AN1417: MBR Application Note

Version 0.1

April 2023

Table of Contents

1 Introduction

The SiWx917SoC offers an option to the customers to mount recommended flash parts externally. For this option, the flash needs intelligence to understand the processor.

Master Boot Record (MBR), the intelligence for the flash to understand the processor. The MBR should be loaded into the flash and a utility is designed for the purpose.

This document describes the usage of the utility that writes the MBR to the flash.

Note: For the recommended flash options, please refer to the datasheet.

Available SoC Variants:

- Common flash (Shared between M4 and TA) - 4 MB/8 MB
- Dual flash (Dedicated flash for M4 and TA) - 4 MB/8 MB/16 MB
- External Flash (Shared between M4 and TA) - 4 MB/8 MB/16 MB

When the SoC is shipped out of the factory, it will have a Common flash MBR. MCU Flash can be mounted on pins 46-51 or 52-57.

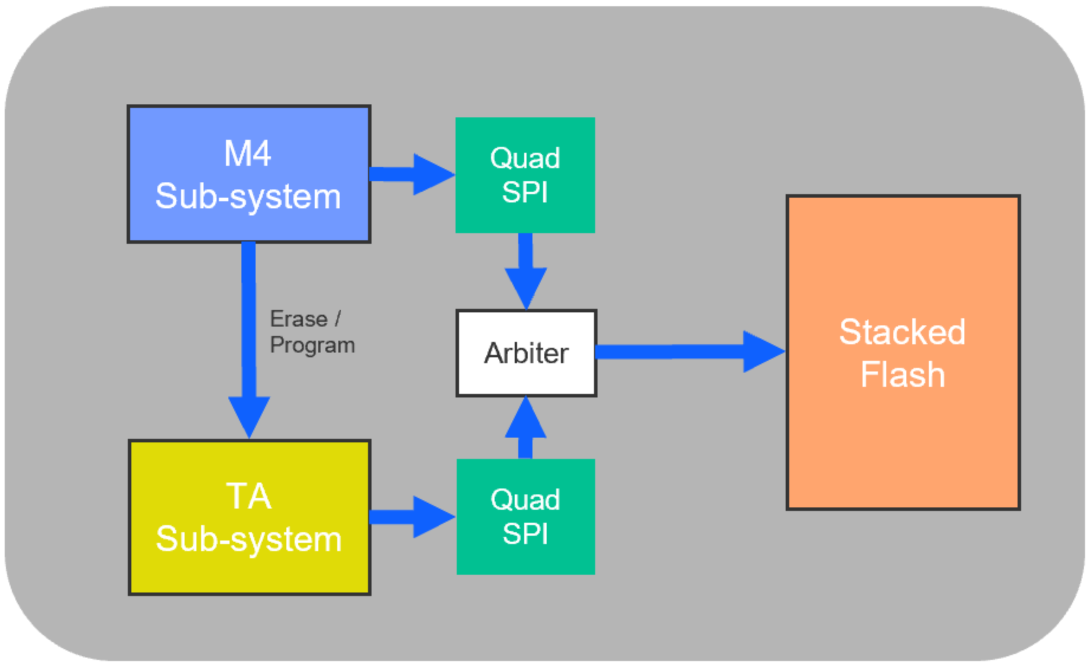


Figure 1.1: Common Flash Block Diagram

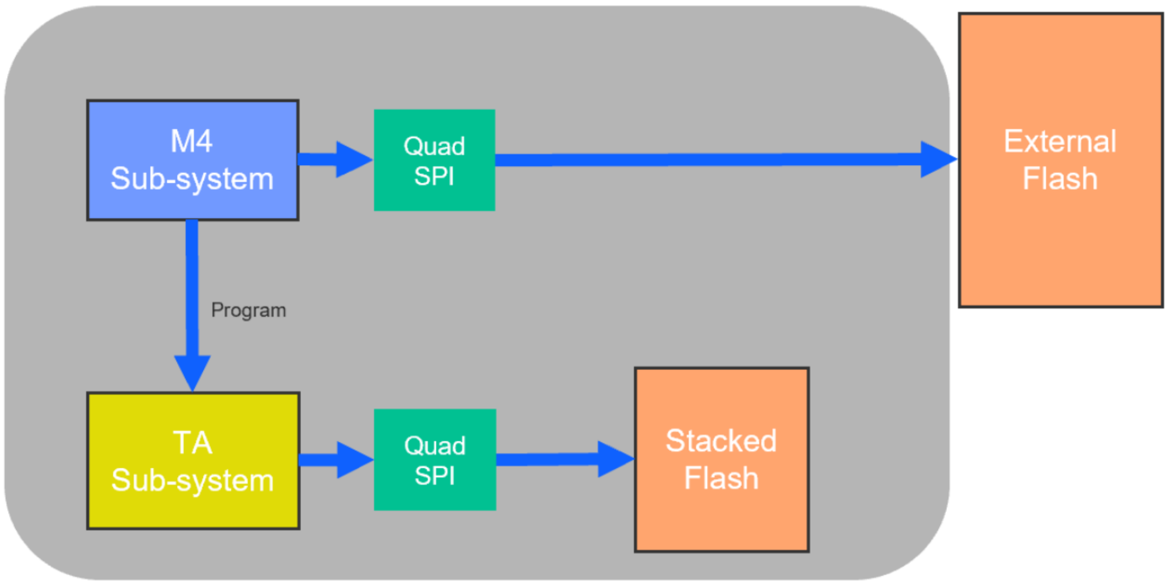


Figure 1.2: Dual Flash Block Diagram

2 The Utility

The utility is designed to flash the MBRs in all the available SiWx917SoC variants depending on the need. This utility has an intelligence to make sure that the calibration data is intact and is not affected.

It is advised to flash the MBRs only once and refrain from re-flashing. Every variant has a unique MBR and will be provided for by Silicon Labs.

- **Common Flash:** The utility can rewrite the common flash MBR if needed.
- **Dual Flash:** When there is an external flash, this utility writes the TA MBR in the stacked flash and M4 MBR in the external flash. On boot-up, the bootloader reads the TA MBR and gets the information on where to switch next along with the M4 application start address. The utility would also need to configure on which pin the external flash has been mounted. TA MBR must be written with the respective pin set and thereafter writes the M4 MBR.
- **External Flash Only:** There is one more option that may be available where a complete external flash would be used. This would be common between the TA and M4.

  In this case, the utility would be used to configure the pin of the external flash, copy the calibration data to the external flash, and write the MBR.

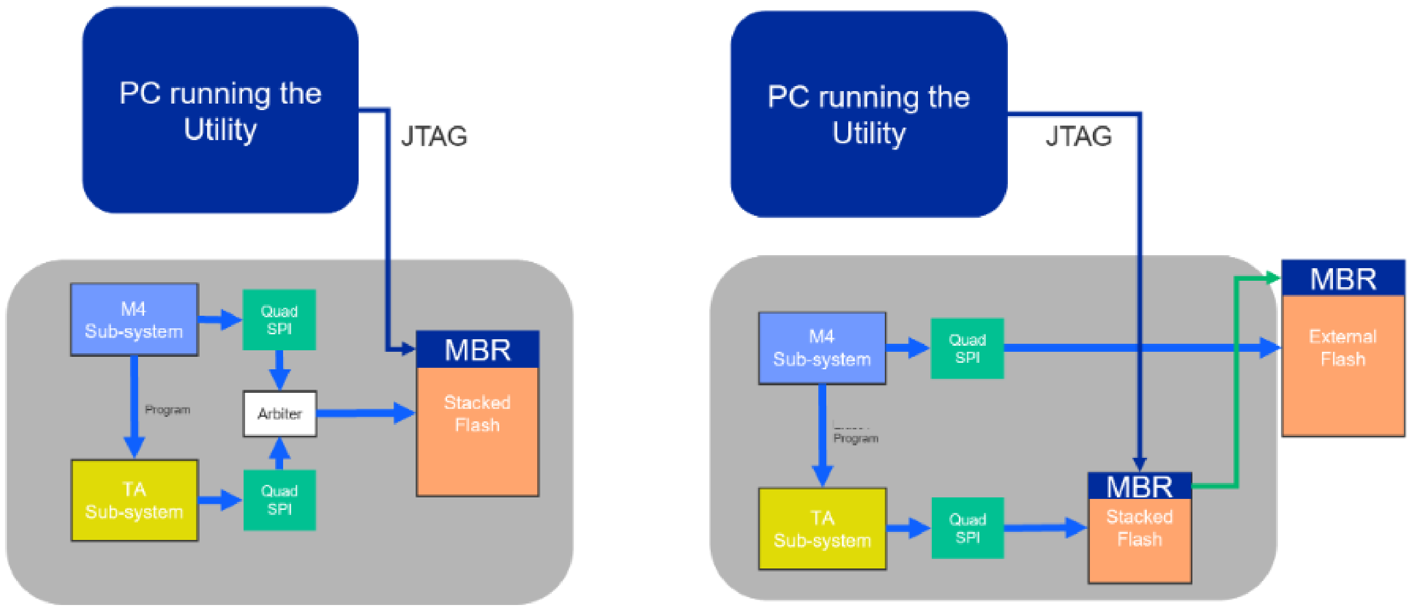


Figure 2.3: External Common Flash Block Diagram

3 Programming the MBR

An application (Utility 0.1) is developed that flashes the MBR in the SiWx917SoC module. This application runs from RAM and writes the MBR to the respective locations.

3.1 Prerequisites

- SEGGER J-Link Commander tool running in a Windows PC
- The respective MBRs
- The application binary, which flashes the MBRs.

3.2 Binary Files

Please contact Silicon Labs support for the latest Binary files.

Type of MBR	Comments
Common Flash MBR	A single binary file will be provided. Running this bin once will flash both TA and M4 MBR.
Dual Flash MBR (External Flash pin set 52 to 57)	Two binary files will be available. One for TA and One for M4. The names of the files will be self-explanatory.
Dual Flash MBR (External Flash pin set 46 to 51)	Two binary files will be available. One for TA and One for M4. The names of the files will be self-explanatory.
Common External flash only option (External flash pin set 46 to 51)	Single Binary file will be available for converting Primary flash to external flash.

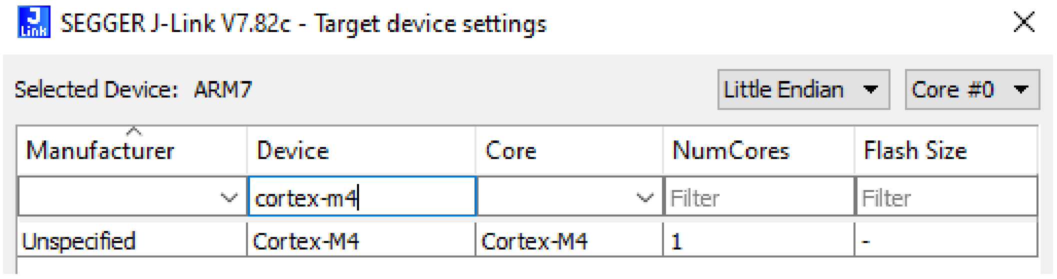
Table 3.1: Binary Files

Note:

We have plans to support XMC and Macronix flashes in the future. If you are planning to use a different flash configuration, get in touch with Silicon Labs to get the schematics as well as part number support reviewed.

3.3 Flashing Procedure

Get the Hardware and Software ready.

1. Connect the board to the PC. Open the SEGGER J-Link Commander software.
   a. Type **"connect"** and press Enter key**.**
   b. Choose all the Default options. If the device/ core is **"Si917", it** asks for device selection. Select **"Cortex-M4"**
   c. 

SEGGER J-Link V7.82c - Target device settings

Selected Device: ARM7

Manufacturer	Device	Core	NumCores	Flash Size
	cortex-m4		Filter	Filter
Unspecified	Cortex-M4	Cortex-M4	1	-

2. Relaunch the J-Link commander, Power cycle the board, and "**connect**" again. Select all the Default options. Now, the device name will be displayed as "**Cortex-M4**"

3.3.1 Flashing the Common Flash MBR

1. Load the given bin file in the address 0x00. "**O.K"** print will be displayed on successful loading. If the loading fails, repeat this step.
   - **Syntax:-** loadfile <path\filename> <Addr>
     - Example:- loadfile C:\Users\chswaroo\Downloads\Common_flash_MBR.bin 0x00
2. Verify that the bin file loaded. A **"Verify successful"** print will be displayed on successful verification. If you see a verification failure, repeat this step.
   - **Syntax:-** verifybin <path\filename> <Addr>
     - Example:- verifybin C:\Users\chswaroo\Downloads\Common_flash_MBR.bin 0x00
3. Issue the following commands in sequence:
   a. wreg "R13 (SP)" 0x00020c00
   b. wreg "R15 (PC)" 0x00000249
   c. w4 0xE000ED08 0x00
   d. go
4. Read the 0x400000 location.
   - mem32 0x400000 1
5. Read the 0x22000400 location. It should contain **something like "00001xxx".** If you do not find the values mentioned, issue the "**reset**" command and repeat the steps from **step 3**.
   - mem32 0x22000400,1

Note:

- After step 1 and step 4, a 10-second timeout is recommended.
- The SP and PC values in step 5 are specific to these binary files. It may change if you have any other bin.
- After the above steps are done, reset the board, create a new common flash "Blinky SoC" example, and flash it.

Sample Log for Common Flash:

```
J-Link>loadfile C:\Users\chswaroo\Downloads\Moen.bin 0x00
'loadfile': Performing implicit reset & halt of MCU.
Reset: Halt core after reset via DEMCR.VC_CORERESET.
Reset: Reset device via AIRCR.SYSRESETREQ.
Downloading file [C:\Users\chswaroo\Downloads\Moen.bin]...
O.K.

J-Link>VerifyBin C:\Users\chswaroo\Downloads\Moen.bin 0x00
Loading binary file C:\Users\chswaroo\Downloads\Moen.bin
Reading 37780 bytes data from target memory @ 0x00000000.
Verify successful.

J-Link>wreg "R13 (SP)" 0x00020c00
R13 (SP) = 0x00020C00
J-Link>wreg "R15 (PC)" 0x00000249
R15 (PC) = 0x00000249
J-Link>w4 0xE000ED08 0x00
Writing 00000000 -> E000ED08
J-Link>go

J-Link>mem32 400000 1
00400000 = 00087C9C

J-Link>mem32 0x22000400,1
22000400 = 000010F0
```

3.3.2 Flashing the Dual Flash MBR

First**, l**oad the TA MBR file:

Load TA MBR:

1. Load the given bin file in the address 0x00. "**O.K"** print will be displayed on successful loading. If the loading fails, repeat this step.
   - **Syntax:-** loadfile <path\filename> <Addr>
     - Example: loadfile C:\917CCP\cases\Moen\DualFlashMBR\TA_mbr.bin 0x00
2. Verify that the bin file loaded. A **"Verify successful"** print will be displayed on successful verification. If you see a verification failure, repeat this step
   - **Syntax:-** verifybin <path\filename> <Addr>
     - Example: VerifyBin C:\917CCP\cases\Moen\DualFlashMBR\TA_mbr.bin 0x00
3. Issue the following commands in sequence:

   a. wreg "R13 (SP)" 0x00020c00
   b. wreg "R15 (PC)" 0x00000249
   c. w4 0xE000ED08 0x00
   d. go
4. Read the 0x400000 location.
   - mem32 0x400000 1
5. Read the 0x22000400 location. It should contain **"000010F0" or something like "000010xx".** If you do not find the values mentioned, issue the "**reset**" command and repeat the steps from **step 3**.
   - mem32 0x22000400,1

Next, **l**oad the M4 MBR:

Load M4 MBR:

1. **Syntax:-** loadfile <path\filename> <Addr>
   - Example: loadfile C:\917CCP\cases\Moen\DualFlashMBR\M4_mbr.bin 0x00
2. Verify that the bin file loaded. A **"Verify successful"** print will be displayed on successful verification. If you see a verification failure, repeat this step
   - **Syntax:-** verifybin <path\filename> <Addr>
     - Example: VerifyBin C:\917CCP\cases\Moen\DualFlashMBRM4_mbr.bin 0x00
3. Issue the following commands in sequence:
   a. wreg "R13 (SP)" 0x00020c00
   b. wreg "R15 (PC)" 0x00000249
   c. w4 0xE000ED08 0x00
   d. go
4. Read the 0x400000 location.
   - mem32 0x400000 1
5. Read the 0x22000400 location. It should contain **something like "000010xx".** If you do not find the values mentioned, issue the "**reset**" command and repeat the steps from **step 3**.
   - mem32 0x22000400,1

Note:

- After steps 1 and 4, a 10-second timeout is recommended.
- The SP and PC values in step 5 are specific to these binary files. It may change if you have any other bin.
- After the above steps are done, reset the board, create a new common flash "Blinky SoC" example, and flash it.

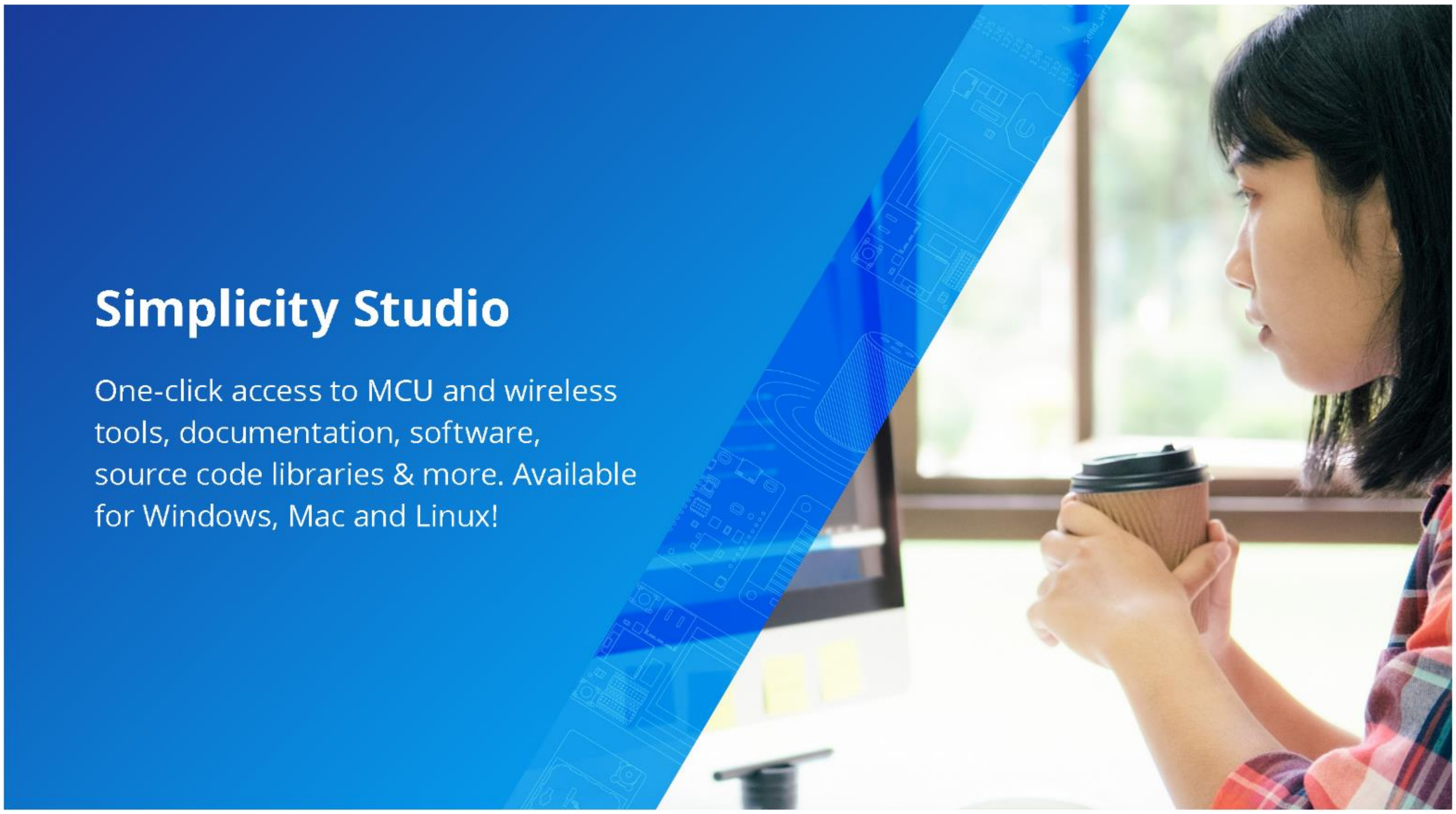
Sample Log for Dual Flash:

```
Just attaching my log for your reference.

J-Link>loadfile C:\917CCP\cases\Moen\DualFlashMBR\TA_mbr.bin
0x00
'loadfile': Performing implicit reset & halt of MCU.
Reset: Halt core after reset via DEMCR.VC_CORERESET.
Reset: Reset device via AIRCR.SYSRESETREQ.
Downloading file
[C:\917CCP\cases\Moen\DualFlashMBR\TA_mbr.bin]...
O.K.

J-
Link>VerifyBin  C:\917CCP\cases\Moen\DualFlashMBR\TA_mbr.bin
0x00
```

```
Loading binary file
C:\917CCP\cases\Moen\DualFlashMBR\TA_mbr.bin

Reading 37720 bytes data from target memory @ 0x00000000.
Verify successful.

J-Link>wreg "R13 (SP)" 0x00020c00
R13 (SP) = 0x00020C00
J-Link>wreg "R15 (PC)" 0x00000249
R15 (PC) = 0x00000249
J-Link>w4 0xE000ED08 0x00
Writing 00000000 -> E000ED08
J-Link>go

J-Link>mem32 0x400000 1
00400000 = 00084A9C

J-Link>mem32 0x22000400,1
22000400 = 000001AC

J-Link>loadfile C:\917CCP\cases\Moen\DualFlashMBR\M4_mbr.bin
0x00
'loadfile': Performing implicit reset & halt of MCU.
Reset: Halt core after reset via DEMCR.VC_CORERESET.
Reset: Reset device via AIRCR.SYSRESETREQ.
Downloading file
[C:\917CCP\cases\Moen\DualFlashMBR\M4_mbr.bin]...
O.K.

J-
Link>VerifyBin  C:\917CCP\cases\Moen\DualFlashMBR\M4_mbr.bi
n 0x00
Loading binary file
C:\917CCP\cases\Moen\DualFlashMBR\M4_mbr.bin
Reading 37540 bytes data from target memory @ 0x00000000.
Verify successful.

J-Link>wreg "R13 (SP)" 0x00020c00
R13 (SP) = 0x00020C00
J-Link>wreg "R15 (PC)" 0x00000249
R15 (PC) = 0x00000249
J-Link>w4 0xE000ED08 0x00
Writing 00000000 -> E000ED08
J-Link>go

J-Link>mem32 0x400000 1
00400000 = 00081E9C

J-Link>mem32 0x22000400,1
22000400 = 00001032
```

4 Revision History

Revision No	Version No	Date	Changes
1	0.1	April 2023	• Initial version


Simplicity Studio
One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

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