SCM1612
Wi-Fi 6 and BLE 5 Low-Power SoC

Manufacturing Guide

Revision 0.2
Apr 15, 2025

Contact Information
Senscomm Semiconductor (www.senscomm.com)
Room 303, International Building, West 2 Suzhou Avenue,
SIP, Suzhou, China
For sales or technical support, please send email to
info@senscomm.com

Disclaimer and Notice

This document is provided on an “as-is” basis only. Senscomm reserves the right to make corrections, improvements and other changes to it or any specification contained herein without further notice.

All liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No licenses express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

All third party’s information in this document is provided as is with NO warranties to its authenticity and accuracy.

All trade names, trademarks and registered trademarks mentioned in this document are property of their respective owners and are hereby acknowledged.

1 Version History

Version	Date	Description
0.1	2024-04-08	Initial draft
0.2	2025-04-15	Description for 2MB/4MB flash configuration

Table of Contents

2 Introduction

This document describes how to use manufacturing (MFG) test firmware.

2.1 Overview

The manufacturing test firmware allows the user to perform device test such as RF calibration and to program the eFuse based on the test result.

There are 2 different ways to use the MFG firmware. The features are the same, but they differ in how the firmware is loaded.

- **RAM mode**: The MFG firmware is loaded into the volatile areas temporarily for the test. It takes less time to load the firmware.
- **XIP mode**: The MFG firmware is programmed into the flash memory and the test can be repeated without losing the firmware over the power cycles. Compared to RAM mode, It takes more time to load the firmware.

3 RAM mode

3.1 Building the firmware

To build the MFG (RAM mode) firmware

```
$ make distclean
$ make scm1612_mfg_ram_defconfig
$ make
```

→ creates wise.mfg.ram.bin

3.2 Loading the firmware

Follow the steps below to load the firmware.

1) Set UART Boot Mode
2) Using the flash tool extension, load the firmware
   `sctool.exe boot_image mfg/wise.mfg.ram.bin`
3) MFG firmware is ready to be used.
4) Perform the test and update the eFuse fields.

4 XIP mode

4.1 Building the firmware

XIP mode provides an additional benefit of flashing the production firmware at the same time. Depending on the requirements, the proper defconfig may be needed. Since there're two kinds of Flash (2MB/4MB) integrated, we need to handle it separately.

4.1.1 Case for 2MB Flash

To build the bootloader,

```
$ make distclean
$ make scm1612s_bl_2m_defconfig
$ make
```

→ creates wise.scmboot.bin

To build the main firmware,

```
$ make distclean
$ make scm1612s_2m_defconfig
$ make
```

→ creates wise.mcuboot.bin

To build the MFG (XIP mode) firmware,

```
$ make distclean
$ make scm1612_mfg_xip_2m_defconfig
$ make
```

→ creates wise.mfg.xip.bin

4.1.2 Case for 4MB Flash

To build the bootloader,

```
$ make distclean
$ make scm1612s_bl_defconfig
$ make
```

→ creates wise.scmboot.bin

To build the main firmware,

```
$ make distclean
$ make scm1612s_defconfig
$ make
```

→ creates wise.mcuboot.bin

To build the MFG (XIP mode) firmware,

```
$ make distclean
$ make scm1612_mfg_xip_defconfig
$ make
```

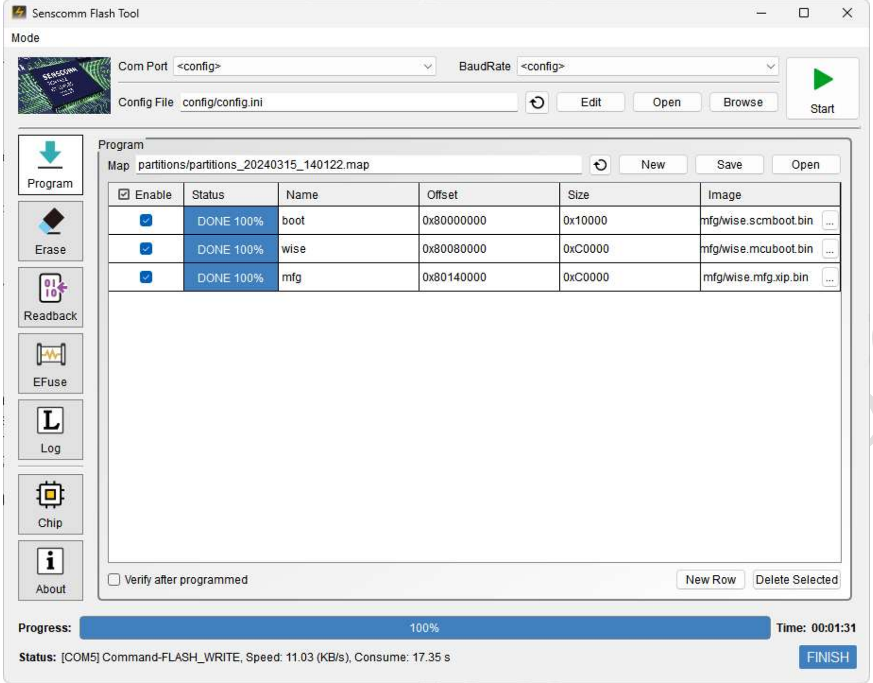
→ creates wise.mfg.xip.bin

4.2 Loading the firmware

Follow the steps below to load the firmware.

1) Set UART Boot Mode
2) Using the flash tool, load all the firmware images.

Please note that the Address of MFG bin is different, depends on if the Flash is 2MB or 4MB.
For the case of 2MB Flash, the Offset of MFG bin is 0x80140000.



Senscomm Flash Tool

Mode

Com Port <config> BaudRate <config>

Config File config/config.ini

Program

Map partitions/partitions_20240315_140122.map

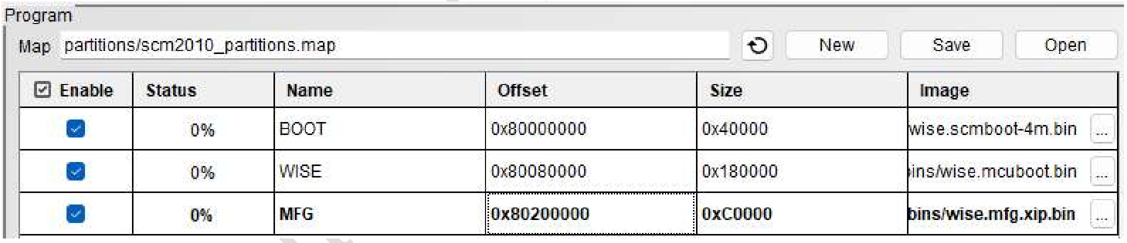
☑ Enable	Status	Name	Offset	Size	Image
☑	DONE 100%	boot	0x80000000	0x10000	mfg/wise.scmboot.bin
☑	DONE 100%	wise	0x80080000	0xC0000	mfg/wise.mcuboot.bin
☑	DONE 100%	mfg	0x80140000	0xC0000	mfg/wise.mfg.xip.bin

□ Verify after programmed

Progress: 100% Time: 00:01:31

Status: [COM5] Command-FLASH_WRITE, Speed: 11.03 (KB/s), Consume: 17.35 s FINISH

While for the case of 4MB Flash, the Offset of MFG bin is 0x80200000.



Program

Map partitions/scm2010_partitions.map

☑ Enable	Status	Name	Offset	Size	Image
☑	0%	BOOT	0x80000000	0x40000	wise.scmboot-4m.bin
☑	0%	WISE	0x80080000	0x180000	ins/wise.mcuboot.bin
☑	0%	MFG	0x80200000	0xC0000	bins/wise.mfg.xip.bin

3) MFG firmware is flashed, but not ready to be used.
4) Set XIP Boot Mode, and reboot
5) Perform the test and update the eFuse fields
6) Erase the MFG firmware

After rebooting wiht XIP mode, the bootloader detects the presence of the MFG firmware, and the bootloader will boot the MFG firmware.

Once the test is done with the MFG firmware, it should be marked that the test is complete. Upon next reboot, bootloader will not detect the MFG firmware, therefore, boot the normal firmware. This process can be done with “mfg done” command. The command only works for the XIP mode.

5 Test and eFuse

5.1 eFuse command

eFuse can be written only one time. For the convenience of device testing, eFuse buffer mode is supported.

- Raw Mode
  - The eFuse commands works on the real eFuse.
- Ram Buffer Mode
  - The eFuse commands read and write to a RAM space.
  - The values are lost over the power cycle.
  - The values can be loaded from the eFuse or synchonized to the eFuse
- Flash Buffer Mode
  - The eFuse commands read and write to a designated flash memory.
  - The values are maintained over the power cycle
  - The values can be loaded from the eFuse or synchonized to the eFuse

To see the available eFuse commands, type “efuse help”

```
$ efuse
Usage: efuse read  [bit offset] [bit length]
  or:  efuse write [bit offset] [bit length] [value in hex]
  or:  efuse clear_buffer [bit offset] [bit length]
  or:  efuse set_mode [mode]
       0 : EFUSE RAW mode
       1 : EFUSE RAM buffer mode
       2 : EFUSE FLASH buffer mode
  or:  efuse get_mode
  or:  efuse load
  or:  efuse sync
```

5.2 Flash Buffer Mode

Using the flash buffer mode, the eFuse values can be collected to the flash memory and synchronized to the real eFuse at once.

To set the flash buffer mode

```
$ efuse set_mode 2
```

To load the values from the eFuse to the flash

```
$ efuse load
```

To set the RF calibrataion data

```
$ efuse write 672 1 1    # bank 1 valid 21*32
$ efuse write 676 4 XX   # tx power     21*32 + 4
$ efuse write 680 4 XX   # rssi         21*32 + 8
$ efuse write 684 6 XX   # cfo          21*32 + 12
```

When using the flash buffer mode, the values can also be cleared.

```
$ efuse clear_buffer 684 6  # clear 6 bits from offset 684
$ efuse clear_buffer 0 1024 # clear all
```

To synchronize the values from the flash to the eFuse.

```
$ efuse sync
```

6 Removing MFG firmware

6.1 Removing MFG firmware

The MFG firmware for XIP mode is a special header field so that the bootloader recognizes it and boots the firmware.

After the test is complete, the MFG firmware can be removed with the following command

```
$ mfg done
```

After this command is executed, the bootloader will not recognize the MFG firmware anymore. The next time the device powers up, the bootloader will boot the normal firmware instead of MFG firmware.