RL78/I1C(512KB) FOTA Sample Software

FOTA Demo Example Project

Table of Contents

Table of Figures

1. Definition of Terms

FOTA	Firmware update Over-The-Air
Bank swap	There are two banks, bank 0 (256KB) for current execution area and bank 1 (256KB) for new firmware, swap the bank 1 and bank 0 by using the Bank Swap Library
Bank programming	Software can run on bank 0 and program the flash on bank 1 simultaneously
Self-programing	The operation that takes place when the firmware update target is the firmware itself
FSL	Flash Self-programming Library

2. Introduction

This document describes an overview of the software design used in the FOTA demo package, r01an5860es0100-rl78i1c512k-fota-demo-package.zip.

3. FOTA Functionality

Bootloader methods or approaches can be broadly divided into few types, depending on where the data are buffered when downloading new firmware.

For example, new firmware information location can be one of the banks with bank swap mechanism used or not used, or external memory (EEPROM or serial flash memory).

In addition, there are a few options available as the communication channel through which the firmware is downloaded, e.g., via UART, via the file system on a storage device (SD card or USB flash driver) or through Ethernet / Wi-Fi.

These sample software for both Fast FOTA and Continuous Metrology FOTA are using the same method: Dual bank / Bank swap mechanism used. And downloaded the new firmware via UART.

3.1. Dual bank / Bank swap mechanism

The RL78/I1C(512KB) MCU has two 256KB banks, referred to as Dual Bank, to support the integration of Fast FOTA and Continuous Metrology FOTA. The bank swap mechanism allows for firmware updates using only the internal MCU flash memory.

The banks are initially defined as Bank 0 as the primary execution area, and Bank 1 as the secondary area for new firmware information.

3.2. Method 1: Fast FOTA

3.2.1. Fast FOTA Bank swap function invoke status

During initial operation, the User Application is executed from the code flash Bank 0.

When there is firmware update request, the bank programming function will erase the Bank 1 information and write the new firmware information into Bank 1.

After the write is completed, the flash banks are swapped, the MCU is reset, and the User Application is executed from the code flash Bank 1.

There is no metrology accumulation during the bank swap duration (less than 1 second), so there is some energy measurement loss during the firmware update period. The figure below shows the status during the bank swap for Fast FOTA.

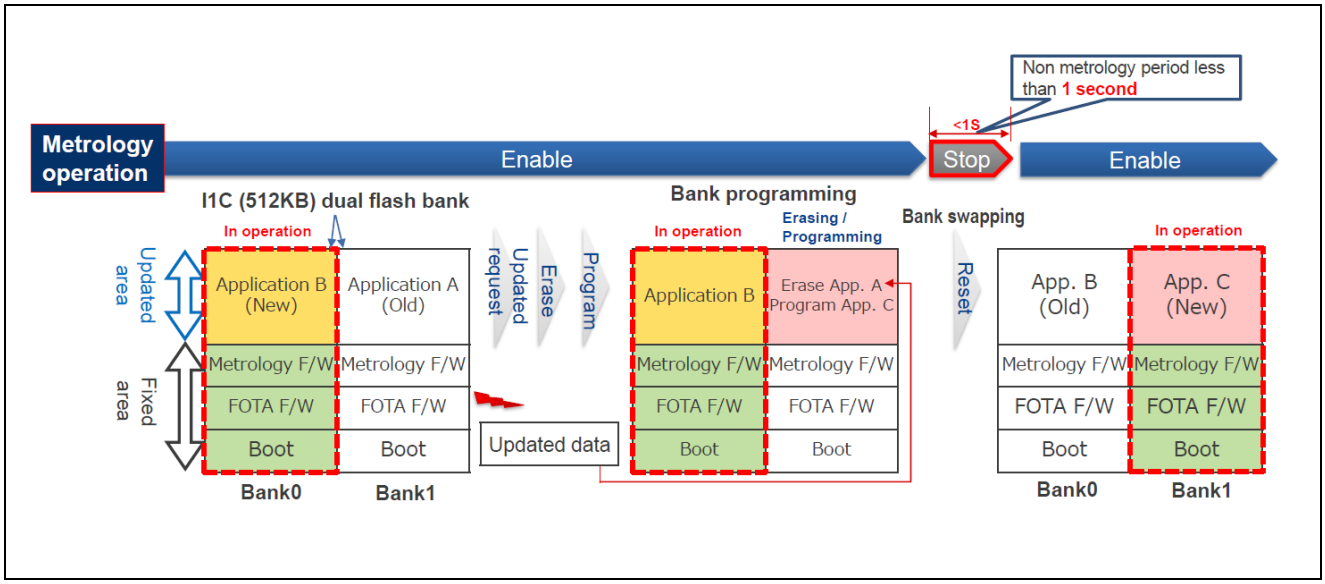


Figure 3-1 Fast FOTA Bank swap status

3.2.2. Fast FOTA sequence

1. Meter accumulates energy data while Metrology is running.
2. User inputs the command to invoke bank swap for Fast FOTA.
3. RL78/I1C(512KB) executes the bank swap action, stopping the Metrology.
4. Bootloader restarts the Meter.
5. Metrology restarts and continues energy accumulation.

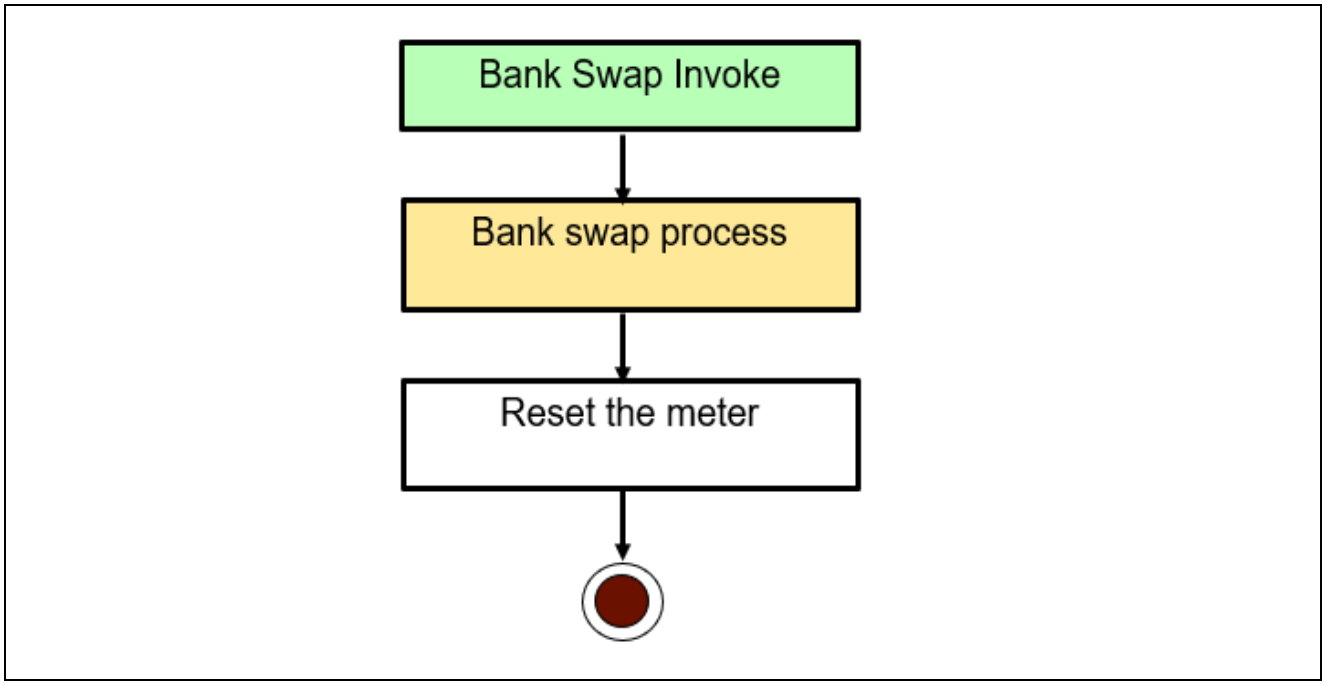


Figure 3-2 Fast FOTA Bank swap sequence

During Fast FOTA, the accumulated energy is backed up to EEPROM before stopping the metrology. The Boot Flag is changed from 0 to 1, or 1 to 0, using the function FSL_InvertBootFlag.

The MCU is then reset by the function FSL_ForceReset, and will boot as normal from the flash bank that was updated by the FOTA image transfer process.

This reset process will cause some loss of energy accumulation. The amount of energy loss depends on the timing of the Boot Swap operation and the alignment of timers used by the metrology.

3.3. Method 2: Continuous Metrology FOTA

3.3.1. Continuous Metrology FOTA Bank swap function invoke status

The proposed new method of firmware update is to continue energy accumulation during the firmware update process. The Meter continues to accumulate energy by running code on RAM, while the bank swap process changes the ROM bank used for code execution.

Using this method, there is no energy measurement loss during firmware update period, and the Meter does not need to reset after bank swapping. The figure below shows the status of the Meter during the bank swap for Continuous Metrology FOTA.

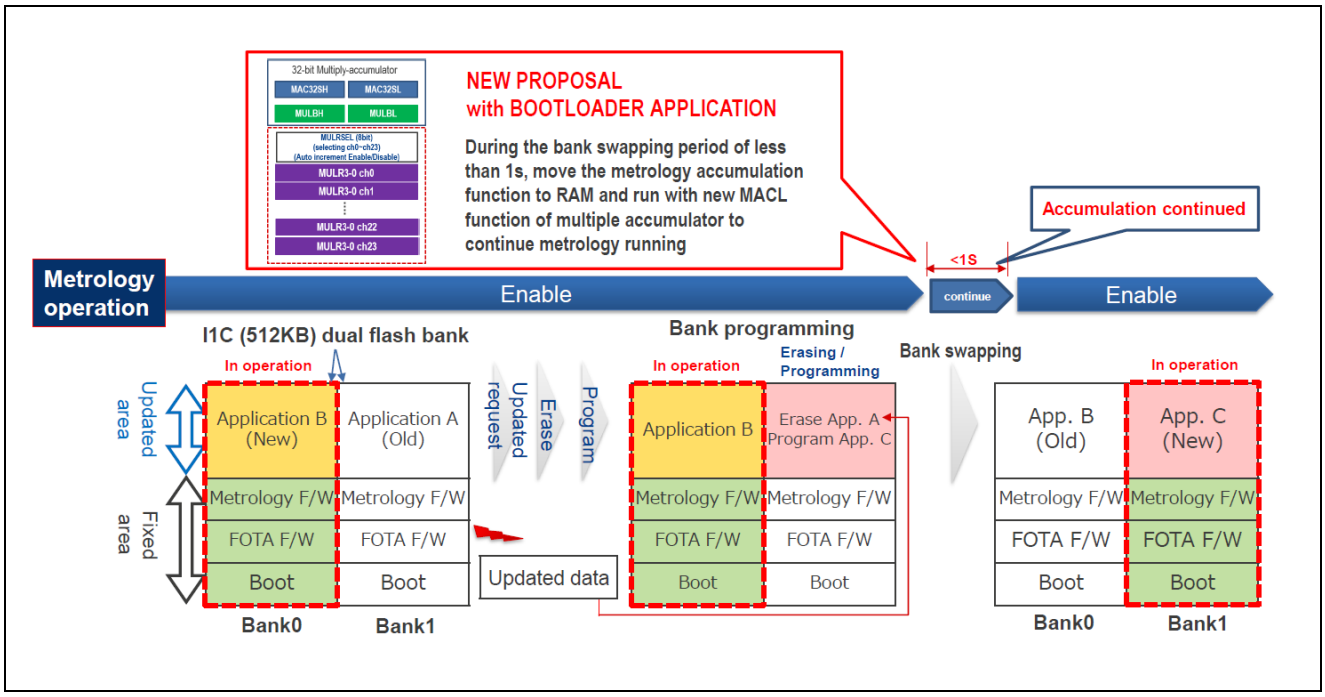


Figure 3-3 Continuous Metrology FOTA Bank swap status

3.3.2. Continuous Metrology FOTA sequence

1. Meter accumulates energy data while Metrology is running.
2. User inputs the command to invoke bank swap for Continuous Metrology FOTA.
3. Bootloader copies the predefined code from ROM to RAM.
4. Change the interrupt vector table from ROM to RAM.
5. RL78/I1C(512KB) executes the bank swap action. Metrology continues running on RAM.
6. Change back the interrupt vector table from RAM to ROM.

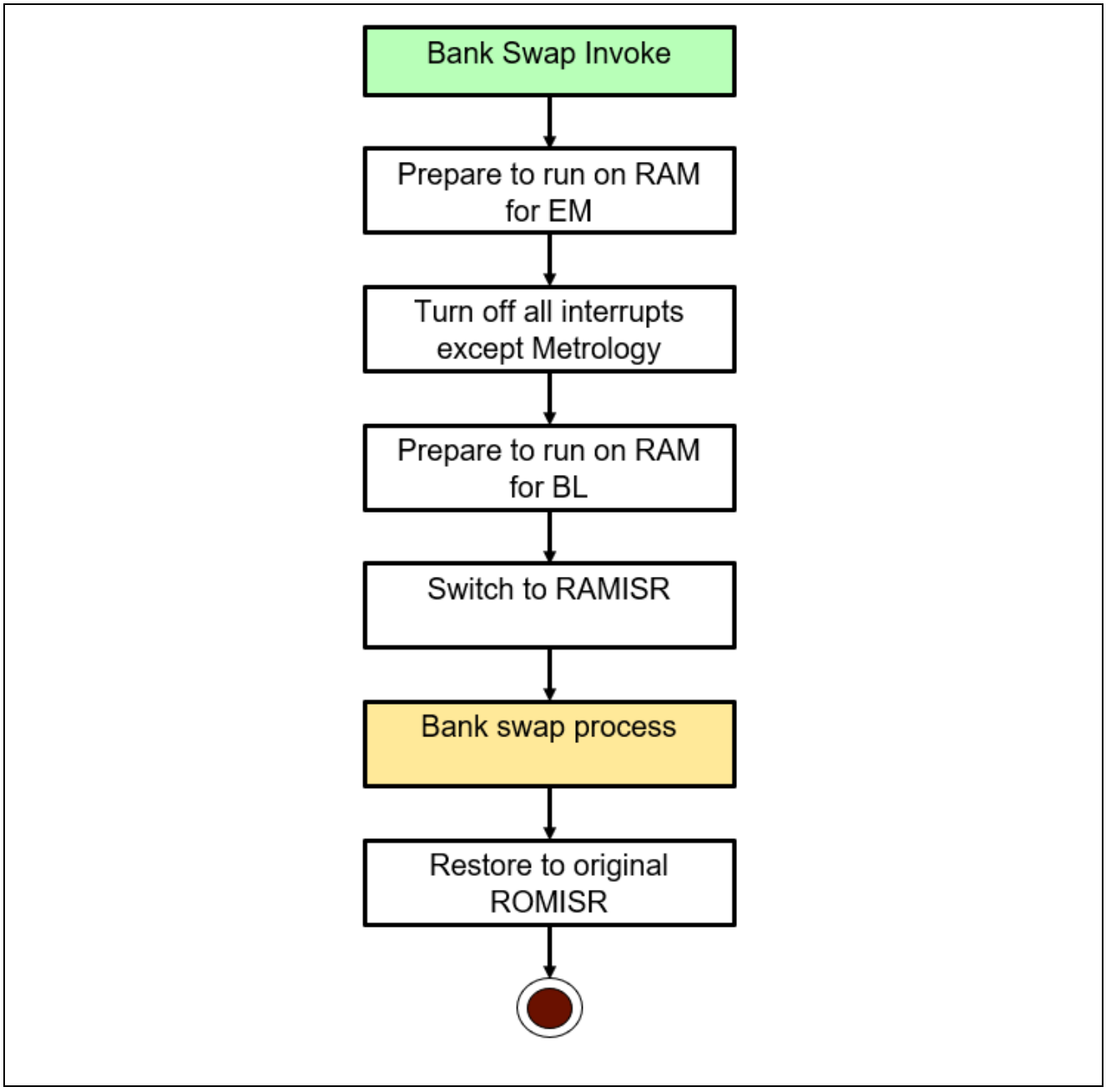


Figure 3-4 Continuous Metrology FOTA Bank swap sequence

During normal operation, the Metrology accumulates energy through functions and DSAD interrupts on ROM.

In order to continue running the Metrology during the bank swap, the required functions and interrupt vector table are copied from ROM to RAM, allowing for basic energy accumulation to continue exclusively on RAM.

After the bank swap operation is finished, the interrupt vector table is moved back to ROM for normal operation to resume.

As there is no MCU reset and the metrology continues running, there is no loss of energy from the “Continuous Metrology FOTA” process, in contrast to the <1s loss caused by the “Fast FOTA” process.

4. Project Structure

In this example project tree, there are three projects, Main project contains all user applications. Metrology subproject contains metrology function and metrology library. Bootloader subproject contains bootloader function and bootloader library.

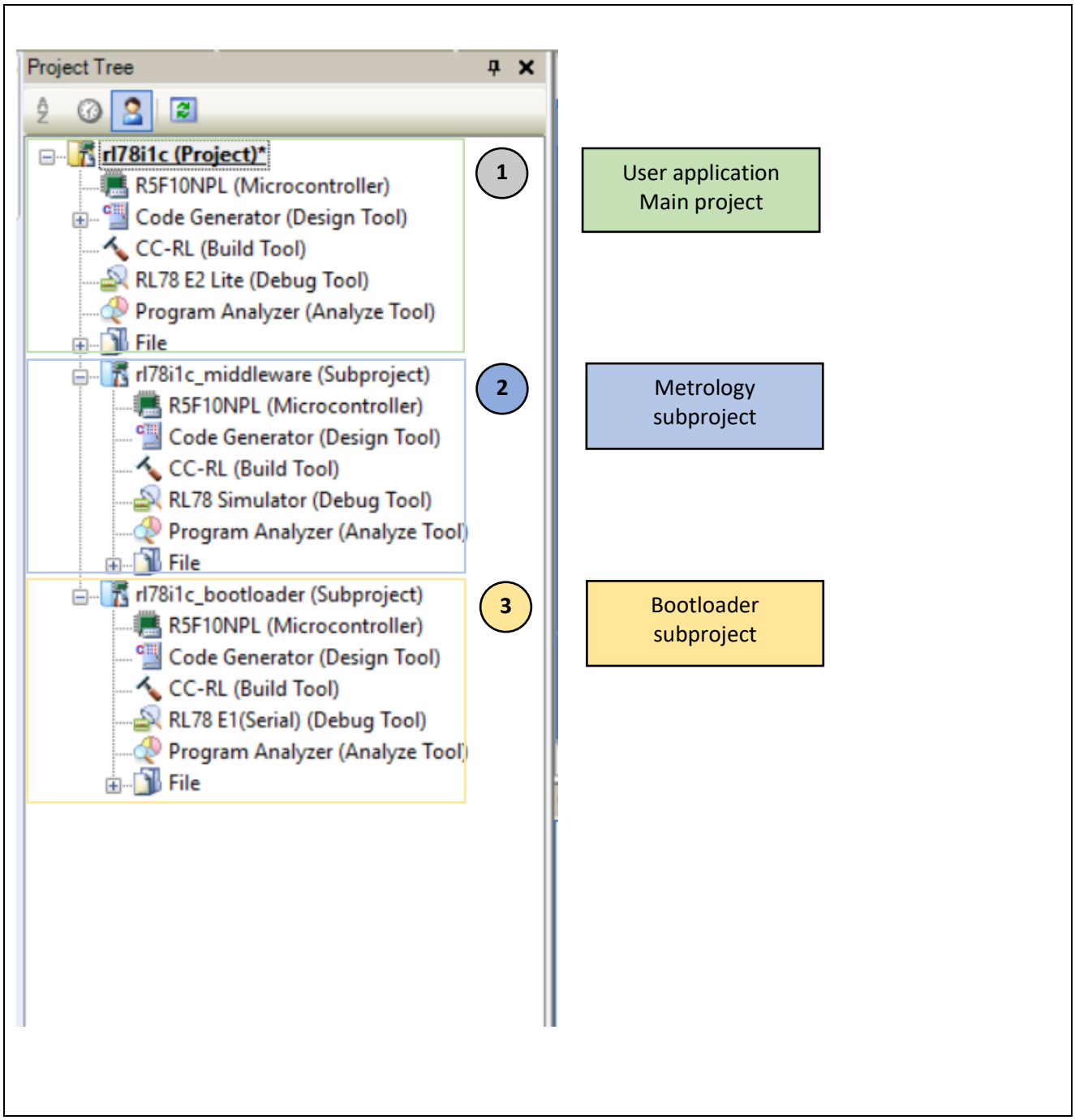


Figure 4-1 Project structure

4.1. Memory allocation for Fast FOTA and Continuous Metrology FOTA

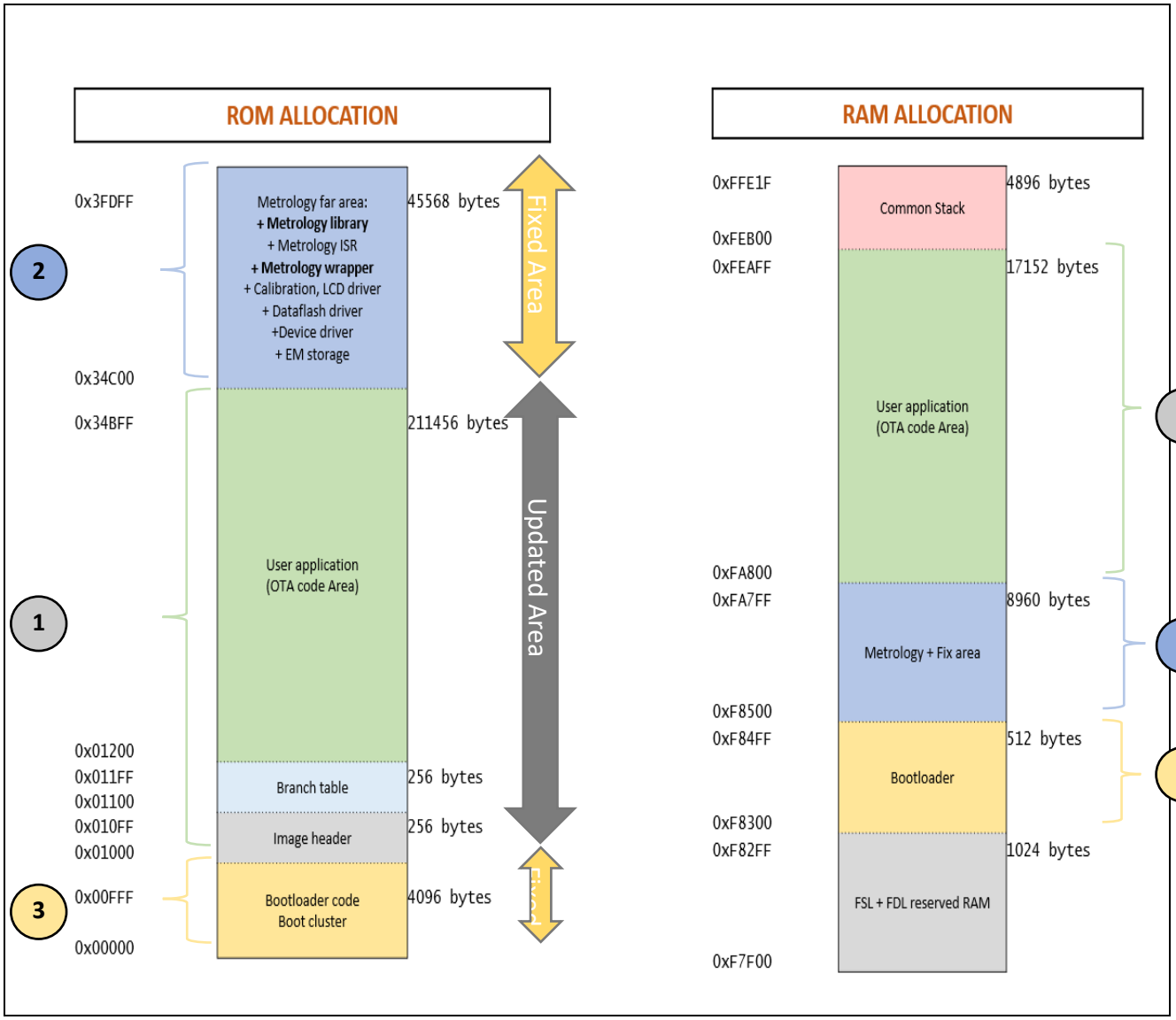


Figure 4-2 Memory allocation for Fast FOTA and Continuous Metrology FOTA

Use application is located in the area defined in **bl_platform.h**

#define USER_APP_START_ADDRESS (0x01000)

#define USER_APP_END_ADDRESS (0x34BFF)

The Hash Range is across the “Updated Area” in Figure 4-2. This is the range for User aaplication rewritten by FOTA. On the other hand, Metrology is located in 0x34c00-3fdff range in the “Fixed Area”, and cannot be rewritten by FOTA.

For the secondary bank, it will add **DEVICE_FLASH_BANK_SIZE** (0x40000) to the addresses.

4.2. Memory location link option in CS+

1. User application layout defined range of ROM and RAM

In the CS+ rl78i1c project, need set the memory location for ROM and RAM: **CC-RL (Build Tool)** -> **Link Options** -> **Verify** -> **Address range of memory type**.

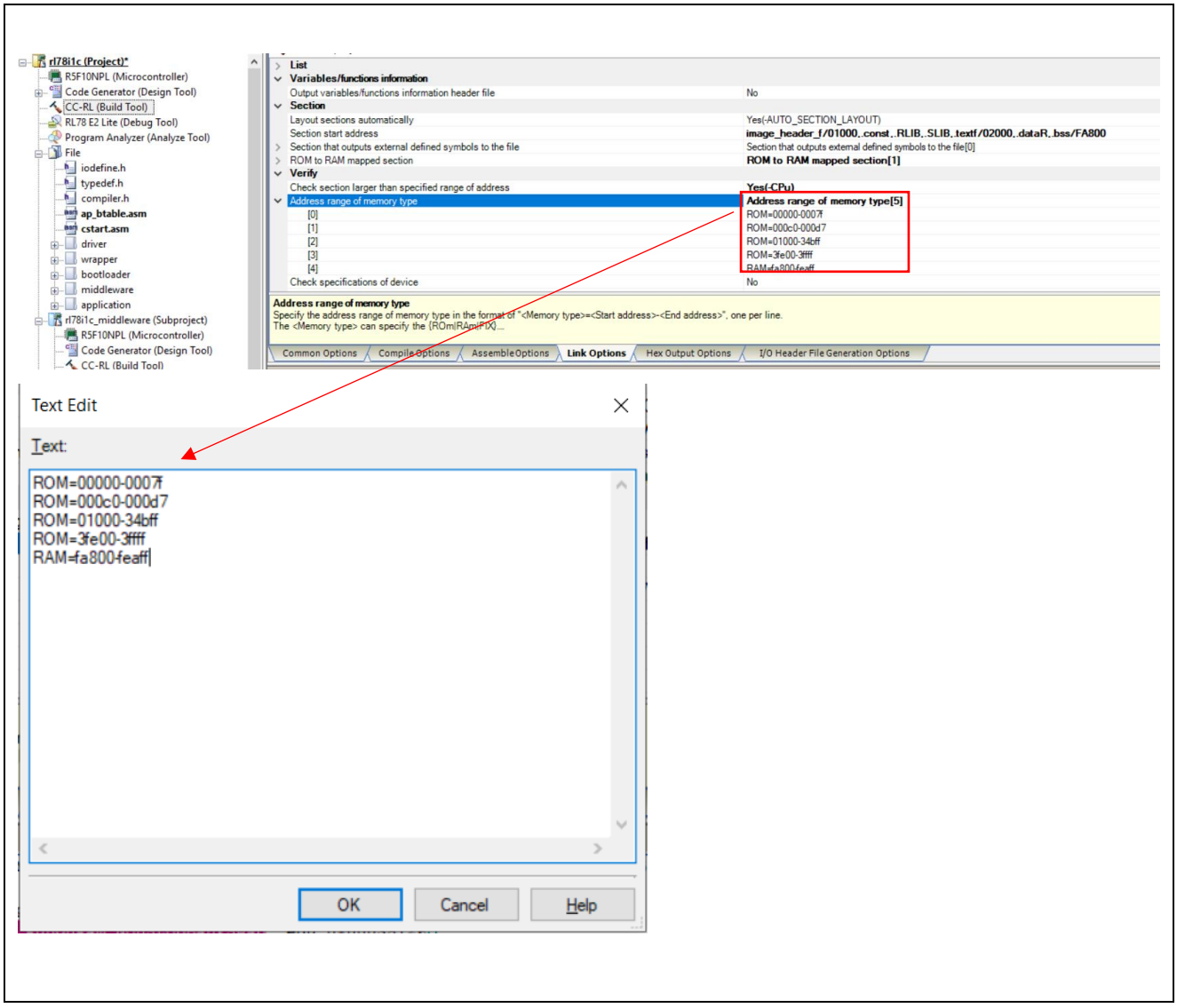


Figure 4-3 Memory location link option

As memory linking is manually assigned, the following sections must be explicitly included for compilation:

0000-0007f: Enables the use of the On-Chip Debugger function.

000c0-000d7: Memory address range for the Option Byte and Security ID required for MCU operation.

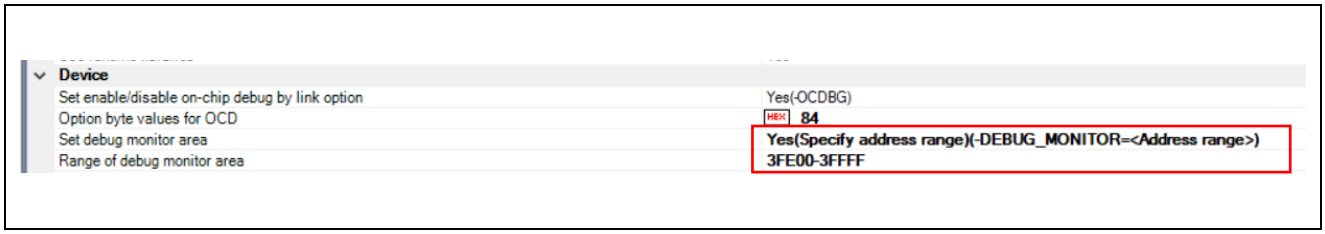


Figure 4-4 Range of debug monitor area

3fe00-3ffff: Defined to include the .monitor2 of debugger, should be aligned with specified monitor area.

Auto section layout with section that need specific location in **CC-RL (Build Tool)** -> **Link Options** -> **Section** -> **Section start address**.

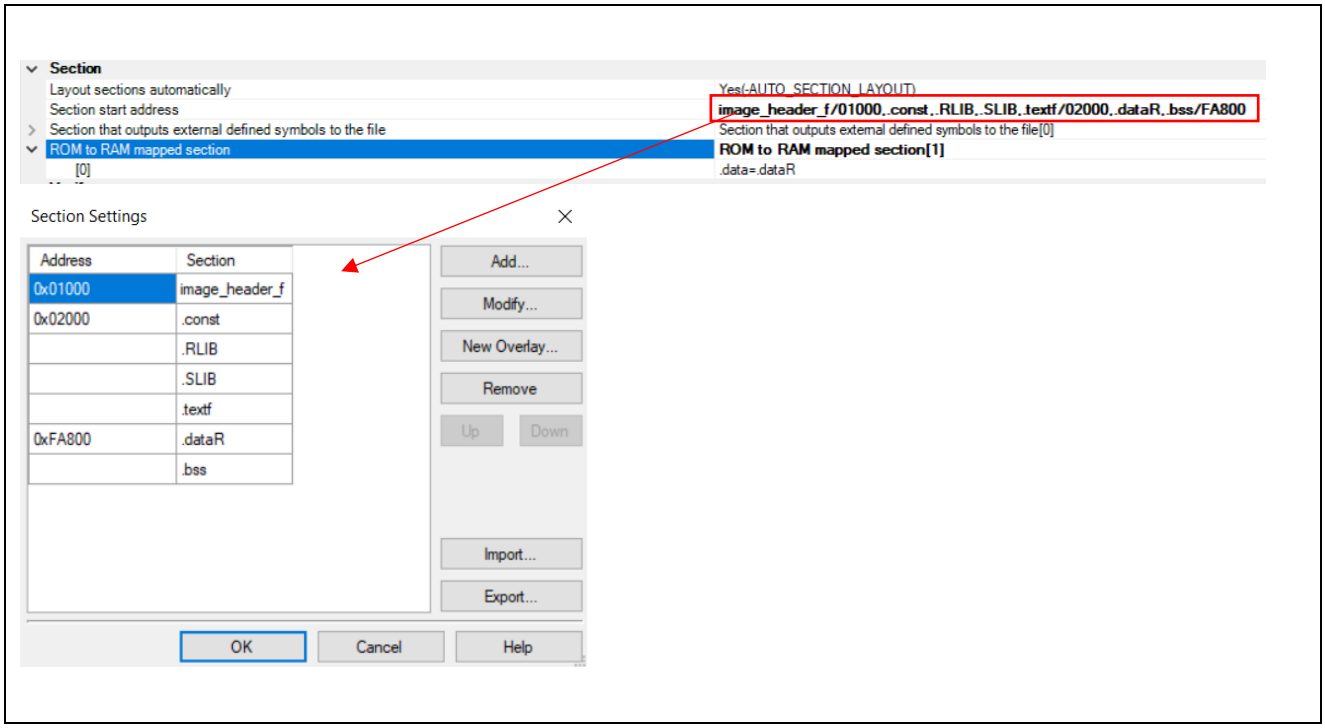


Figure 4-5 User application auto section setting

2. Metrology layout defined range of ROM and RAM

In the CS+ rl78i1c_middleware subproject, need set the memory location for ROM and RAM: **CC-RL (Build Tool)** -> **Link Options** -> **Verify** -> **Address range of memory type**.

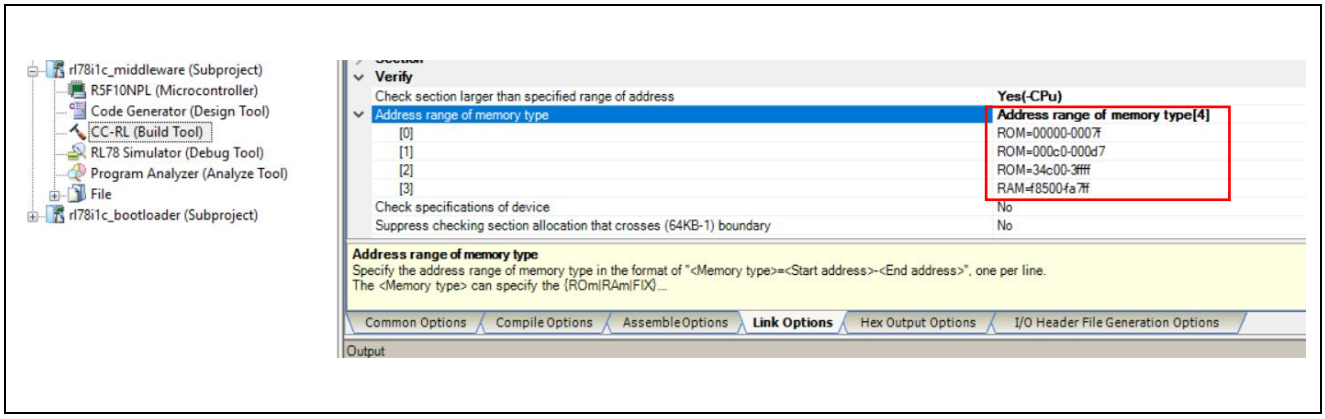


Figure 4-6 Metrology location link option

Auto section layout with section that need specific location in **CC-RL (Build Tool)** -> **Link Options** -> **Section** -> **Section start address and section that outputs external defined symbols to the file**.

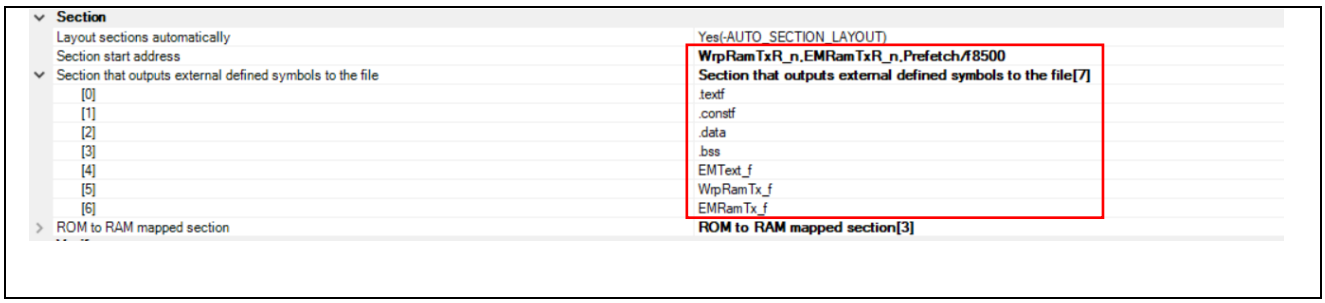


Figure 4-7 Metrology auto section setting

3. Bootloader layout defined range of ROM and RAM

In the CS+ rl78i1c_Bootloader subproject, need set the memory location for ROM and RAM: **CC-RL (Build Tool)** -> **Link Options** -> **Verify** -> **Address range of memory type**.

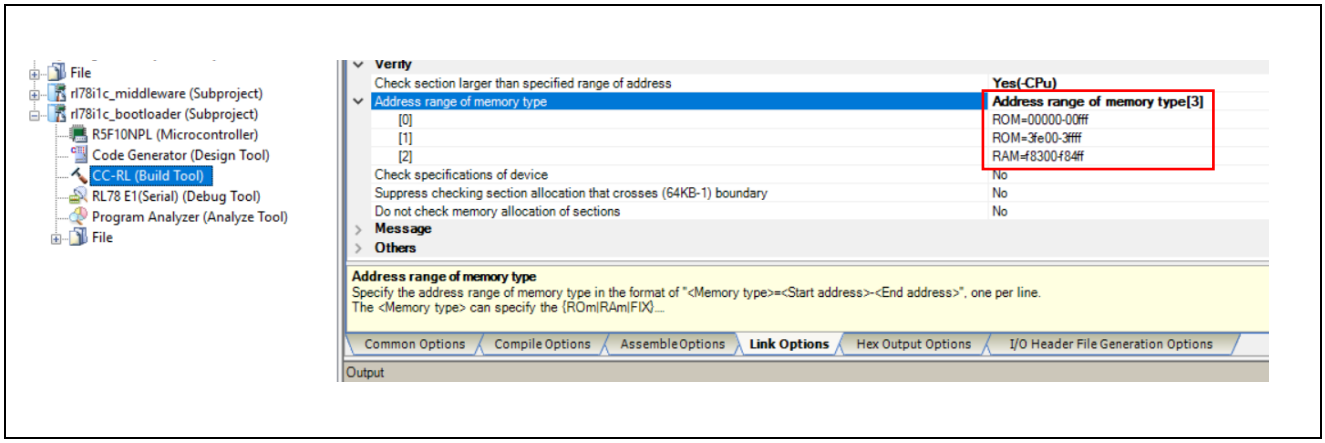


Figure 4-8 Bootloader location link option

Auto section layout with section that need specific location in **CC-RL (Build Tool)** -> **Link Options** -> **Section** -> **Section start address and section that outputs external defined symbols to the file**.

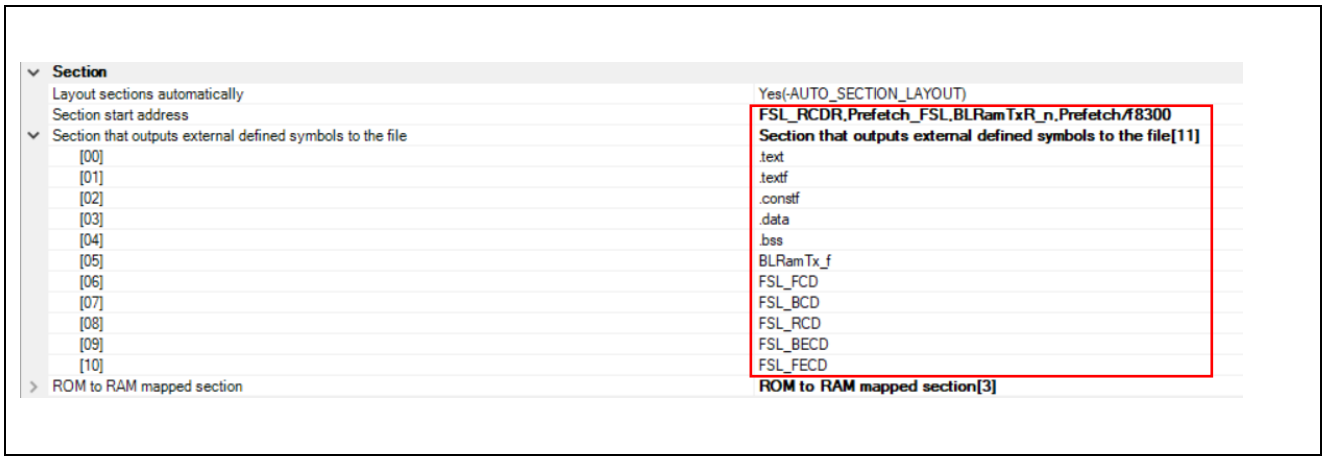


Figure 4-9 Bootloader auto section setting

4.3. Running interrupt service routine on RAM

There are two different routes for signal accumulation in DSAD interrupt: ROM path and RAM path.

- ROM path: interrupt jump to ROM vector table
- RAM path: interrupt jump to RAM function

Only during the firmware update, RAM path is used for temporary energy accumulation (part of) for Continuous Metrology FOTA function. So, we need to move the RAM path code from ROM to RAM before Continuous Metrology FOTA.

Below figure show copy the code portion and do ROMization from ROM to RAM.

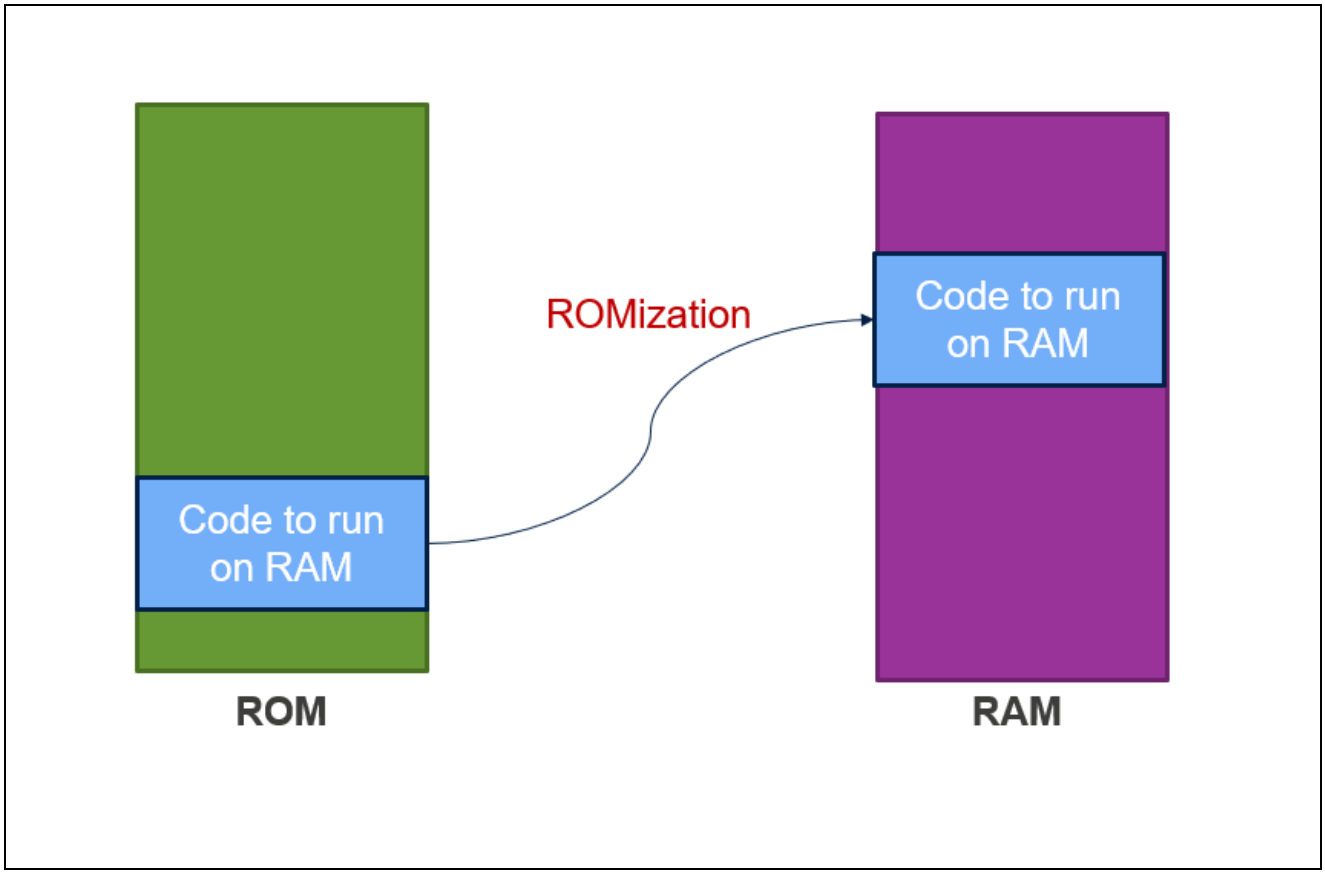


Figure 4-10 Allocate ROM code to RAM portion

The **EM_ADC_InterruptCallback** callback function is replaced by **EM_ADC_RAM_InterruptCallback** during RAM execution. The **r_dsadc_interrupt** is replaced by **EM_RunOnRam_RamIsr** which handles all interrupts during Run on RAM, so it must manually check and clear the **DSAIF** flag to process the DSAD interrupt.

The two functions above are copied from ROM to RAM by the function **EM_RunOnRam_PrepareFunctions**.

4.3.1. ROM mapping section setting in CS+

1. User application project ROM mapping

In the CS+ rl78i1c project mapping the ROM data to RAM data area need to define as below: **CC-RL (Build Tool) -> Link Options -> Section -> ROM to RAM mapped section.**

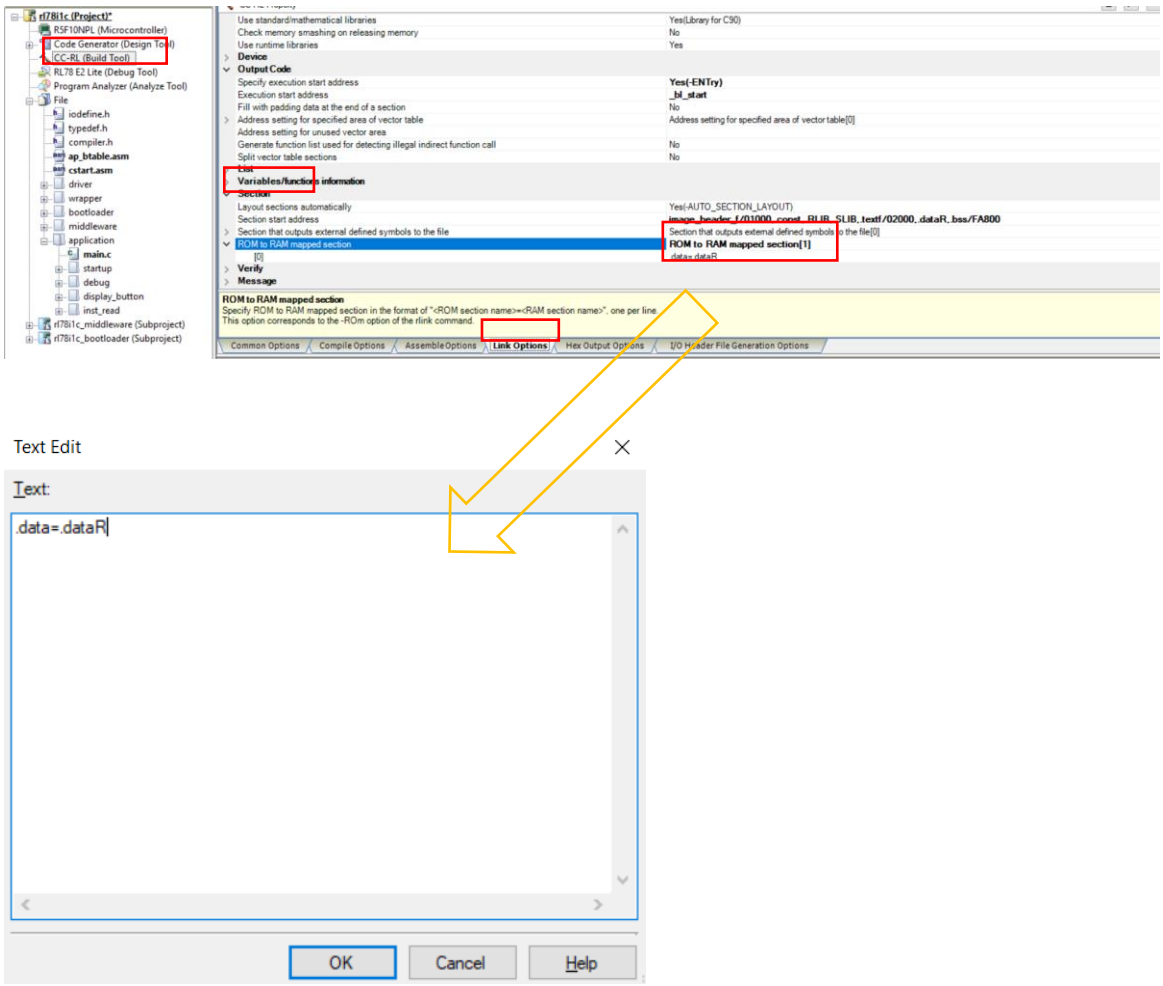


Figure 4-11 User application project ROM section setting

2. Metrology subproject ROM mapping

In the CS+ rl78i1c_Metrology subproject, mapping the ROM data to RAM data area need to define as below: **CC-RL (Build Tool) -> Link Options -> Section -> ROM to RAM mapped section.**

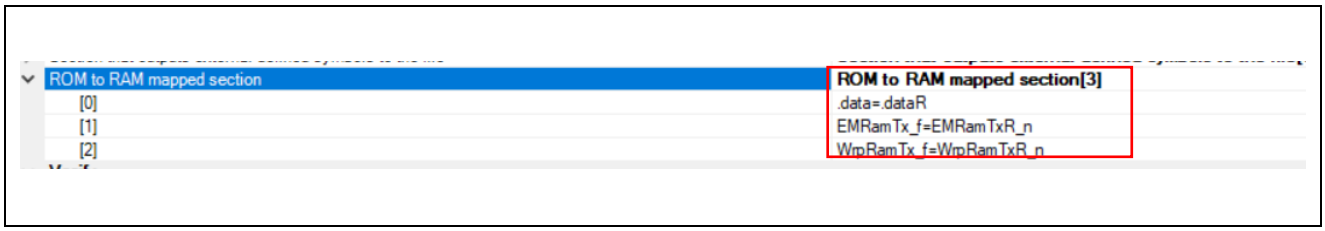


Figure 4-12 Metrology ROM section setting

3. Bootloader layout defined range of ROM and RAM

In the CS+ rl78i1c_Bootloader subproject, mapping the ROM data to RAM data area need to define as below: **CC-RL (Build Tool) -> Link Options -> Section -> ROM to RAM mapped section.**

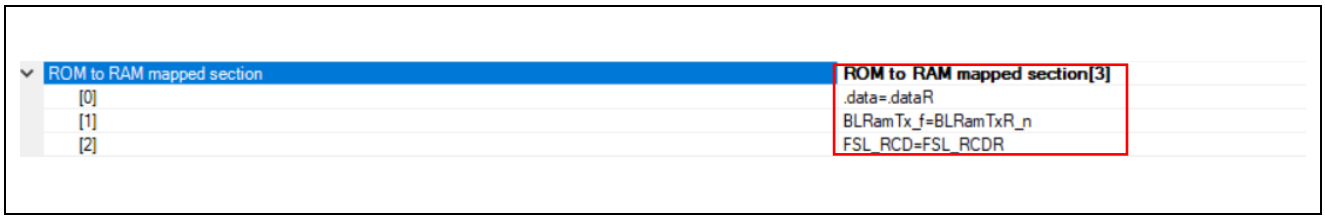


Figure 4-13 Bootloader ROM section setting

4.3.2. Source code related switch from ROM to RAM

The source code in **em_run_on_ram.c** file in rl78i1c_middleware subproject.

```
  /* Prepare functions on RAM */
   EM_RunOnRam_PrepareFunctions();
   EM_RunOnRam_DisableInterruptsExceptMetrology();
   BL_RunOnRam_PrepareFunctions();

  /* Swap with interrupt run on RAM */
   status =  BL_FLASH_RAM_SwapBankWithRamIsr((void * ) &EM_RunOnRam_RamIsr);
   if  (status == BL_OK)
  {
        /* Jump to branch bank swap entry */
        BL_FLASH_RAM_JumpBankSwapEntry();
   }
...
```

```
uint16_t BL_FLASH_RAM_SwapBankWithRamIsr(void * p_ram_isr)
{
    uint16_t status;

    /* Switch to RamIsr */
    status = BL_FLASH_Prepare();
    if (status != BL_OK) { return status; }

    /* Swap flow:
     * ISR --> RAM (preset function pointer set in BL_FLASH_SetRamIsrFunction)
     * Swap
     * ISR --> ROM
     * Jump bankswap entry
     * ** If bank swap error, return status code
    */
    DI();
    FSL_ChangeInterruptTable((fsl_u16)p_ram_isr);
    EI();
```

FSL_ChangeInterruptTable function used to point RAM vector table.

4.4. Branch table flow

For the Continuous Metrology FOTA, using branch table for branching the interrupt vector table to user application interrupt functions.

Branching would be faster than function pointer calls.

User code and bootloader code can share the vector but cannot change the vector dynamically.

4.4.1. Split vector table section

The metrology is fixed, and branch table split into 2 parts, **ap_btable.asm** and **em_btable.asm**.

In the CS+ rl78i1c_Bootloader subproject, split vector table section need: **CC-RL (Build Tool) -> Link Options -> Output Code -> Split vector table sections.**

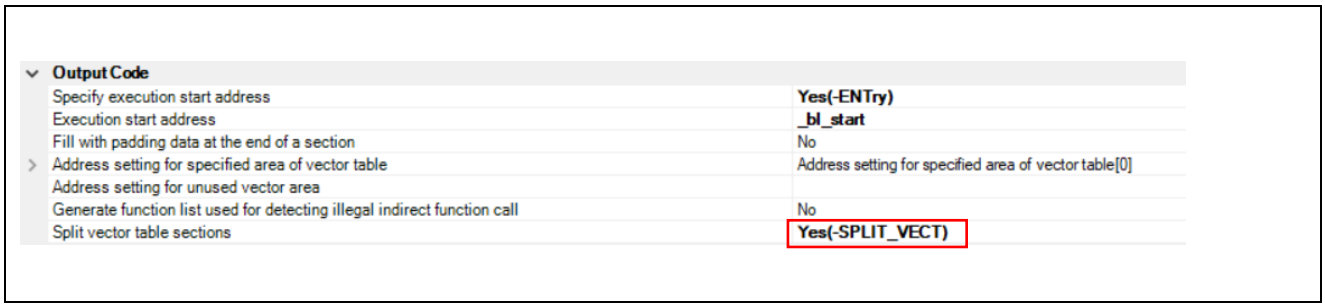


Figure 4-14 Spilt vector table

Below figure shows branch table flow for metrology part.

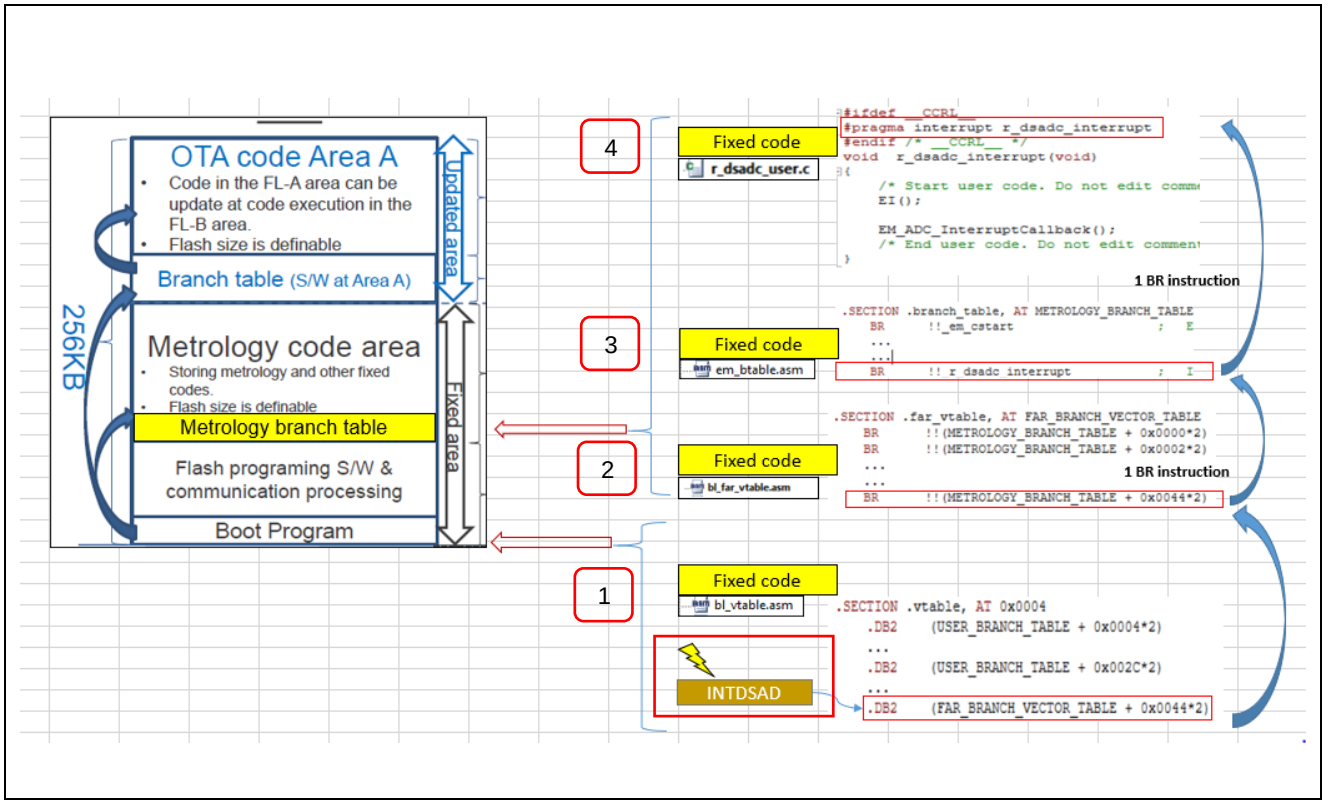


Figure 4-15 Branch table flow

When DSAD interrupt **INTDSAD** occur at step 1, it will jump to defined location

```
.DB2     (FAR_BRANCH_VECTOR_TABLE + 0x0044*2)
```

But due to fixed metrology now located in far address, so it jump to an intermediate branch table at step 2. Then to jump to metrology branch table address at step 3 and go to user application interrupt function **r_dsadc_interrupt** at step 4.

Note: The source code related file list down as below

1 **bl_vtable.asm** in in rl78i1c_bootloader subproject

2 **bl_far_vtable.asm** in rl78i1c_bootloader subproject

3 **em_btable.asm** in rl78i1c_metrology subproject

4 **r_dsadc_user.c** in rl78i1c_project

4.5. Signal accumulation during bank swap

- The same accumulation path must be retained (DSAD Interrupt, signal processing, signal accumulation) when allocating to RAM for temporary accumulation during Continuous Metrology FOTA bank swap.
- However, executing the whole same code in RAM could not enough time since code execution on RAM usually take long clock cycle (take minimum 4 times longer according to RL78 software manual)
- Maintain a minimum code path (active and reactive energy) when executing code on RAM to accumulate signals to power accumulator (used for energy accumulation).

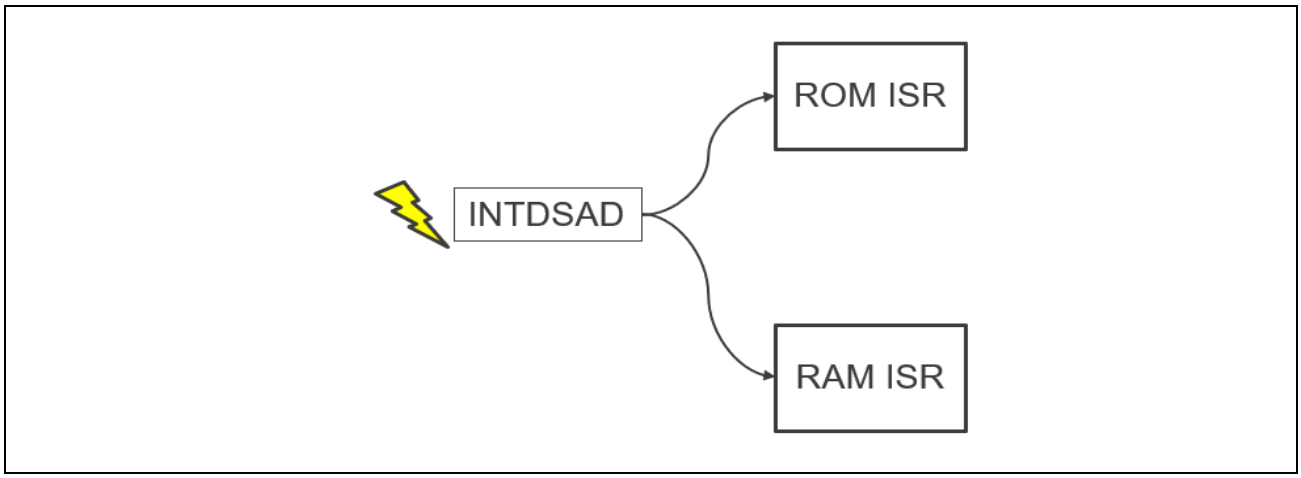


Figure 4-16 Signal accumulation in DSAD interrupt

Figure 4-17 show the ISR execution from ROM and RAM during the normal operation and Bank Swap operation.

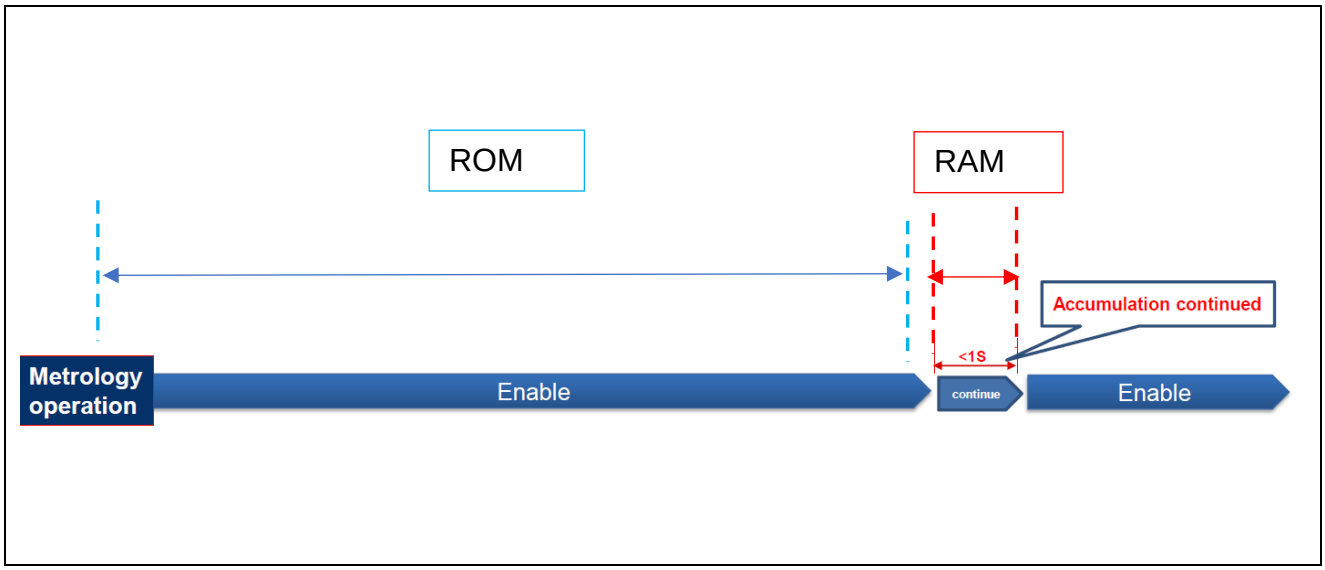


Figure 4-17 Metrology operation status

4.5.1. Signal sampling path in ROM

During normal operation, DSAD ISR run form ROM ISR, which will perform the full accumulation for all parameters for metrology as shown in figure 4-18.

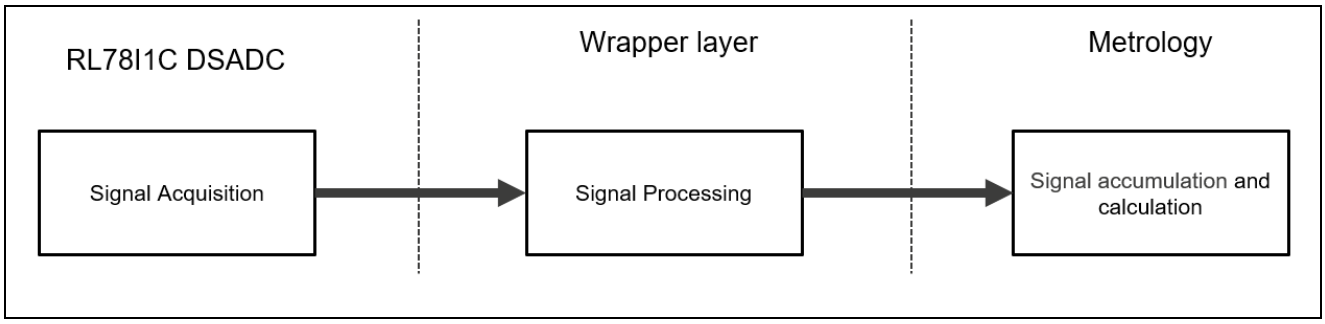


Figure 4-18 Signal accumulation path in ROM

Figure 4-19 shows the source code for signal sampling path in ROM in **wrp_em_adc.c** file, **EM_ADC_InterruptCallback** function in rl78i1c_middleware subproject.

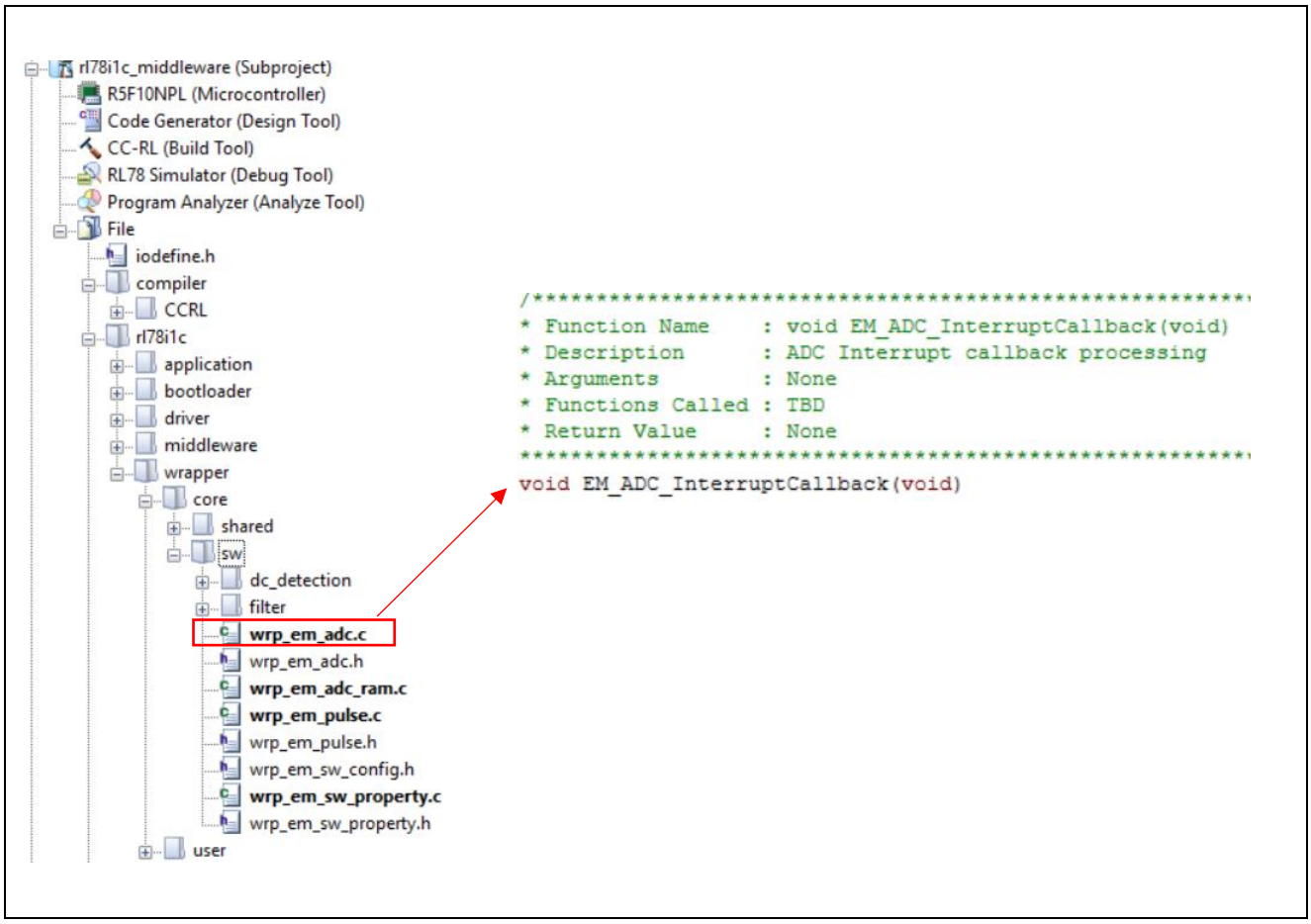


Figure 4-19 Signal accumulation path in ROM wrapper layer file

4.5.2. Signal sampling path in RAM

During bank swap operation, DSAD ISR run form RAM ISR. Due to the performance of RL78 code execution from RAM, which will take 4 times longer than execution from ROM, we only perform partial accumulation for metrology only (active energy, reactive energy and apparent energy) as shown in figure 4-20.

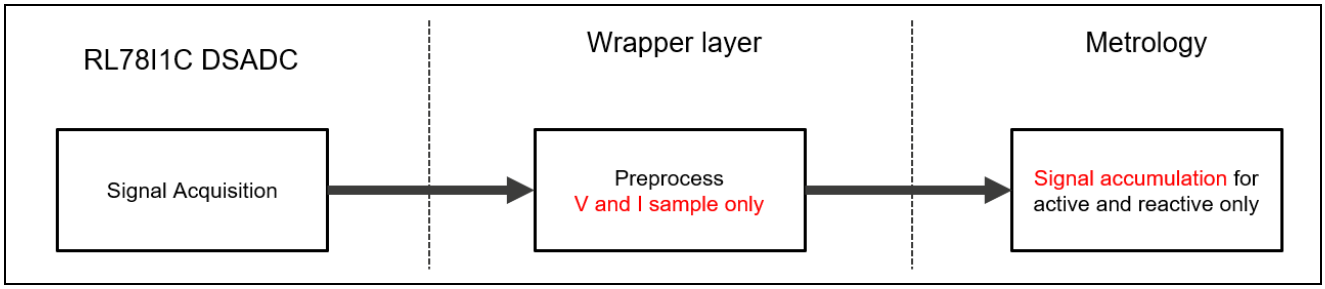


Figure 4-20 Signal accumulation path in RAM

The source code for signal sampling path in ROM in **wrp_em_adc_ram.c** file, **EM_ADC_RAM_InterruptCallback** in rl78i1c_middleware subproject.

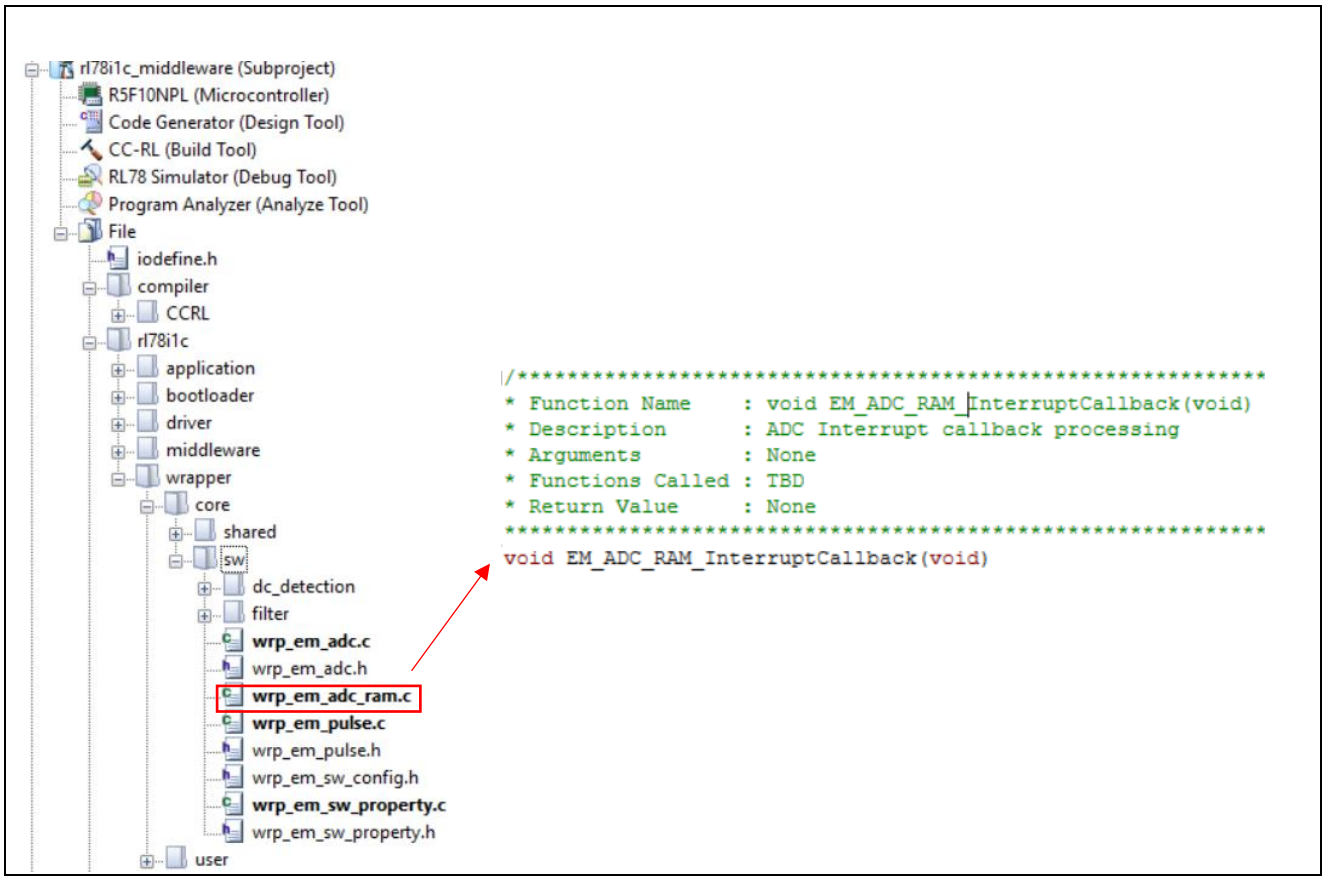


Figure 4-21 Signal accumulation path in RAM wrapper layer file

5. FOTA API functions

Following table list all functions of FOTA.

Function	Explanation
COMMAND_PollingProcessing	Processes received UART commands
COMMAND_InvokeInvertBootFlagAndReset	Command to invoke Fast FOTA
BL_FLASH_Prepare	Call flash prepare functions
FSL_InvertBootFlag	Invert boot flag and swap the bank
FSL_ForceReset	Reset the meter
COMMAND_InvokeBankSwap	Command to invoke Continuous Metrology FOTA
EM_GetEnergyCounter	EM User API. Get the accumulating energy counter
STORAGE_EM_SetEnergyData	Set to metrology counter in storage EEPROM
EM_RunOnRam_NonStopBankSwap	Continuous Metrology firmware update
EM_RunOnRam_PrepareFunctions	Prepare to run wrapper and metrology on RAM (Copy code from ROM to RAM)
EM_RunOnRam_DisableInterruptsExceptMetrology	Mask off all other interrupt except metrology
BL_RunOnRam_PrepareFunctions	Prepare to run wrapper and metrology on RAM
BL_FLASH_RAM_SwapBankWithRamIsr	Swap active boot cluster with running interrupt service routine on RAM
FSL_ChangeInterruptTable	Change vector table to RAM ISR
FSL_SwapActiveBootCluster	Swap the bank
FSL_RestoreInterruptTable	Restore vector table to ROM ISR
BL_FLASH_RAM_JumpBankSwapEntry	Call bank swap entry function
g_bl_properties.entry_func.bl_bswap	Jump to bootloader bankswap entry function

5.1. Usage of Fast FOTA API on Meter

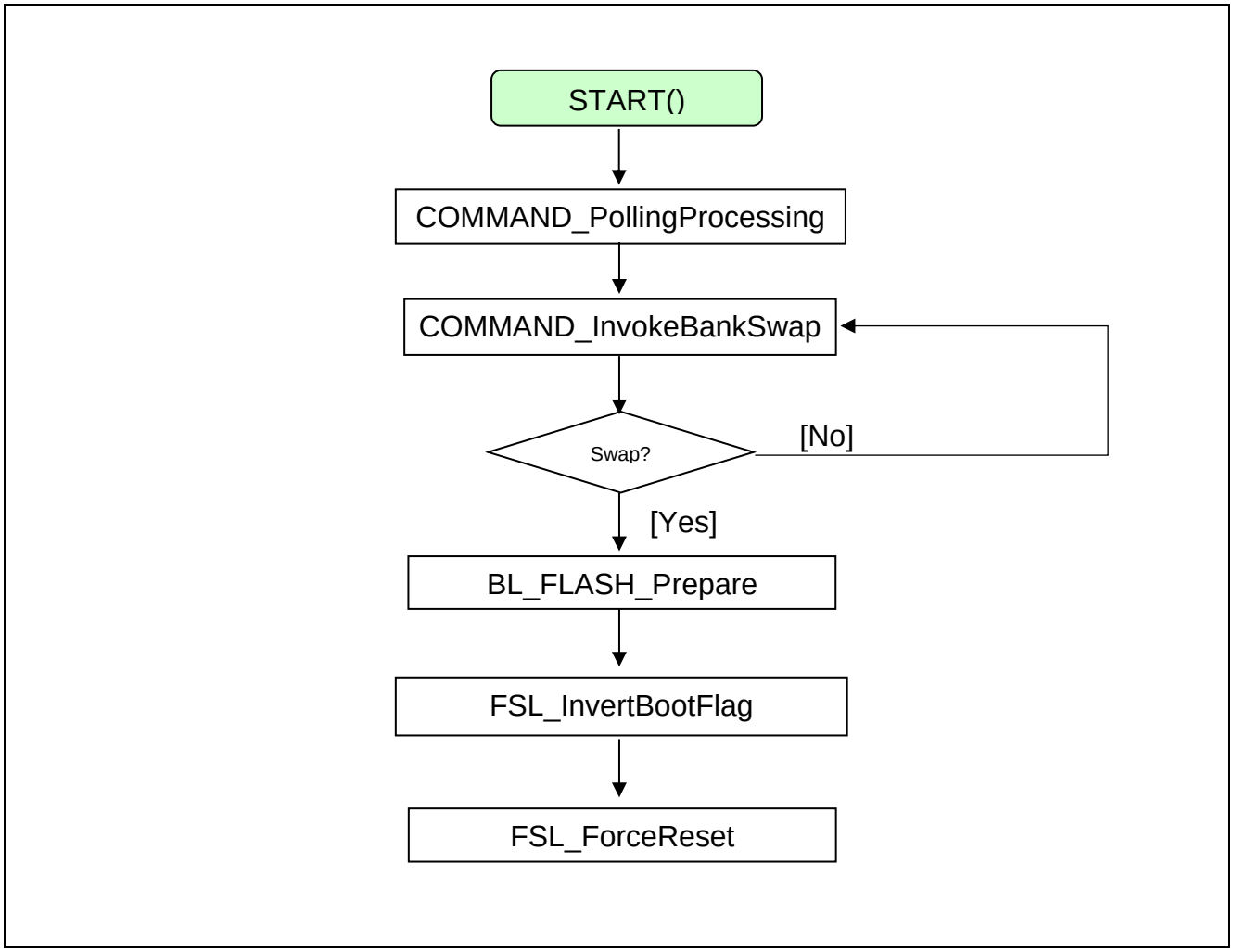


Figure 5-1 Usage of Fast FOTA API

5.2. Usage of Continuous Metrology FOTA API on Meter

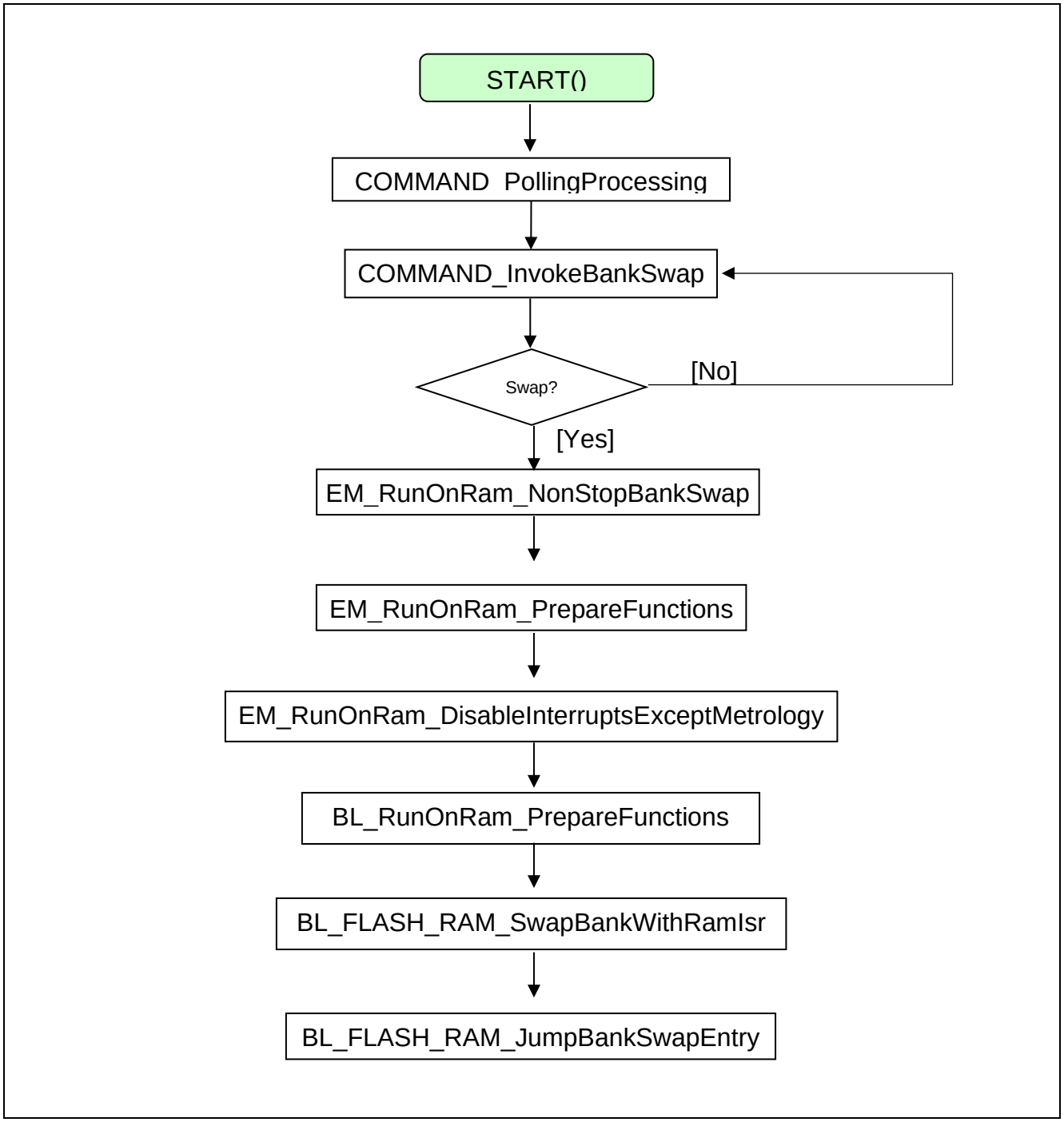
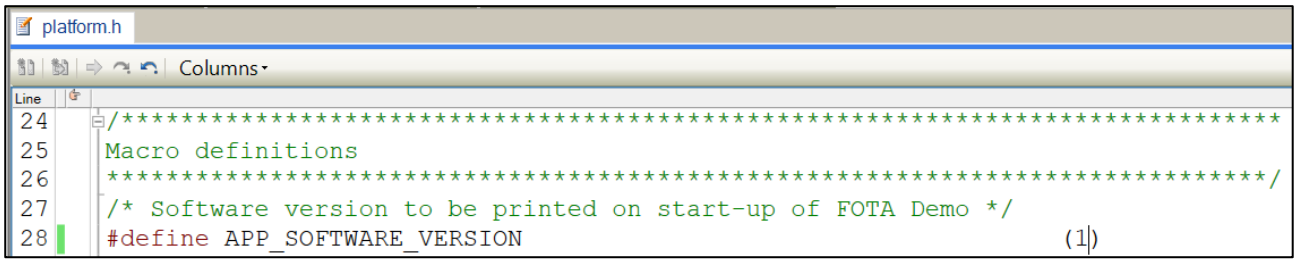


Figure 5-2 Usage of Continuous Metrology FOTA API

6. FOTA Demo Sample Firmware Build

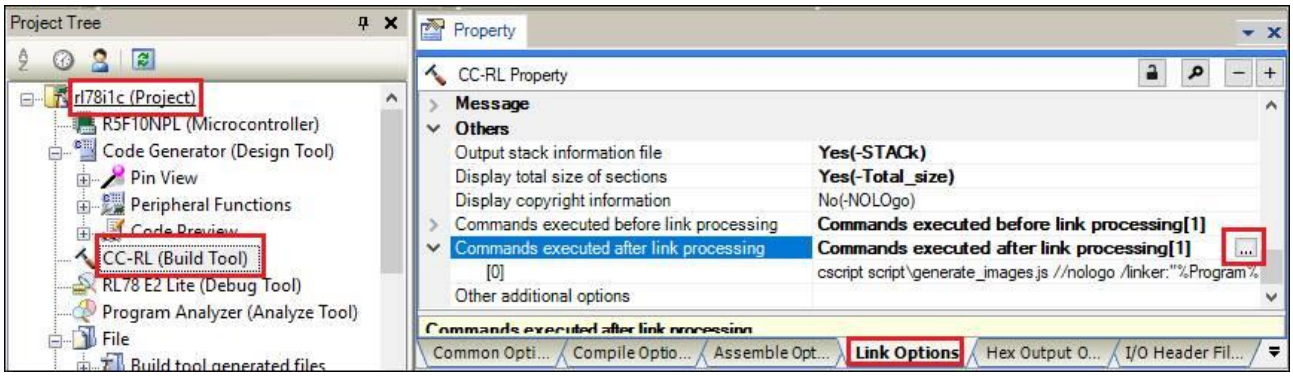
This chapter describes how to create the MOT files, “rl78i1c_production.mot”, “rl78i1c0 v001.mot”, “rl78i1c0 v002.mot”, and “rl78i1c0 v003.mot”, for r01an5860es0100-rl78i1c512fota-sample-project.zip.

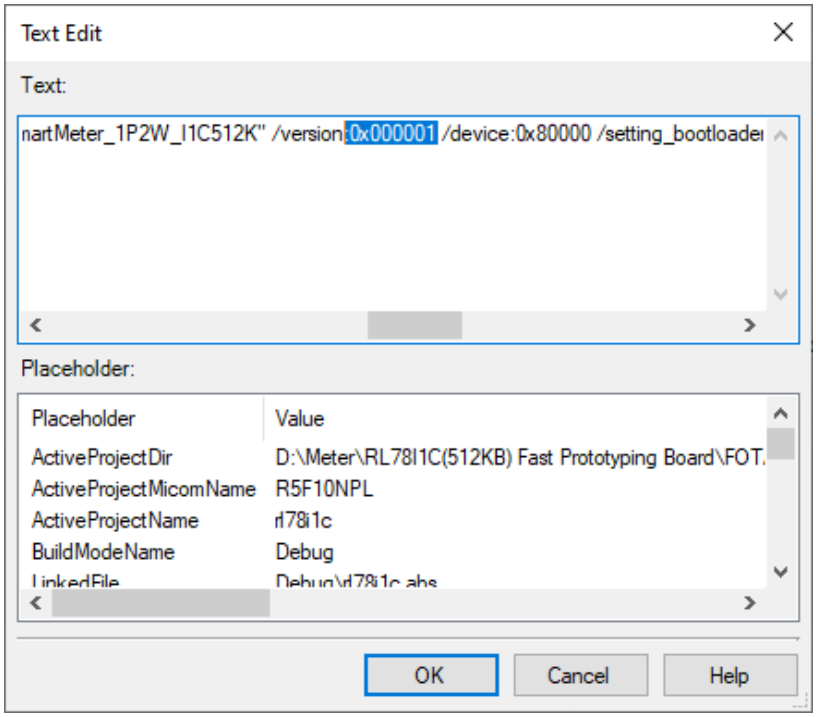
The Application Version number displayed on start-up of the FOTA Demo software is controlled by the macro definition **APP_SOFTWARE_VERSION** stored in [**platform.h**].



This allows for changes in Application Version number to affect the hash value of the User Application, as the source code of the User Application has changed.

The image header for the Primary and Secondary banks is generated post-build by the script [**generate_images.js**]. The input parameters to the script are stored within the CS+ project Build Tool settings.





To generate [**rl78i1c_production.mot**], set the macro definition **APP_SOFTWARE_VERSION** to (**1**), and the script parameter **version** to **0x000001**.

To generate the new software image to be transferred by FOTA (e.g. version 0.0.2), set the macro definition **APP_SOFTWARE_VERSION** to (**2**), and the script parameter **version** to **0x000002**.

The [**rl78i1c0.mot**] is generated in the [**\Source\CS+_CCRL\Debug\Image**] folder on building the project. After that, rename [**rl78i1c0.mot**] to [**rl78i1c0 v002.mot**].

Similarly, make the other version of the software image.

7. Diving Deeper

1. To learn more about the RL78/I1C (512KB) Fast Prototyping Board, refer to the RL78/I1C (512KB) User's Manual available in the User Guides & Manuals of the RL78/I1C webpage at renesas.com/br/en/products/microcontrollers-microprocessors/rl78-low-power-8-16-bit-mcus/rl78i1c-ultra-low-power-microcontrollers-high-end-smart-electricity-meter-market
2. Renesas provides several example projects that demonstrate different capabilities of the RL78/I1C (512KB) Fast Prototyping Board. These example projects can serve as a good starting point for users to develop custom applications. Example projects (source code and project files) are available in the RL78/I1C (512KB) Fast Prototyping Board Example Project Bundle.

8. Website and Support

Visit the following URLs to learn about the kit and the RA family of microcontrollers, download tools and documentation, and get support.

- RL78/I1C Resource renesas.com/br/en/products/microcontrollers-microprocessors/rl78-low-power-8-16-bit-mcus/rl78i1c-ultra-low-power-microcontrollers-high-end-smart-electricity-meter-market
- RL78 Product Information renesas.com/br/en/products/microcontrollers-microprocessors/rl78-low-power-8-16-bit-mcus
- RL78 Knowledge Base en-support.renesas.com/knowledgeBase#31025
- Renesas Support en-support.renesas.com/dashboard

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	July 9, 2021	−	Initial release

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: www.renesas.com/contact/.