

SILICON LABS SiWG917 TA Flash Memory Map User Guide

[Home](#) » [SILICON LABS](#) » SILICON LABS SiWG917 TA Flash Memory Map User Guide 

Contents

- [1 SILICON LABS SiWG917 TA Flash Memory Map](#)
- [2 Introduction](#)
- [3 Prerequisites](#)
- [4 MBR Version check](#)
- [5 Programming the MBR](#)
- [6 Configuring the M4 Application](#)
- [7 SiWx917 Board recovery](#)
- [8 Simplicity Studio](#)
- [9 Disclaimer](#)
- [10 Trademark Information](#)
- [11 Documents / Resources](#)
 - [11.1 References](#)
- [12 Related Posts](#)



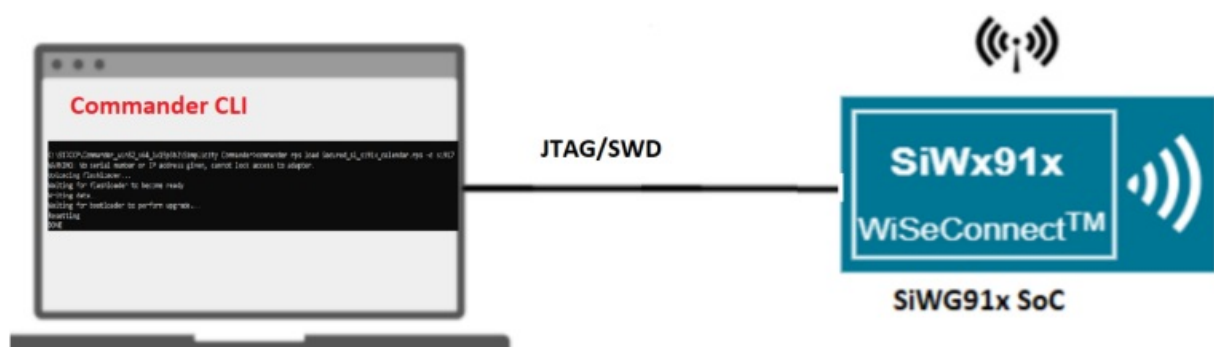
SILICON LABS SiWG917 TA Flash Memory Map



Introduction

- For SiWG917 IC (SiWG917M111MGTBA), Flash is shared between the Wireless processor (TA) and Cortex M4 processor(M4). So far, the Wireless image has been 1.6MB. All eval boards and IC shipments done before Oct 2023 are based on this configuration.
- The Master boot Record (MBR) and SW releases are based on this 1.6MB Wireless image. New Wireless enhancements/features adoption may increase the size of the Wireless image to 1.8MB. Users are recommended to change their existing devices to support
- Wireless Image with 1.8MB and make them future-proof. For this change to affect, an MBR upgrade is needed.
- This document is a user guide that helps in identifying the MBR version of the device and guides the user to update the MBR (if required) using the Commander CLI tool. The tool performs best with systems running on Windows, Linux, and MacOS.

Users must adapt the change based on the results from the MBR version check. If 'MBR version check' returns 1F, then no change to MBR is needed. In this case, users can directly make project-related changes per section Configuring the M4. In case the MBR value is 1B, the user needs to use Simplicity Commander CLI and update the MBR per the following section starting from Simplicity Commander CLI.



- **Note:** After the MBR is updated, it is mandatory to follow Configuring the M4 Application section and flash the application.
- If the above step is skipped, the device will be corrupted and may be difficult to recover in certain scenarios.

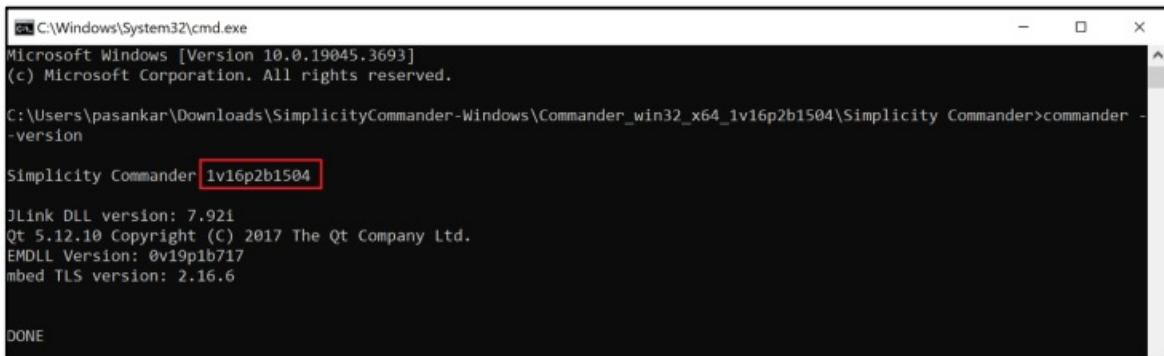
Prerequisites

Hardware

- BRD4338A with BRD4002A.
- USB Type C cable to connect to the PC.

Software

- [Simplicity Commander CLI](#) (1v16p1 and above versions)
- **Note:** To check commander version open the cli mode (To open the commander cli mode follow Section 3.1 from step 1 to step 3) and give the following.
- **command:** commander –version



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.3693]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pasankar\Downloads\SimplicityCommander-Windows\Commander_win32_x64_1v16p2b1504\Simplicity Commander>commander -
-version

Simplicity Commander 1v16p2b1504

JLink DLL version: 7.921
Qt 5.12.10 Copyright (C) 2017 The Qt Company Ltd.
EMDLL Version: 0v19p1b717
mbed TLS version: 2.16.6

DONE
```

MBR Version check

Using Simplicity Commander CLI

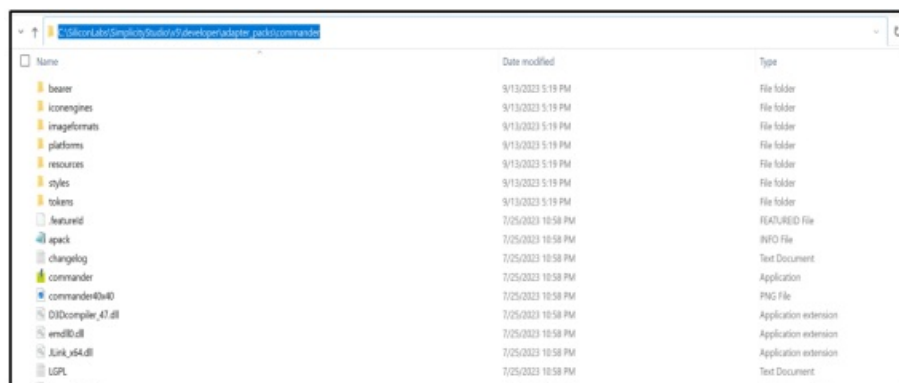
- This section will guide the user to read the address location “0x4000194”.
- Depending on the output the user will know what MBR version is loaded in the board/IC.

Steps to follow:

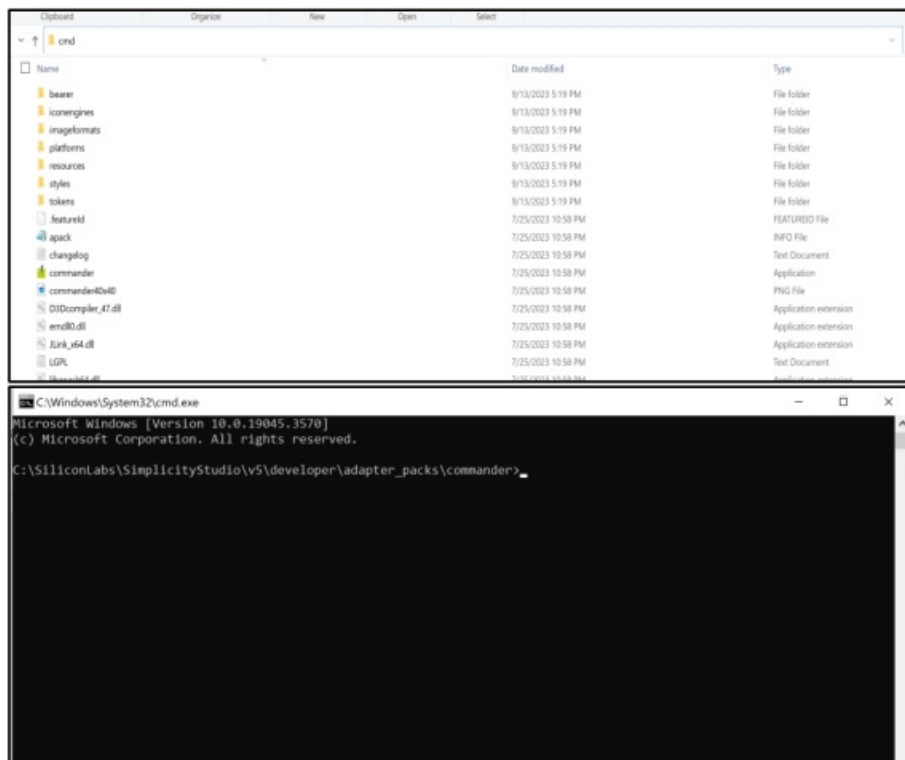
1. Connect the device to the PC using a USB cable (type C).
2. Traverse to the path where Simplicity Studio is installed

- **For our case, the path (default path) is:**

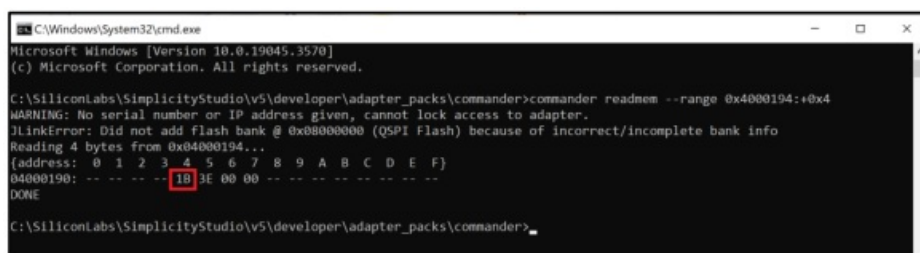
C:\SiliconLabs\SimplicityStudio\v5\developer\adapter_packs\commander



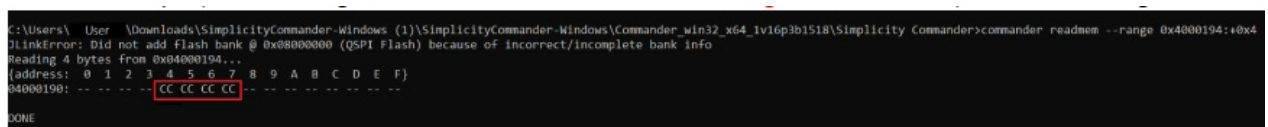
3. Type “cmd” in highlighted part section in the above picture and hit enter. It will open the command prompt CLI in that path.



4. Enter the following command in the command prompt CLI to read the MBR from the memory location address "0x4000194". Command: `commander readmem --range 0x4000194:+0x4`
5. The user would get output similar to below after executing the above command.



6. If the output contains "1B"(as marked in a red box in the image above) then the device has 1.6 MB MBR flashed in it Go to the section "Programming the MBR"
 - If the output contains "1F" (same place where the red box is in the image above) then the device has 1.8 MB MBR flashed in it.
 - Users can use the GA release to proceed or Go to the section "Configuring the M4 Application" if using a release earlier than the GA release to find out linker changes.
 - **Note:** If the output (after executing the command: `commander readmem --range 0x4000194:+0x4`) is like the below figure: Having all "CC"s(marked inside red box), then the board is corrupted. Please go to Section 6 to reflash 1.8v MBR in it.



Programming the MBR

The following steps are for Programming the MBR in the Common Flash Device(s).

Backup the original TA, M4, and efuse contents

- This step is recommended before doing any update.

- **TAMBR:** commander manufacturing read tambr –out <filename.bin>
- **Example:** commander manufacturing read tambr –out tambr.bin
- **M4 MBR:** commander manufacturing read m4mbrcf –out <filename.bin>
- **Example:** commander manufacturing read m4mbrcf –out m4mbr.bin
- **eFusecopy:** commander manufacturing read efusecopy –out <filename.bin>
- **Example:** commander manufacturing read efusecopy –out efusecopy.bin

MBR File(s)

The file that shall be used to update the MBR to support the 1.8MB Wireless Image.

Board Number	MBR File Link
BRD4338A	ta_mbr_SiWG917M1xxMGTBA.bin

- **Note:** PSRAM support is disabled in the above-mentioned MBR file, if PSRAM support is needed, please contact Silicon Labs support.
- Download this file and copy it to the commander folder. E.g. In this case default path
C:\SiliconLabs\SimplicityStudio\v5\developer\adapter_packs\commander

Flashing Procedure

Following is the sequence to program the device(s).

1. Write TA MBR
2. Write M4 MBR
3. Write the calibration data to the M4 Flash

Write TA MBR

- Use the below command to update the TA MBR.
- **Command:** commander manufacturing provision –mbr <filename.bin> -d <full opn>
- **Example:** commander manufacturing provision –mbr ta_mbr_SiWG917M1xxMGTBA.bin -d SiWG917M111MGTBA

```
C:\Users\...Downloads\SimplicityCommander-Windows\Simplicity Commander>commander manufacturing provision --mbr ta_mbr_SiWG917M111MGTBA.bin -d SiWG917M111MGTBA
Loading TA firmware. Writing 321504 bytes to 0x00400000...
Setting TA PC to 0...
Writing 0xA0AC to 0x41050034...
Found PC set to 0x0000F344... (RELEASE_REG=0x00000000, HOLD_REG=0x00000000)
Loading TA firmware. Writing 321504 bytes to 0x00400000...
Setting TA PC to 0...
Writing 0xA0AC to 0x41050034...
Command: 0xa022, Address: 0x00000000, Length: 0x00000000, Flags: 0x00000010
Writing 12 bytes to 0x0042d000...
Interrupting TA...
Command completed with code: 0xa05a
Programming TA MBR...
Command: 0xa00a, Address: 0x00000000, Length: 0x00000210, Flags: 0x00000020
Writing 12 bytes to 0x0042d000...
Writing 528 bytes to 0x0042d00c...
Interrupting TA...
Command completed with code: 0xa05a
Data loaded successfully
DONE
```

- **Note:** User might see failure multiple times while updating the TA MBR(refer to below image). “Reset” the board and try again.

```

C:\Users\...\Downloads\SimplicityCommander-Windows\Simplicity Commander>commander manufacturing provision --mbr ta_mbr_SiWG917M111MGTBA.bin -d SiWG917M111MGTBA
Loading TA firmware. Writing 321504 bytes to 0x00400000...
Setting TA PC to 0...
Writing 0xA0AC to 0x41050034...
Command: 0xA022, Address: 0x00000000, Length: 0x00000000, Flags: 0x00000010
Writing 12 bytes to 0x0042d000...
Interrupting TA...
Timeout waiting for firmware to complete command (0x4105003c = 0x0000, PC0 = 0x610926, PC2 = 0x01bc)
ERROR: Update flash info failed.
ERROR: Could not execute provisioning sequence!
DONE

C:\Users\...\Downloads\SimplicityCommander-Windows\Simplicity Commander>commander manufacturing provision --mbr ta_mbr_SiWG917M111MGTBA.bin -d SiWG917M111MGTBA
Loading TA firmware. Writing 321504 bytes to 0x00400000...
Setting TA PC to 0...
Writing 0xA0AC to 0x41050034...
Command: 0xA022, Address: 0x00000000, Length: 0x00000000, Flags: 0x00000010
Writing 12 bytes to 0x0042d000...
Interrupting TA...
Timeout waiting for firmware to complete command (0x4105003c = 0x0000, PC0 = 0x610926, PC2 = 0x01bc)
ERROR: Update flash info failed.
ERROR: Could not execute provisioning sequence!
DONE

C:\Users\...\Downloads\SimplicityCommander-Windows\Simplicity Commander>
C:\Users\...\Downloads\SimplicityCommander-Windows\Simplicity Commander>
C:\Users\...\Downloads\SimplicityCommander-Windows\Simplicity Commander>commander manufacturing provision --mbr ta_mbr_SiWG917M111MGTBA.bin -d SiWG917M111MGTBA
Loading TA firmware. Writing 321504 bytes to 0x00400000...
Setting TA PC to 0...
Writing 0xA0AC to 0x41050034...
Command: 0xA022, Address: 0x00000000, Length: 0x00000000, Flags: 0x00000010
Writing 12 bytes to 0x0042d000...
Interrupting TA...
Timeout waiting for firmware to complete command (0x4105003c = 0x0000, PC0 = 0x61b016, PC2 = 0x0fe2)
ERROR: Update flash info failed.
ERROR: Could not execute provisioning sequence!
DONE

C:\Users\...\Downloads\SimplicityCommander-Windows\Simplicity Commander>commander manufacturing provision --mbr ta_mbr_SiWG917M111MGTBA.bin -d SiWG917M111MGTBA
Loading TA firmware. Writing 321504 bytes to 0x00400000...
Setting TA PC to 0...
Writing 0xA0AC to 0x41050034...
Command: 0xA022, Address: 0x00000000, Length: 0x00000000, Flags: 0x00000010
Writing 12 bytes to 0x0042d000...
Interrupting TA...
Timeout waiting for firmware to complete command (0x4105003c = 0x0000, PC0 = 0x61b016, PC2 = 0x0fe2)
ERROR: Update flash info failed.
ERROR: Could not execute provisioning sequence!
DONE

```

Write M4 MBR

- TA and M4 regions have the same data in both their MBR regions. Pick the relevant binary files from section “5.2 MBR File(s)” and use the below command to update the M4 MBR
- Note:** User needs to use the same MBR for both TA and M4 thus in this case use ta_mbr_SiWG917M1xxMGTBA.bin
- Command:** commander manufacturing write m4mbrcf --data <filename.bin> -d <full opn>
- Example:** commander manufacturing write m4mbrcf --data ta_mbr_SiWG917M1xxMGTBA.bin -d SiWG917M111MGTBA

```

C:\Users\...\Downloads\SimplicityCommander-Windows\Simplicity Commander>commander manufacturing write m4mbrcf --data ta_mbr_SiWG917M111MGTBA.bin -d SiWG917M111MGTBA
Writing data to region: m4mbrcf
Reading 496 bytes from 0x04000000
Loading TA firmware. Writing 321504 bytes to 0x00400000...
Setting TA PC to 0...
Writing 0xA0AC to 0x41050034...
Found PC set to 0x0000b2c8... (RELEASE_REG=0x00000000, HOLD_REG=0x00000000)
Loading TA firmware. Writing 321504 bytes to 0x00400000...
Setting TA PC to 0...
Writing 0xA0AC to 0x41050034...
Command: 0xA022, Address: 0x00000000, Length: 0x00000000, Flags: 0x00000010
Writing 12 bytes to 0x0042d000...
Interrupting TA...
Command completed with code: 0xA05a
Command: 0xA00a, Address: 0x001f0000, Length: 0x000001f0, Flags: 0x00000020
Writing 12 bytes to 0x0042d000...
Writing 528 bytes to 0x0042d00c...
Interrupting TA...
Command completed with code: 0xA05a
Data loaded successfully
Region 'm4mbrcf' was successfully written to device.
DONE

```

Write the calibration data to the M4 Flash

- Step 1:** Copy the calibration data from TA to a bin file.
- Step 2:** Write the copied data to M4 Flash (the same bin file will be given as input).
 - Note:** In case the above procedure fails please reset the board a few times and retry the steps.

Copy the Calibration data from TA to a bin file

- Command:** commander manufacturing read taipmu --out <filename.bin>
- Example:** commander manufacturing read taipmu --out ipmu.bin

Write the copied data to M4 Flash

- Command:** commander manufacturing write m4ipmucf --data <file.bin> -d <full opn>

- **Example:** commander manufacturing write m4ipmucf –data ipmu.bin -d SiWG917M111MGTBA
- **Note:** After the flashing is done, read the location 0x4000194. It should return 1F. Refer to section MBR Version check for steps to read the MBR version
- Update the TA firmware. Refer to [Upgrade SiWx91x Connectivity Firmware](#). Else use the TA firmware image that is included in the SDK that you are using.
- **Note:** In case the above procedure fails please reset the board a few times and retry the steps.

Configuring the M4 Application

- **Note:** This is a mandatory section to follow. If not followed, the device will be corrupted and is irrecoverable.
- Wi-Fi SDK 3.1.0 is the latest release that is out when this document is made. Users using 3.1.0 or older release(s) need to make some configuration changes to their projects.
- Unless this is done, the applications will not work with the updated MBR. The next release is GA 3.1.1 will have this addressed by default.
- Here are the configuration changes that shall be made in the Application projects.
 1. In the rsi_ipmu.h file, update the following Macros with the corresponding addresses given below.
 - **#define** PACKAGE_TYPE_VALUES_OFFSET_COMMON_FLASH 0x81F0292
 - **#define** SILICON_REV_VALUES_OFFSET_COMMON_FLASH 0x81F0293
 - **#define** COMMON_FLASH_IPMU_VALUES_OFFSET 0x81F0258
 - Path to rsi_ipmu.h file: wiseconnect3_sdk_3.1.0 > siwx917_soc > drivers > systemlevel > inc > rsi_ipmu.h
 2. Change the origin address of rom to 0x8202000 in the linker_SoC.ld file of the project. The linker file is available under the “autogen” folder.

```
MEMORY
{
    rom      (rx)  : ORIGIN = 0x8202000, LENGTH = 0x6e000
    ram      (rwx) : ORIGIN = 0xc, LENGTH = 0x30000
    psram    (rwx) : ORIGIN = 0xa000000, LENGTH = 0x800000
}
```

- **Note:** For POWER SAVE-related applications the following address needs to be changed in the preprocessor defines -IVT_OFFSET_ADDR = 136323072
- 3. Clean and build the project and flash it in the device. The application should work as mentioned in its readme file.

Note Older addresses:

- **#define** PACKAGE_TYPE_VALUES_OFFSET_COMMON_FLASH 0x81B0292
- **#define** SILICON_REV_VALUES_OFFSET_COMMON_FLASH 0x81B0293
- **#define** COMMON_FLASH_IPMU_VALUES_OFFSET 0x81B0258
- **rom (rx)** : ORIGIN = 0x81c2000, LENGTH = 0x6e000

SiWx917 Board recovery

- Due to 2 variants of boot loaders shipped to our customers as part of the alpha program, there is a possibility of MBR corruption that results in board failure.
- The following two instances would result in MBR corruption on the SiWx917 Board.

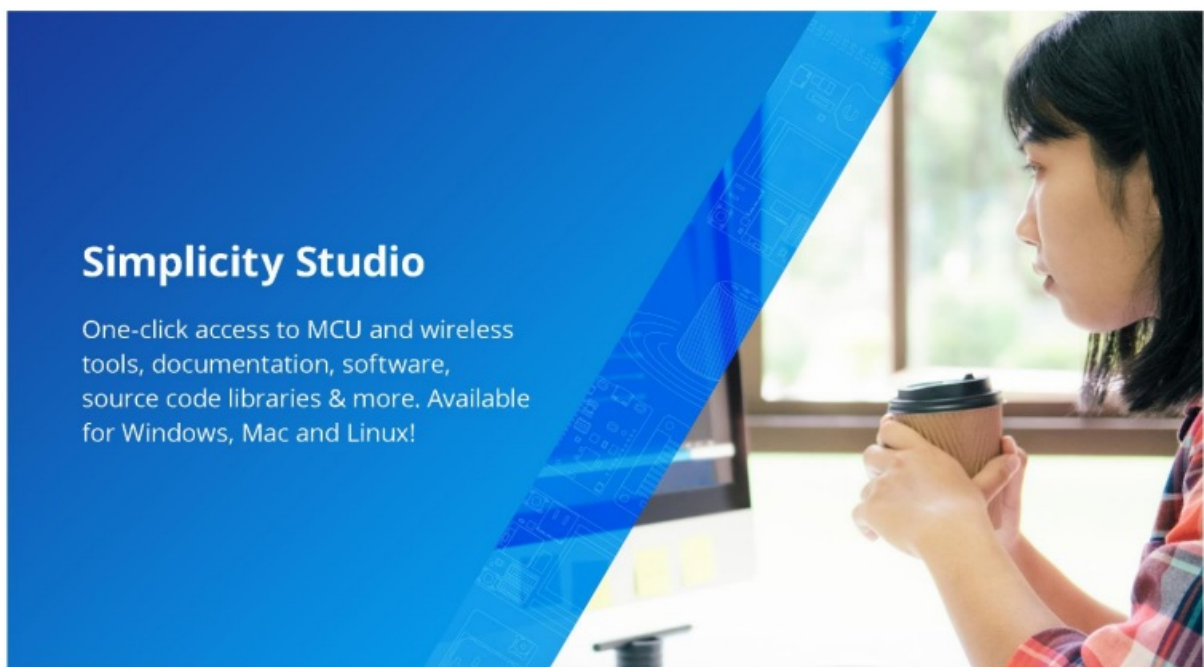
1. The BRD4338A board with 1.8v MBR, if the user tries to flash any matter application from <https://docs.silabs.com/matter/2.1.1/matter-wifi-getting-started/> or any example from matter extension release version(2.1.1) and SMG release(2.2.0-1.2)
2. The BRD4338A board with 1.6v MBR, if the user tries to update/flash the firmware from version 2.9.0.0.30 or later.

Steps to follow to reflash the 1.8v MBR(recovery steps):

- Download the MBR file: [ta_mbr_SiWG917M1xxMGTBA.bin](#)
- Execute the below commands sequentially to reflash the MBR.
 1. Commander manufacturing provision –mbr ta_mbr_SiWG917M1xxMGTBA.bin -d SiWG917M111MGTBA
Example: commander manufacturing provision –mbr ta_mbr_SiWG917M111MGTBA.bin -d SiWG917M111MGTBA
 2. commander manufacturing write m4mbrcf –data <filename.bin> -d SiWG917M111MGTBA
 - **Example:** commander manufacturing write m4mbrcf –data ta_mbr_SiWG917M111MGTBA.bin -d SiWG917M111MGTBA
 3. commander manufacturing read taipmu –out filename.bin
 4. commander manufacturing write m4ipmucf –data filename.bin

Simplicity Studio

- [One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac, and Linux!](#)



IoT Portfolio



SW/HW



Quality



Support & Community

- IoT Portfolio www.silabs.com/loT
- SW/HW www.silabs.com/simplicity

- **Quality** www.silabs.com/quality
- **Support & Community** www.silabs.com/community

Disclaimer

Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes, and memory addresses refer to each specific device, and “Typical” parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice to the product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Without prior notification, Silicon Labs may update product firmware during the manufacturing process for security or reliability reasons. Such changes will not alter the specifications or the performance of the product. Silicon Labs shall have no liability for the consequences of the use of the information supplied in this document. This document does not imply or expressly grant any license to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any FDA Class III devices, applications for which FDA premarket approval is required or Life Support Systems without the specific written consent of Silicon Labs. A “Life Support System” is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological, or chemical weapons, or missiles capable of delivering such weapons. Silicon Labs disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to the use of a Silicon Labs product in such unauthorized applications.


Note: This content may contain offensive terminology that is now obsolete. Silicon Labs is replacing these terms with inclusive language wherever possible. For more information, visit www.silabs.com/about-us/inclusive-lexicon-project

Trademark Information

Silicon Laboratories Inc., Silicon Laboratories, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, EFM®, EFM32®, EFR, Ember®, Energy Micro, Energy Micro logo and combinations thereof, “the world’s most energy friendly microcontrollers”, Redpine Signals®, WiSeConnect, n-Link, ThreadArch®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, Gecko OS, Gecko OS Studio, Precision32®, Simplicity Studio®, Telegesis, the Telegesis Logo®, USBXpress®, Zentri, the Zentri logo and Zentri DMS, Z-Wave®, and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3, and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. Wi-Fi is a registered trademark of the Wi-Fi Alliance. All other products or brand names mentioned herein are trademarks of their respective holders.

- **Silicon Laboratories Inc.**
- **400 West Cesar Chavez**
- **Austin, TX 78701**
- **USA**
- www.silabs.com
- silabs.com
- **Smart. Connected. Energy-friendly.**

Documents / Resources

 SiWG917 - TA Flash Memory Map Change Guide Version 1.1 November 2015 <small>THIS DOCUMENT IS UNCLASSIFIED AND IS NOT FOR DISTRIBUTION OUTSIDE THE COMPANY. IT IS THE PROPERTY OF SILICON LABS AND IS NOT TO BE REPRODUCED OR TRANSMITTED IN ANY FORM OR BY ANY MEANS, ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPYING, RECORDING, OR BY ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM. ANY UNAUTHORIZED REPRODUCTION OR TRANSMISSION OF THIS DOCUMENT IS STRICTLY PROHIBITED. FOR MORE INFORMATION, CONTACT SILICON LABS AT 400-873-7373.</small>	SILICON LABS SiWG917 TA Flash Memory Map [pdf] User Guide SiWG917 TA Flash Memory Map, SiWG917, TA Flash Memory Map, Memory Map, Map
---	--

References

- [User Manual](#)

[Manuals+](#), [Privacy Policy](#)

This website is an independent publication and is neither affiliated with nor endorsed by any of the trademark owners. The "Bluetooth®" word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. The "Wi-Fi®" word mark and logos are registered trademarks owned by the Wi-Fi Alliance. Any use of these marks on this website does not imply any affiliation with or endorsement.