Update, and Update Mechanisms
- Secure Firmware Update

Table of Contents

1. Introduction

The SiWG917 includes two processors: Silicon Labs' ThreadArch® (TA) and an ARM® Cortex® M4 Processor. All the networking and wireless stacks run on independent threads of the ThreadArch®.

In addition, in adherence to the Trusted Execution Environment architecture, the TA subsystem also acts as the secure processing domain and takes care of secure boot and secure firmware update, as well as provides access to security accelerators and secure peripherals through pre-defined APIs. The Cortex-M4 is dedicated to peripheral and application-related processing.

A firmware is the software that is embedded into a hardware device. It contains a set of commands that control the behavior of a network device. Whenever a new firmware version is available, it is recommended that users update their devices to the latest version. Complete details about the latest firmware will be available in the Release Notes (shared as part of release package), which will help users decide whether to update to the new firmware or not.

The firmware in the SiWG917 device can be updated using the following mechanisms:

1. **Firmware update via Over-The-Air (OTA)** - In this mechanism, the firmware in the device can be updated by the following methods:
   - **HTTP/S:** The firmware is updated by downloading the firmware file from a remote HTTP/S or cloud server via Wi-Fi. The firmware file is directly downloaded to TA flash location.
   - **M4 as Host:** Using host interfaces – SPI/UART/SDIO or a remote TCP server via Wi-Fi or BLE, the firmware is reaching in chunks to M4. The user can choose to send the firmware to TA Bootloader for upgrade or save in the external flash as per their requirements.
2. **Firmware update via Bootloader** - In this mechanism, the firmware in the device can be updated by the following methods.
   - **Kermit:** Firmware is updated using the Kermit protocol in a serial terminal like Tera Term running in a Windows/Linux PC The connected to the device through UART interface in ISP mode.
   - **Simplicity Commander Tool/ Command Line Interface (CLI):** Using the Simplicity Commander tool or by using the CLI commands, the firmware is updated.

Note: SiWG917 has both the Secure and Non-Secure Firmware update options. The above mechanisms are the same for both secure and non-secure options, except that the firmware does security-related integrity checks before loading the device with the new firmware.

2. Block Diagram

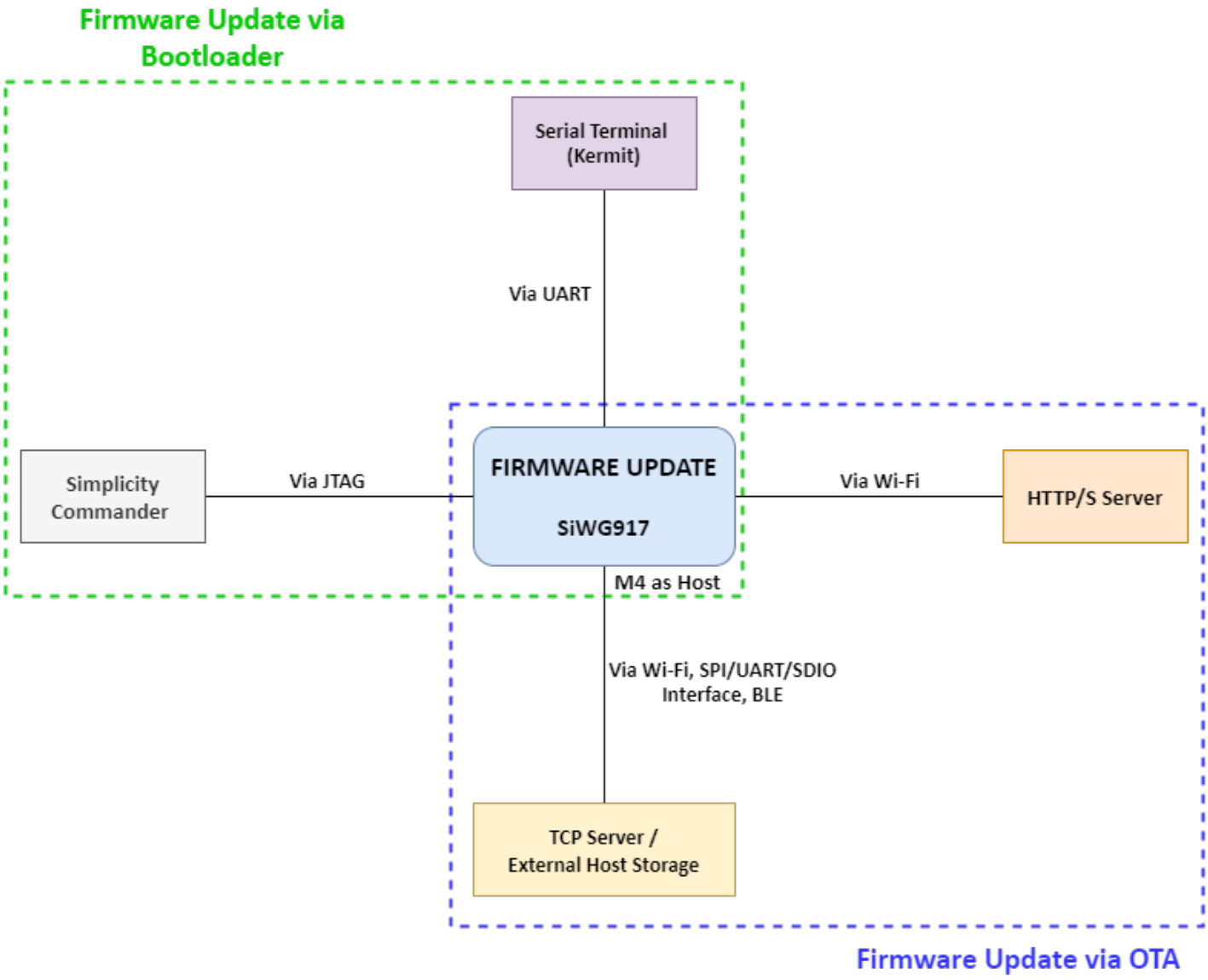

Figure 2.1. Firmware Update Mechanisms

Firmware File Format (RPS)

The TA Bootloader uses a proprietary format for its upgrade images, called RPS . These files have extension “. rps”. The RPS Format is a binary executable format understood by the Bootloader to perform the required integrity and authenticity checks and load and execute the application .

The Firmware Image in the RPS format includes an RPS header, boot descriptors, the application's binary image, and an optional trailer (digital signature).

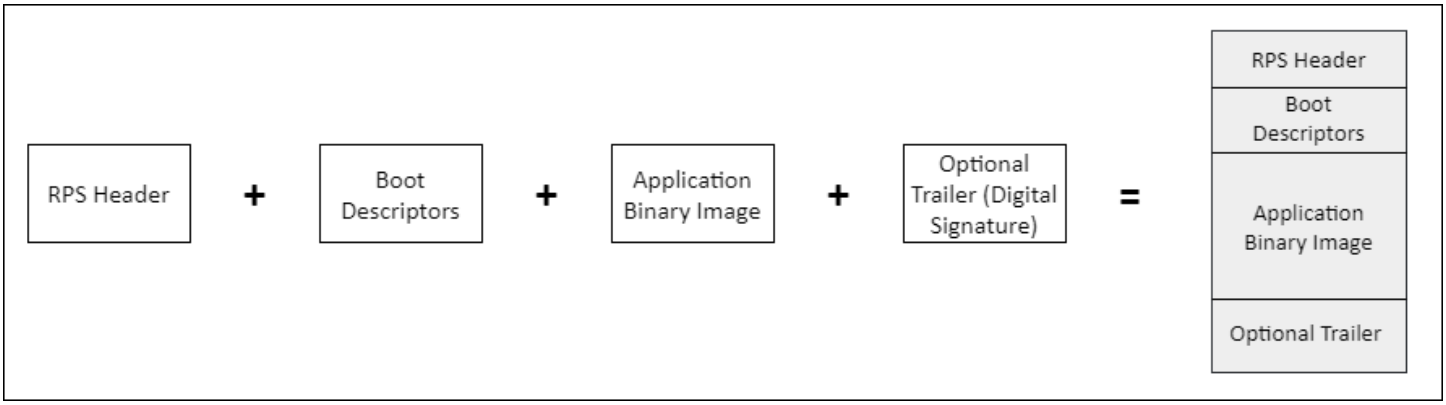

Figure 2.2. RPS Format

3. SiWG917 Firmware Load and Update Process

The following figures show the TA and M4 firmware load and update processes for the SiWG917 device. The firmware load process loads firmware onto the SiWG917 device for the first time in the case of new devices. Figure 3.1 on page 6 shows TA firmware load process. The Firmware update process updates the SiWG917 device with the latest firmware by replacing the firmware already existing in the device. Figure 3.2 on page 6 shows the TA firmware update process. Figure 3.3 on page 7 shows the M4 application update process.

The steps are as follows:

1. To update an existing firmware or a device without a firmware, download the new firmware file to the device's flash memory from the host (MCU/PC) through any host interface or through the OTA process.
2. After reboot, the current firmware is replaced by the new firmware in flash memory, and the device is updated with the new firmware.

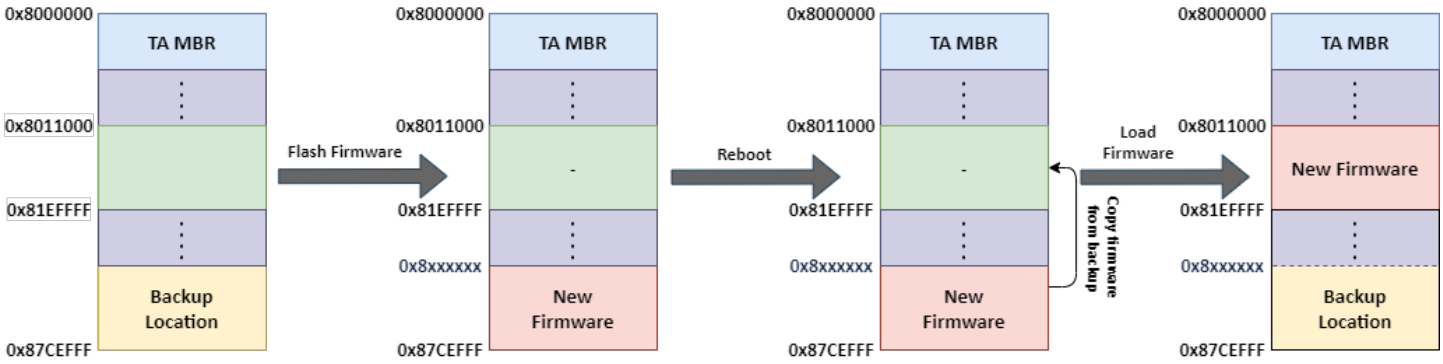

Figure 3.1. TA Firmware Load Process (on Empty Flash)

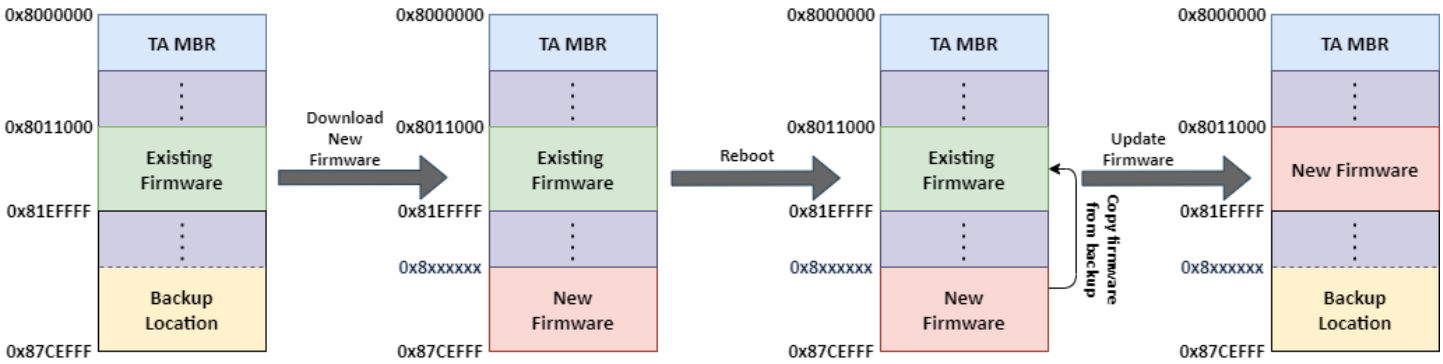

Figure 3.2. TA Firmware Update Process

A similar process is followed to load the M4 application, the below figure shows the M4 application loading process.

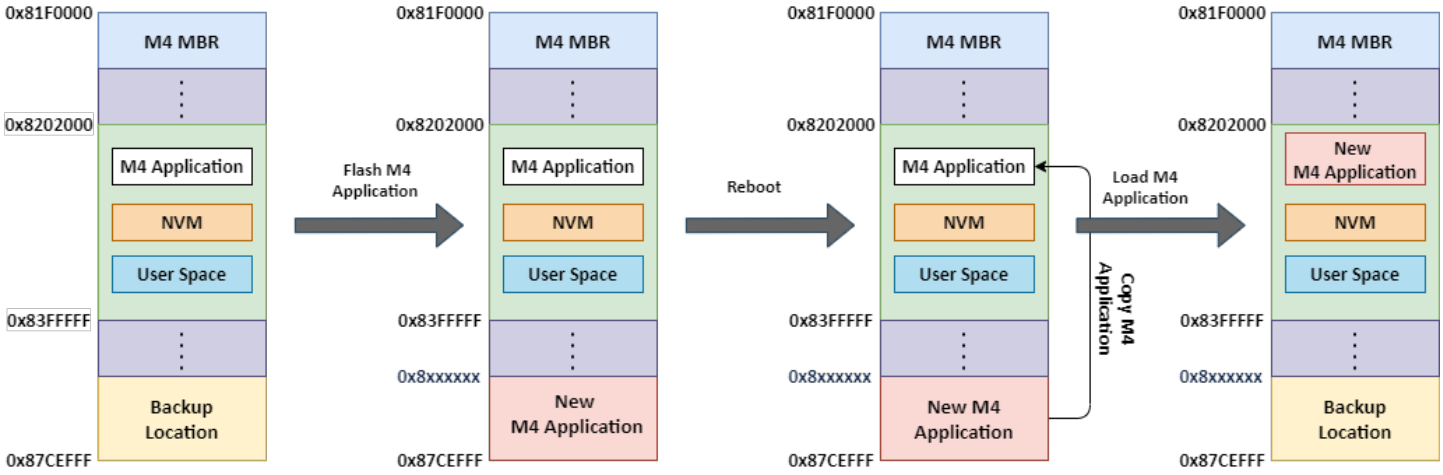

Figure 3.3. M4 Application Update Process

Note:

- The memory addresses shown in the above diagrams are only for reference purposes.
- User don't have control over determining the start address of the backup location.
- The 196 KB memory space after the backup location is used for Certificates (40KB), Reserved (140KB) and Bootloader Data Back-up & Store Config (16KB).
- The TA Bootloader is located in the Read Only memory (ROM), it cannot be modified by the user.

4. SiWG917 Firmware Update Mechanisms

Firmware update mechanisms are categorized into two:

1. Firmware Update via OTA
2. Firmware Update via Bootloader

The following subsections provide descriptions for both.

4.1 Firmware Update via OTA

The following steps are executed during Firmware Update Over The Air:

1. The current application receives and saves the new firmware image (RPS) in backup location.
2. The existing firmware does the integrity check of the new firmware based on the RPS header configuration.
3. If the integrity is verified and is valid, the existing firmware does a soft reset.
4. The device boots into the Bootloader.
5. The Bootloader finds that a new firmware is available in the download location and again does an integrity check using CRC, MIC, or Signature-based check (If security is enabled), depending on the control flags of the RPS and the MBR configuration.
6. After the image is verified, the Bootloader transfers the image from download location to the execution/target location. If the image is encrypted, decryption is performed during the process of transferring and parsing the RPS file.

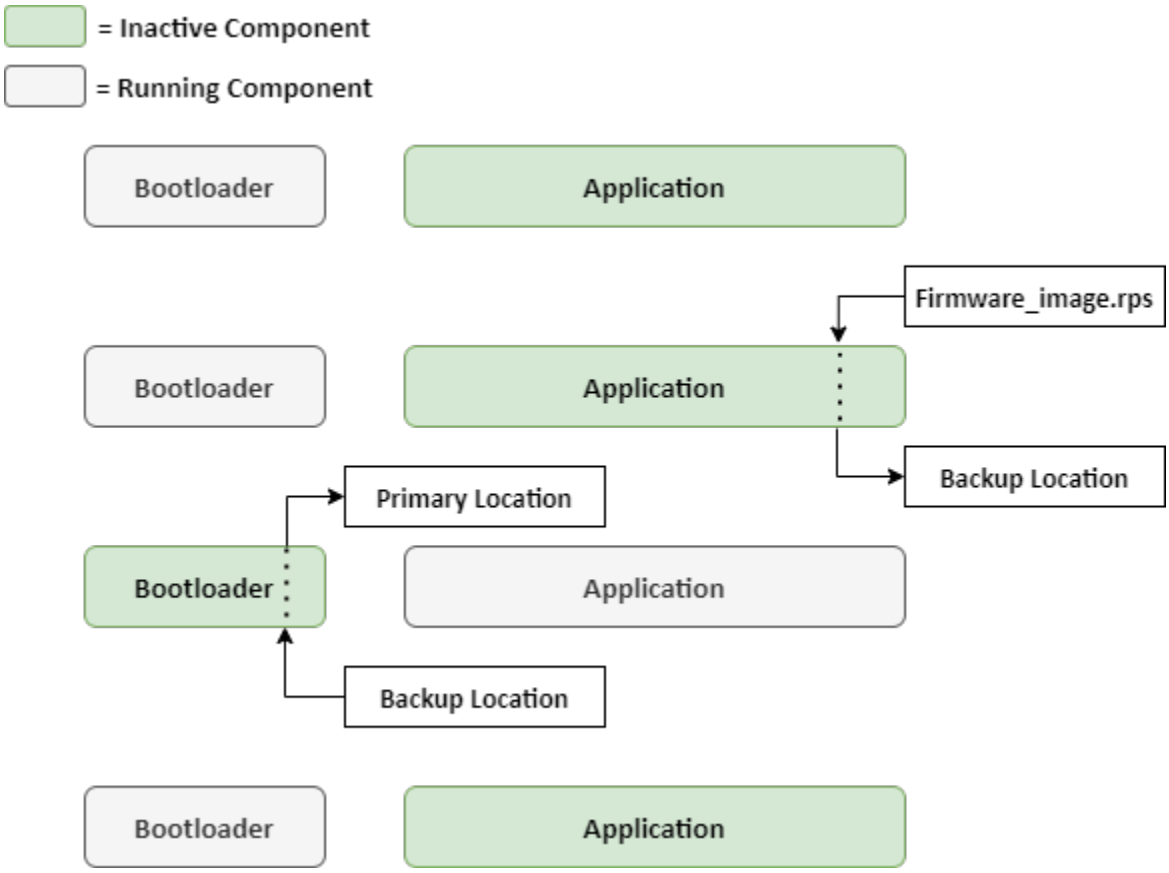

Figure 4.1. Firmware Update via OTA

There are two mechanisms to update Firmware via OTA.

1. Firmware Update via HTTP/S
2. Firmware Update via M4 as Host

The following subsections explains the mechanisms in detail.

4.1.1 Firmware Update via HTTP/S

The SiWG917 connects to the Access Point as an HTTP/S client and establishes connection with an HTTP/S server or the cloud storage server. After a successful HTTP/S connection, the SiWG917 sends a firmware file request to the remote server and the server responds with the firmware file.

The server-transferred firmware file gets loaded/updated in the SiWx91x module flash memory when the respective APIs are called. Figure 4.2 on page 9 and Figure 4.3 on page 10 shows the firmware update process in the form of an image and flow chart respectively.

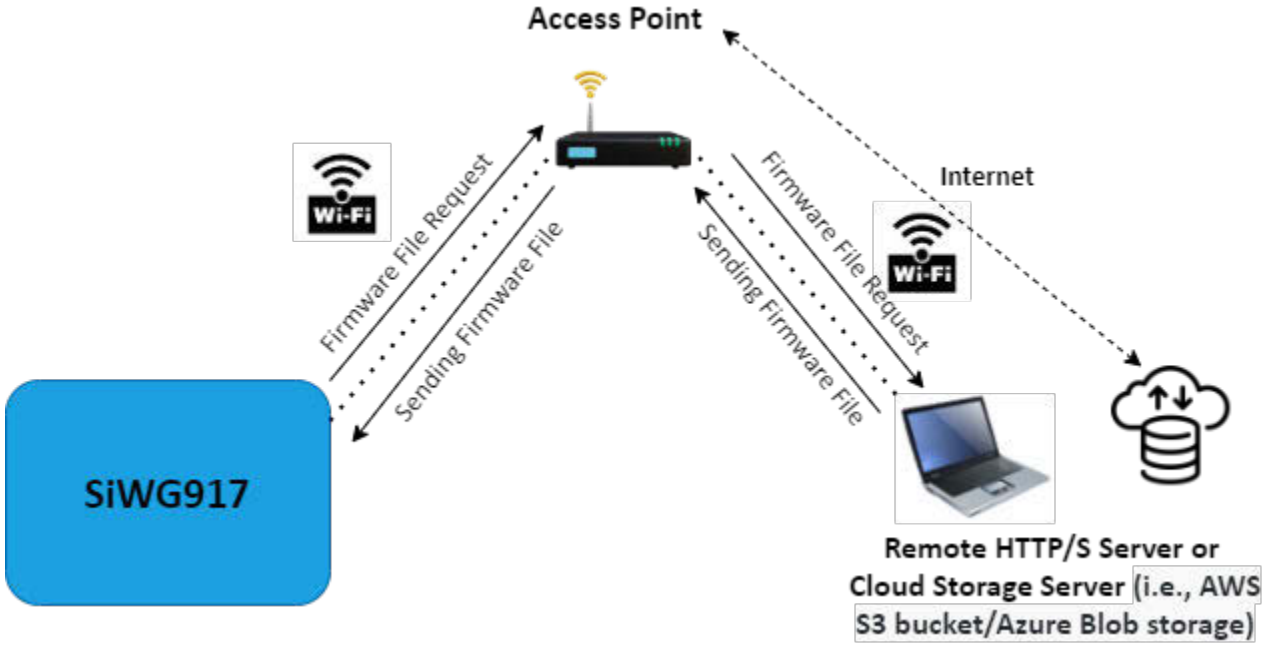

Figure 4.2. Firmware Update via HTTPS (OTA)

Firmware Update APIs

The following APIs are used for downloading and updating the TA firmware and M4 application.

Download Firmware:

- ***sl_si91x_http_otaf()*** - the API used for downloading is the same for TA and M4

Update Firmware:

- For TA Firmware – First ***sl_net_deinit()*** and next, ***sl_net_init()*** should be called
- For M4 Application – ***sl_si91x_soc_soft_reset()***

The following flowchart shows the steps involved in the firmware update process via HTTPS.

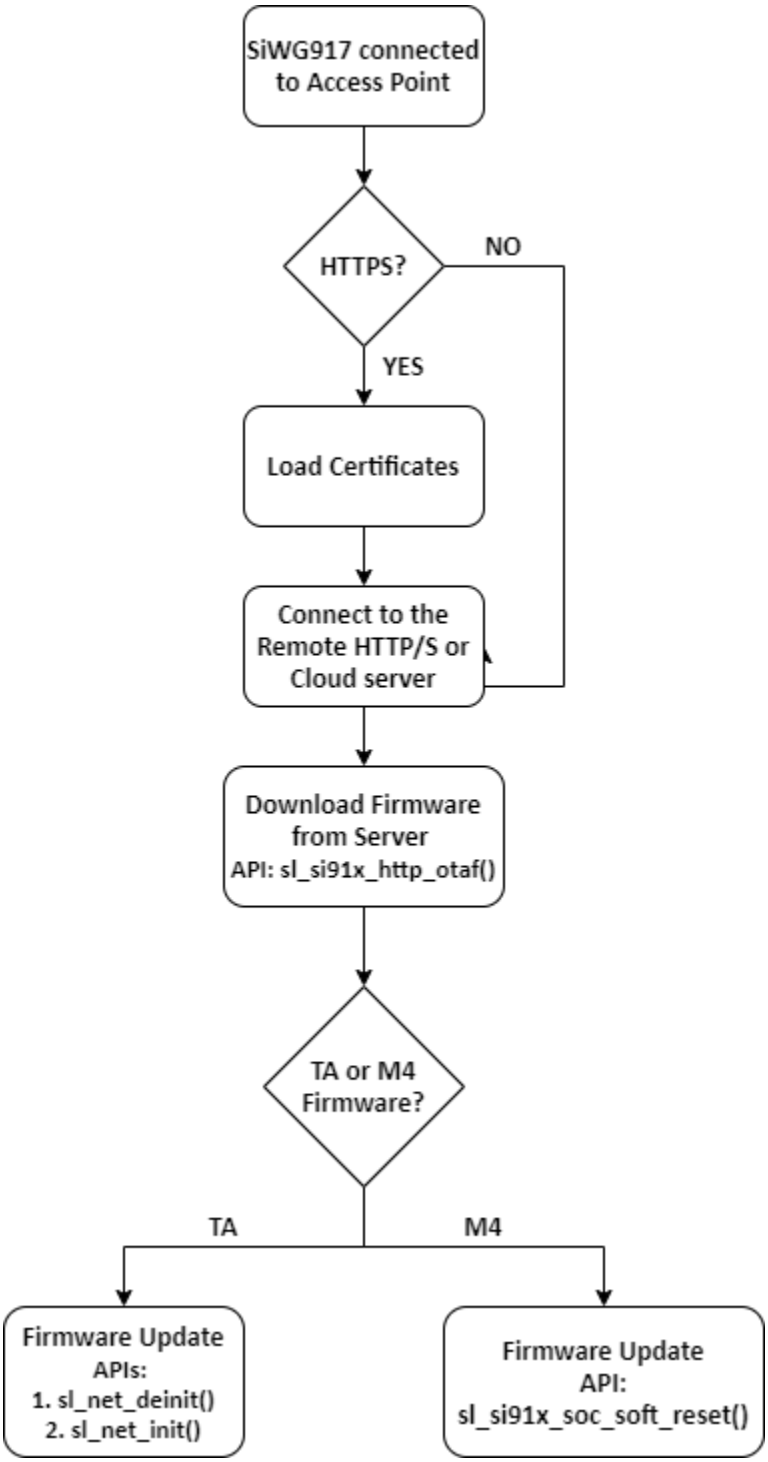

Figure 4.3. Firmware Update via HTTPS (OTA) – Flowchart

4.1.2 Firmware Update via M4 as Host

The SiWG917 connects to a remote TCP server via Wi-Fi or to an external host storage via SPI/UART/SDIO, or BLE to get the firmware file to the M4. The user can update the firmware by calling the required APIs.

Figure 4.4 on page 11 and Figure 4.5 on page 11 shows the firmware update process via M4 as host through an image and a flowchart respectively.

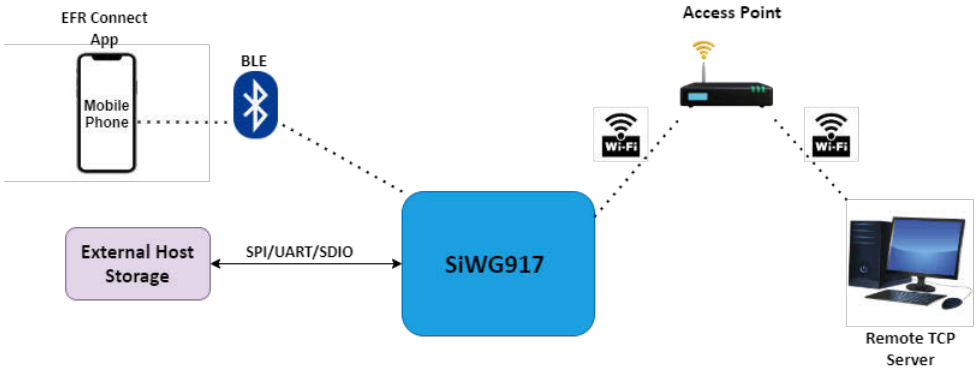

Figure 4.4. Firmware Update via M4 as Host (OTA)

The following flowchart shows the steps involved in the firmware update process via M4 as host:

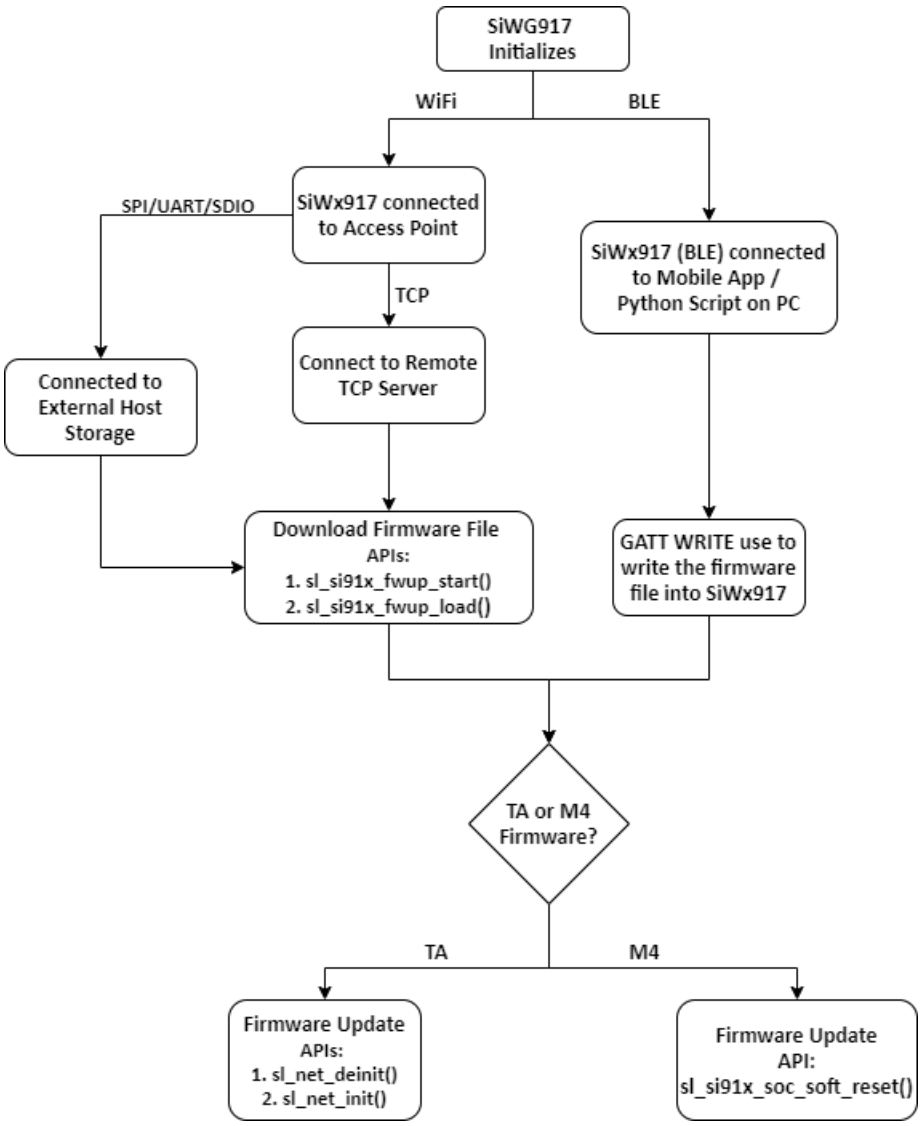

Figure 4.5. Firmware Update via M4 as Host (OTA) – Flowchart

4.2 Firmware Update via Bootloader

The Firmware Update via Bootloader is a two-stage process where the TA Bootloader places the application image in a separate download location and then does the integrity check of the received image. The TA Bootloader then replaces the current application image with the newly received authenticated image.

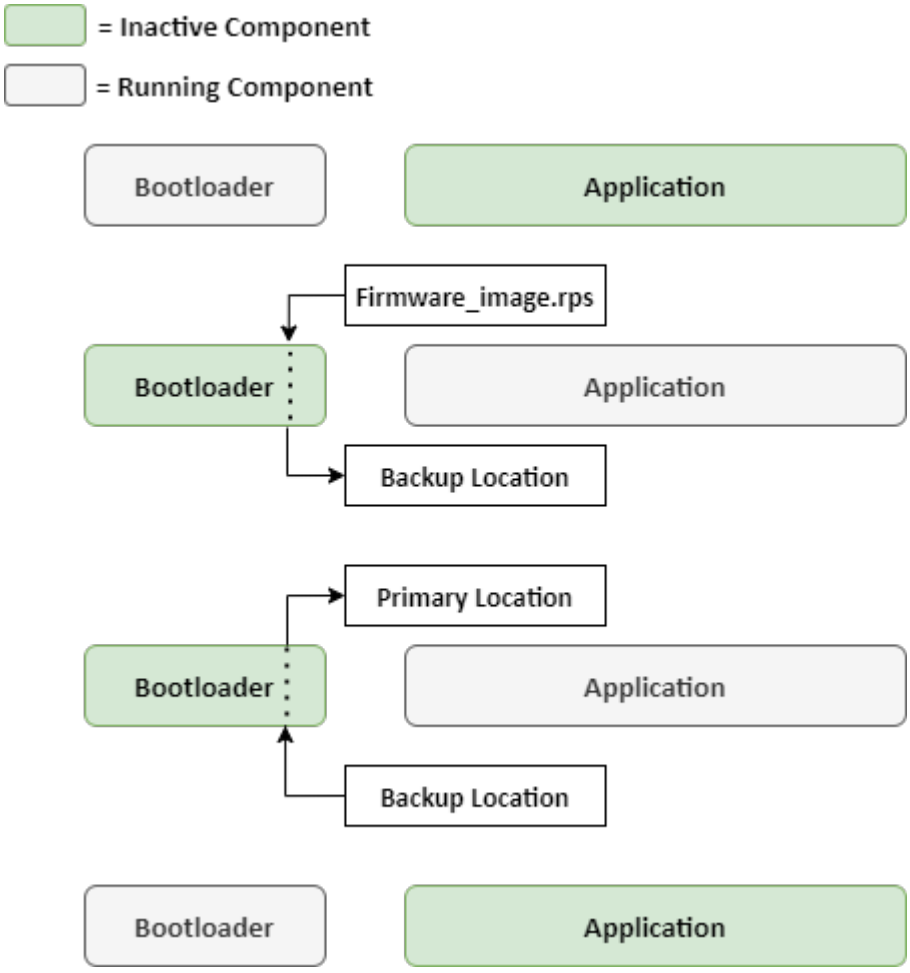

Figure 4.6. Firmware Update via Bootloader

There are two mechanisms to update the Firmware via Bootloader:

1. Firmware Update via Kermit
2. Firmware Update via Simplicity Commander Tool or CLI

The following subsections explains the mechanisms in detail.

4.2.1 Firmware Update via Kermit

In updating the firmware via Kermit, the SiWG917 device is connected to the PC using the USB cable. The device is detected as a UART COM Port. In this process, a serial terminal like Tera Term acts like a host to the SiWG917 and gives Bootloader commands to load the firmware.

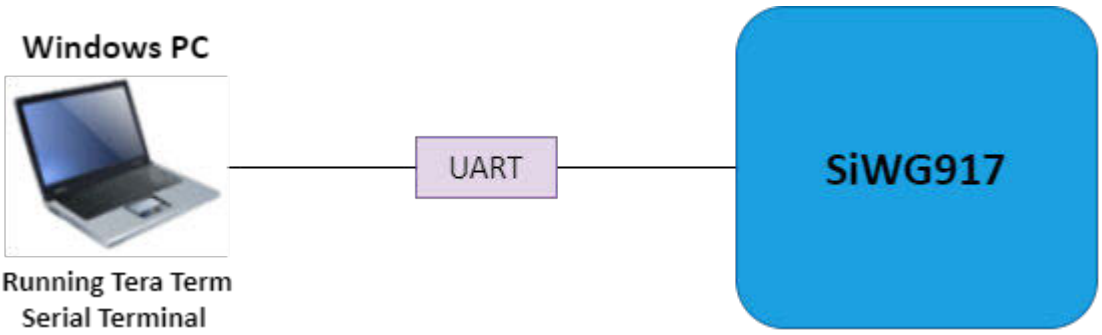

Figure 4.7. Firmware Update via Kermit

Steps for TA Firmware/M4 Application Update

1. Connect the RX pin of the TTL board (FTDI) to the F9 pin on the main board and the TX pin of the TTL board (FTDI) to the F8 pin.
2. Press the ISP-button, and while holding down the ISP-button, press and release the Reset-button (on the main board) and release the ISP-button, this will enable ISP mode.
   - You can exit the ISP mode by pressing the reset button on the main board.
3. Open Tera Term and choose COM Serial Port as shown below.

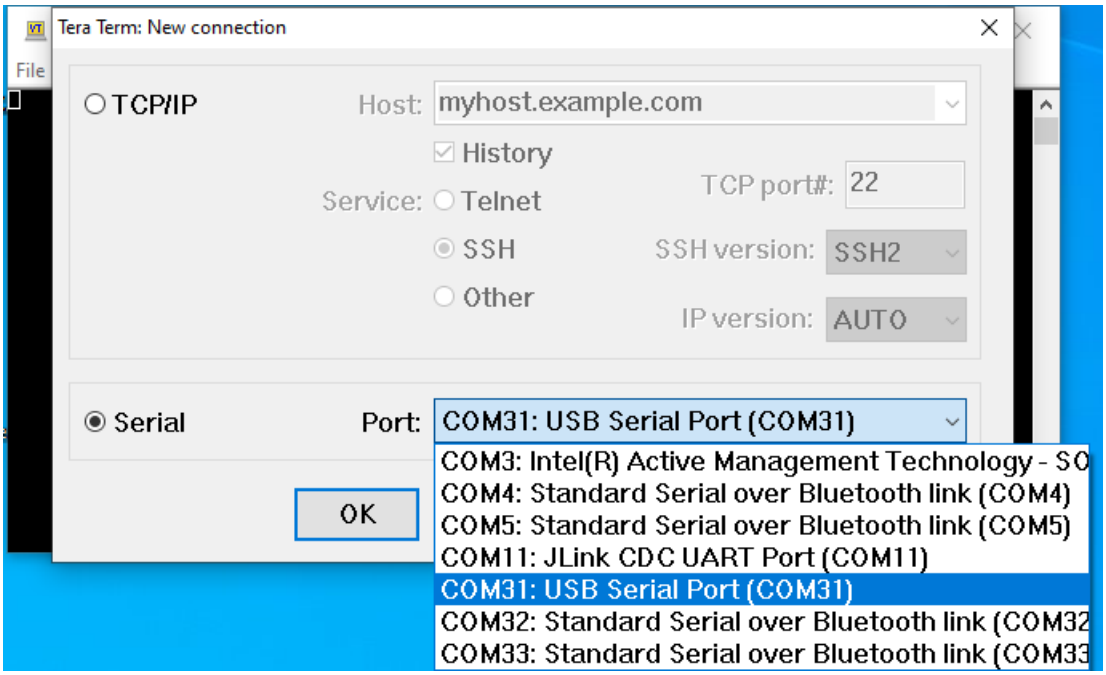

Figure 4.8. Serial COM Port Selection in Tera Term

4. Go to Setup → Serial port and select speed as **115200.**

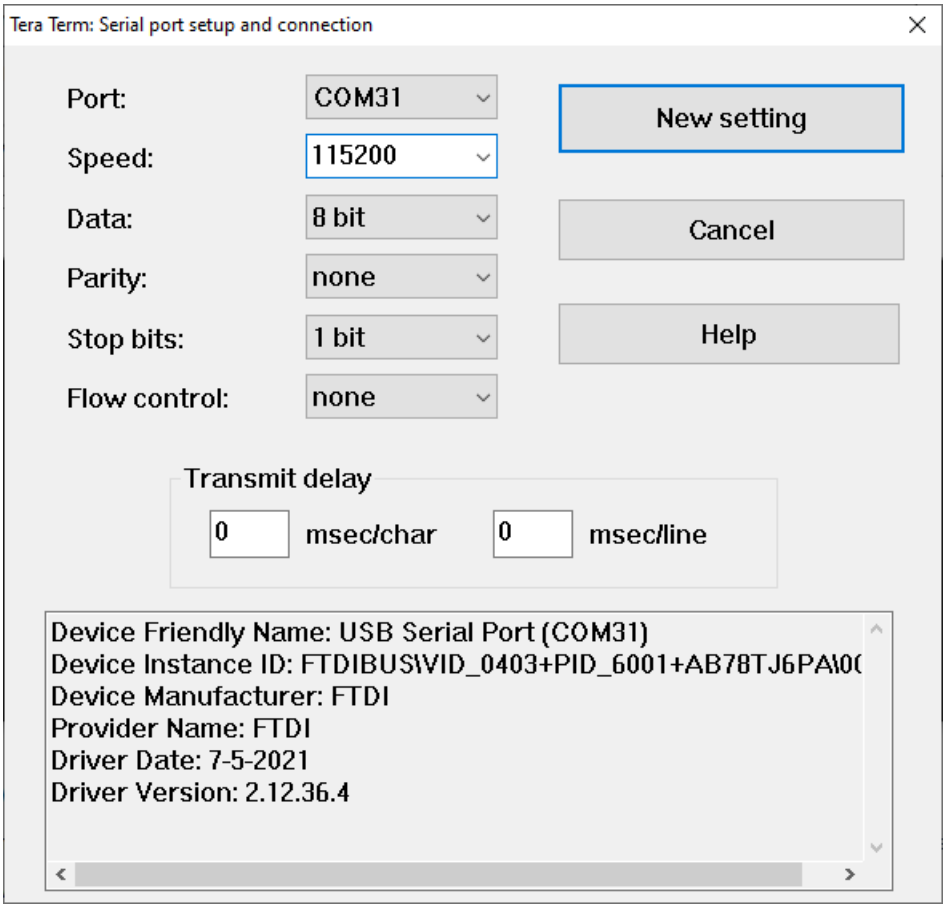

Figure 4.9. Serial Port Speed Selection in Tera Term

5. Press any key for boot message - "U", then type "U" for boot menu → type "B" → "0" to burn wireless (TA firmware) (or) "4", "1" to burn MCU (M4) application.

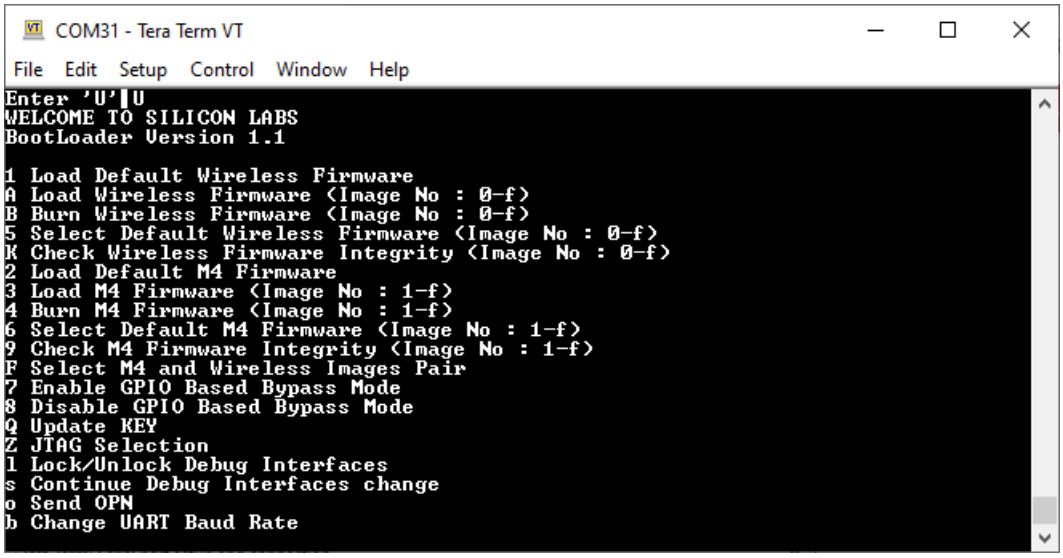

Figure 4.10. SiWG917 Boot Up

6. Go to File → Transfer → Kermit → Send.. and select '**.rps**' or '**.bin**' file, this starts to load the image into the SiWG917 device.

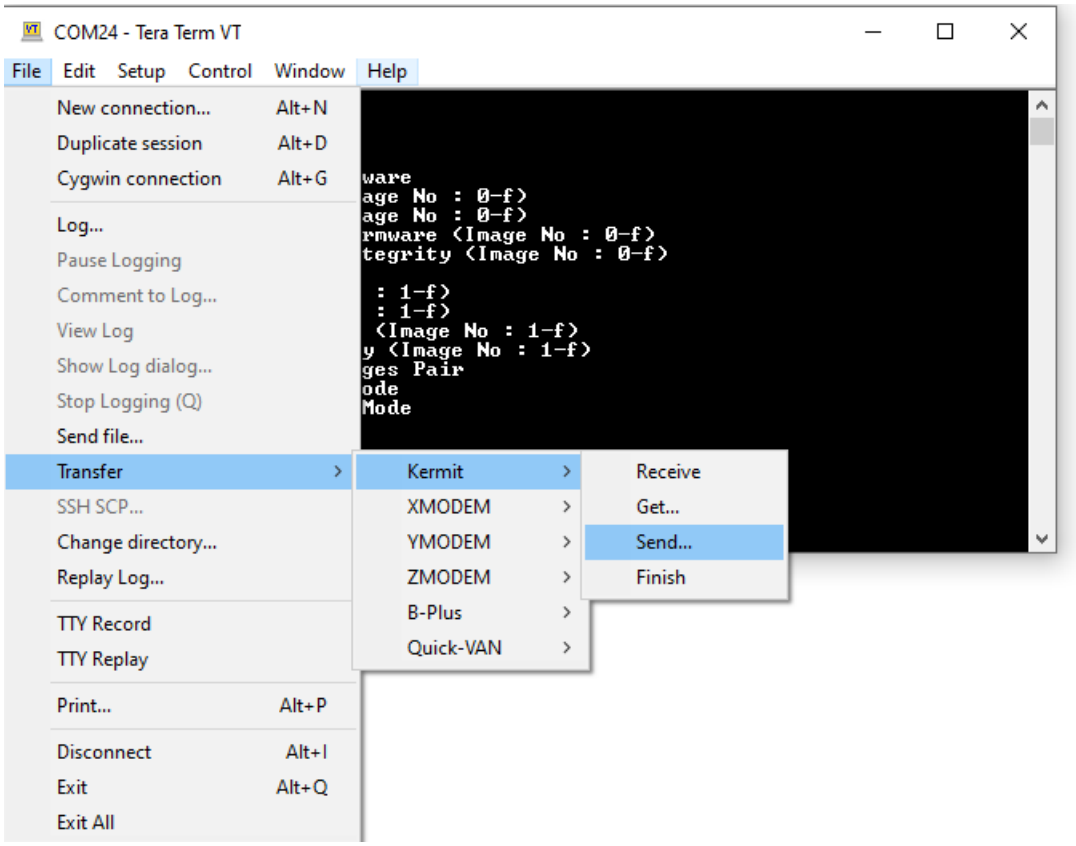

Figure 4.11. Send Firmware File in Tera Term via Kermit

7. Once the upgrade is successful, give "1", the message output "Loading..." will be displayed. It is now safe to exit from Tera Term.

4.2.2 Firmware Update via Simplicity Commander Tool or CLI

Simplicity Commander is a single, all-purpose tool to be used in a production environment. The steps to update the firmware using the Simplicity Commander tool are mentioned in the Upgrade SiWx91x Connectivity Firmware section in the Getting started with SoC mode. It can also be invoked using a simple Command Line Interface (CLI).

The following figure shows the steps involved in firmware update process using the Simplicity Commander.

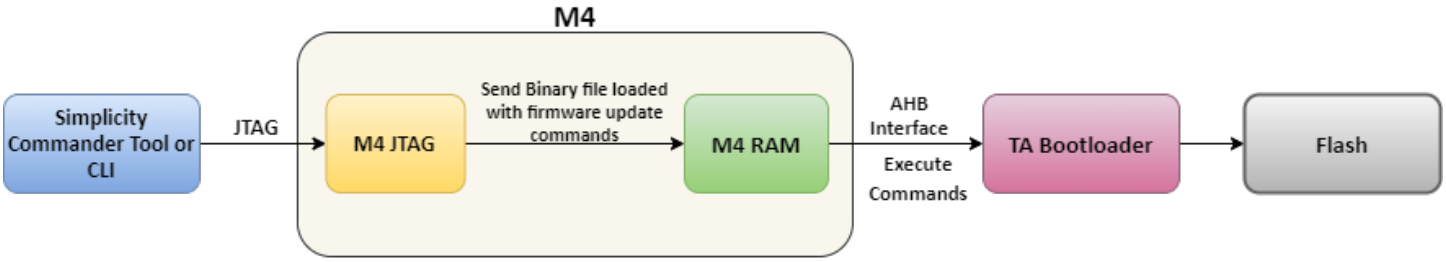

Figure 4.12. Firmware Update via Simplicity Commander

The general command line structure looks like this: **commander [command] [options] [arguments]** where:

- **commander** is the name of the tool
- **command** is one of the commands supported by Simplicity Commander, such as, flash, write, convert, etc.
- **argument** is an item of information provided to Simplicity Commander when it is started. An argument is commonly used when the command takes one or more input files.
  - square brackets indicate optional parameters as in this **example**: commander write [filename(s)] [options]
  - angle brackets indicate required parameters as in this **example**: commander read --output <filename>

The user can open the CLI, by navigating to the path: $:\..\SimplicityStudio\v5\developer\adapter_packs\commander and type **cmd.** The Simplicity Commander CLI opens.

TA Firmware Update

- **Command**: commander rps load <filename.rps> -d si917
- **Example**: commander rps load ta_fw.rps -d si917

M4 Firmware Update

- **Command:** commander rps load <filename.rps> -d si917
- **Example**: commander rps load **Secured_sl_si91x_calendar.rps** -d **si917**

For more information about the Simplicity Commander Line Interface, refer to the Customer Manufacturing Tool document.

Note: If the security is enabled in your device, you will have to load the secured image; otherwise, the loading fails.

5. Secure Firmware Update

In case of Bootloader and OTA, a secure firmware update can be done by enabling the security in the firmware image using the Manufacturing Utility (Refer to the UG574: SiWx917 SoC Manufacturing Utility User Guide). The process of firmware update remains the same for Non-Secure and Secure Firmware updates, except that in the case of secure firmware image, the integrity checks are done.

The following flowchart shows the Secure Firmware Update process.

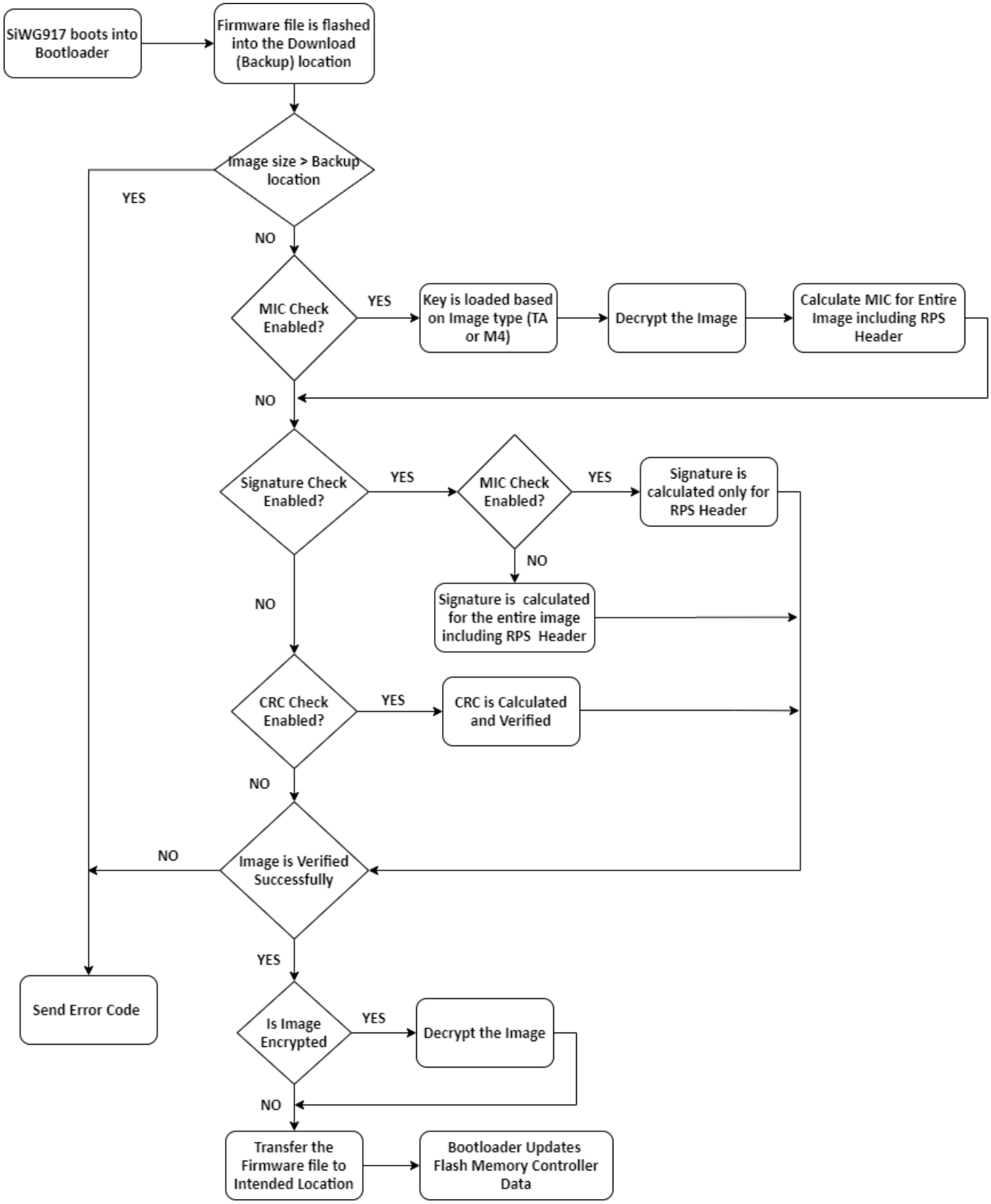

Figure 5.1. Secure Firmware Update Flowchart

The firmware image for both TA and M4 supports the following:

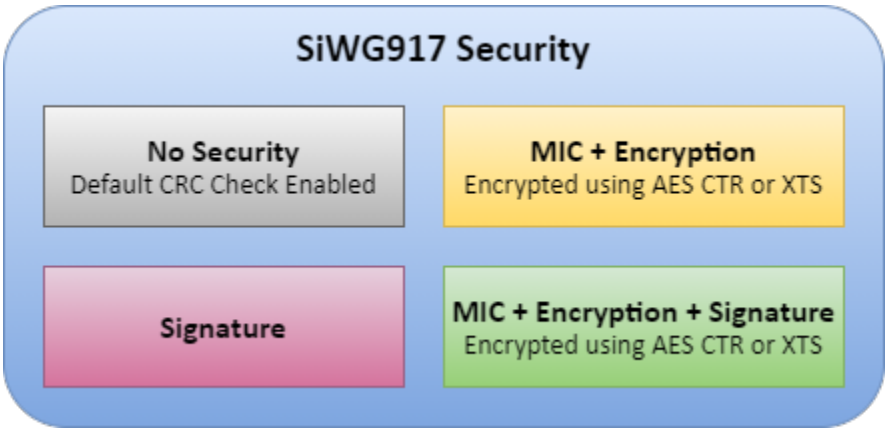

Figure 5.2. Supported Firmware Image Security

- No Security: Default CRC of the image within the RPS header.
- MIC + Encryption: The MIC is calculated for the whole plain image and saved within the RPS header. The image is encrypted and appended to the RPS header. The image can be encrypted using AES CTR or XTS mode.
- Signature: The size of the image in the RPS header is incremented by the size of the signature. Finally, the whole RPS file's (RPS Header+ image) signature is calculated and appended to the end of the file.
- MIC + Encryption + Signature: The MIC is calculated for the whole plain image and saved within the RPS header. The image is encrypted and appended to the RPS header. The image can be encrypted using AES CTR or XTS mode. The size of the image in the RPS header is incremented by the size of the signature. Finally, the whole RPS file's (RPS Header+ image) signature is calculated and appended to the end of the file.

In each mode, the Control Flag of the RPS Header is updated with the correct configuration.

Flash Memory Controller

The Bootloader has a reserved space in flash where it saves the details of the current firmware. These details are saved when a new valid firmware saved in the flash. This data includes the start address in the flash where the valid firmware is available, as well as the firmware's size, CRC and MIC value of the firmware, security flags, and the image's version number.

The FMC data itself is protected with CRC and MIC validation. Each time the FMC is modified after a new image is updated, the FMC is saved in a backup location and then the original location is updated to avoid any issues arising from the FMC getting corrupted while modifying it or a power down happening during the FMC modification process.

5.1 Combined Image

The combined image is a single image which is obtained by combining the TA and M4 images. The process of creating the combined image involves encrypting both the TA and M4 images, and then adding a RPS header and signature. In the case of non-secure firmware, we will not be adding the signature.

- The Combined Image RPS header format will be the same as the M4 RPS header format with few reserved bytes changes.
- The signature for complete combined image will be calculated and appends at the end of image.
- MIC computation and signature can be used to maintain integrity and confidentiality of the combined image.
- Encryption of combined image is discarded as it will add overhead for firmware to decrypt and store into flash location. The TA and M4 images are individually encrypted.

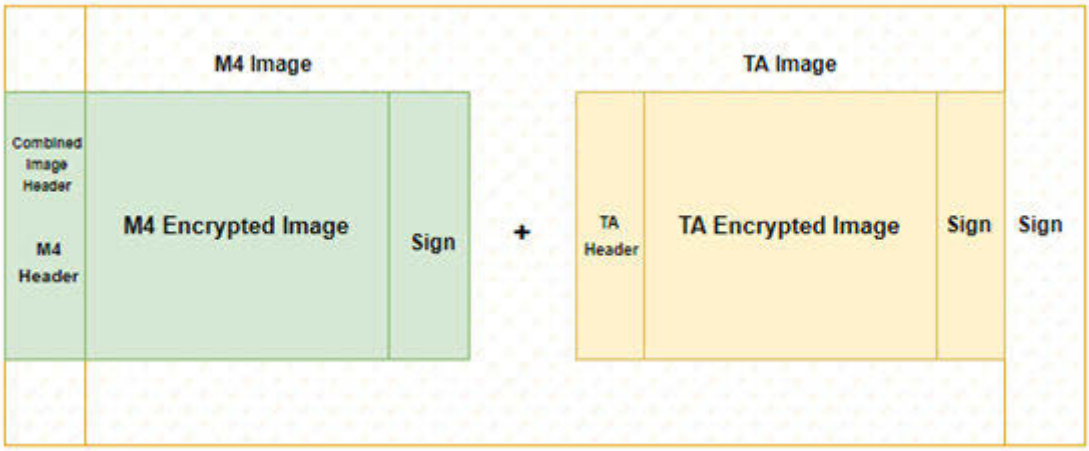

Figure 5.3. Combined Image with Signature

Note: For more information on combined image generation, refer to UG574: SiWx917 SoC Manufacturing Utility User Guide.

5.2 Importance of Secure Over-the-Air Updates with Combined Images

- Over The Air (OTA) updates have become a popular way to update device firmware, ensuring that devices have the latest features and security enhancements. However, the security of OTA updates is critical, as attackers can exploit vulnerabilities in the update process to gain unauthorized access to devices and data.
- To address this, combined firmware images are now used for OTA updates. These images contain multiple firmware components, simplifying the update process, and reducing the risk of errors or incompatibilities.
- To ensure security, digital code signing is used to verify the authenticity of the combined image, while encryption protects the image during transmission to prevent unauthorized access and tampering.
- These measures provide robust protection, ensuring that OTA updates are performed safely and securely, without compromising the confidentiality or integrity of the device.
- Overall, OTA updates using combined firmware images are important for device manufacturers and end-users, enabling convenient and secure device firmware updates.

6. Guidelines and Recommendations

1. It is strongly recommended to use the latest version of firmware, especially for all major releases, as this has many changes compared to minor releases.
2. The SiWG917 should be kept powered on during the complete firmware update process, you need to ensure there are no power fluctuations.
3. While firmware update is happening, it is advisable to avoid doing multiple tasks.
4. The SoC should not be placed in power save mode during any portion of the firmware update process.
5. We recommend using the simplicity commander tool for firmware update via Bootloader.
6. Anti-rollback check is enabled as part of efuse configs and is used to block downgrading the firmware. It is disabled by default, but the users would enable it in the end of their development.

7. References

The below are few firmware update example references. Click on the link of your preferred example.

- Firmware update using a TCP server on a local PC or a cloud service such as Amazon AWS or Microsoft Azure: GitHub Link
- Firmware update using remote HTTP/s server or cloud storage server: GitHub Link
- M4 Firmware Update using a TCP server on a local PC or a cloud service such as Amazon AWS or Microsoft Azure: GitHub Link

8. Troubleshooting

- On a fresh SiWG917 chip with no TA firmware present, if you try to run the application – You will see the error code: 0x56, this means Valid Firmware Not Present. Load the SiWG917 with the TA firmware and run the application to resolve this issue.
- Loading non-secure firmware on secure board and vice-versa, results in 108 error (CRC check failure)
    1. If board is having secure MBR, we'll add security to the firmware image with the same keys and load the firmware in the device.
- Loading 1.6 MB M4 application in device with 1.8 MB MBR, results in 102 error (ISP mode ON)
    1. Check the MBR present in the device using SiWG917 manufacturing utility.
    2. If 1B → Load M4 application having 1.6MB flash addresses
    3. If 1F → Load M4 application having 1.8MB flash addresses
    4. Anything else (CC, 00) → program MBR again

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