Introduction:

This note addresses an issue where a user wants to do OTA updates to 2 separate regions of memory that are not juxtaposed. In particular, the user needs to update a program running in Flash and a data structure residing in QSPI.

MCUboot is employed to handle image verification and swapping/reverting images. MCUboot treats an image as contiguous from its start address to length. Since the program is at 0x00000000 and the data structure starts at 0x60000000 that’s over 4 gigs!

The solution proposed here is to call boot\_go(&fap) twice before vectoring to the application. The original const flash\_map[] structure is replaced with a ram-based structure, and a function is defined to fill it. First, with the flash map for the application and then with the flash map for the “resource” ie. the data structure.

Samples of applications are created that link the data structure in the code. Therefore, the linked image has both the application, down in the 0x000000ish area, and the data structure mapped at 0x60000000 region. Post build batch files use srec\_cat to split the image into its application (ap) and resources (rs) components. Then each is signed and the two are concatenated for 1 upload image with a size of ap\_size and rs\_size.



Procedure:

Required equipment:

* EK-RA6M4 (or EK-RA4M3 and ad-lib filenames)
* A PMOD UART/USB (or suffer with the virtual console)

A blue circuit board with black and white chips

AI-generated content may be incorrect.

* Collateral here: <https://github.com/daledrinkard/mcuboot_ms>
* srec\_cat utilities: <https://srecord.sourceforge.net/>
* jlink utilities: <https://www.segger.com/downloads/jlink/>
* Have python installed and setup according to the MCUboot ap note.
* FSP 5.8 e2 2025-01

System checkout:

Download the collateral. Open e2studio and choose Rename and import existing project into workspace.

A screenshot of a computer program

AI-generated content may be incorrect.

Navigate to the mcuboot\_ms/lab/lab\_resources folder and select the **MCUboot\_Encryption\_QSPI\_Solution.zip** archive. Name the project RA6\_boot and select the ra\_mcuboot\_ra6m4\_swap\_enc\_qspi project.

A screenshot of a computer

AI-generated content may be incorrect.

Import RA6\_boot\_scripts and tools projects from **mcuboot\_ms/e2\_projects/mcuboot\_ms\_x1\_042425.zip**

A screenshot of a computer

AI-generated content may be incorrect.

Right-click on the RA6\_boot project that was imported and open a command prompt.

Enter srec\_cat -version and verify it is installed and the path is correct

A computer screen with white text

AI-generated content may be incorrect.

Type jlink --version and verify it is installed,

A black background with white text

AI-generated content may be incorrect.

Type python --version

A black background with white text

AI-generated content may be incorrect.

Connect the EK debug cable. Right-click on the tools project and select System Explorer

A screenshot of a computer

AI-generated content may be incorrect.

Double-click on the erase\_qspi\_all.bat to run it.

A screenshot of a computer

AI-generated content may be incorrect.

This will take more than 2 minutes to complete and will erase the entire contents of the QSPI. ^C to break or press the spacebar to continue. If you’re not sure the QSPI is erased press the spacebar to erase it.

A progress dialog will pop up if all is working correctly.

A screenshot of a computer

AI-generated content may be incorrect.

Modify RA6\_boot.

Remove encryption by changing the Encryption scheme in the MCUboot properties.

A screenshot of a computer

AI-generated content may be incorrect.

Change the debugger output for MCUboot by edting the properties of the MCUboot logging module.

A screenshot of a computer

AI-generated content may be incorrect.

Logging calls printf. Implement printf.

There are 2 choices for implementing printf: virtual console or SCI. The virtual console does not require any additional hardware but is limited in function and ONLY works while debugging. SCI9 printf is much preferred and works whether in debug or not.

If using virtual debugger:

* Set the linker misc other linker flags to **--specs=rdimon.specsA screenshot of a computer

  AI-generated content may be incorrect.**
* Add this declaration in hal\_entry.c: **extern void initialise\_monitor\_handles(void);**
* Call initialise\_monitor\_handles(); at the top of hal\_entry(void) function.
* Add printf(“hal\_entry\n”); after the initialize as a test.

If using a pmod adapter and SCI 9:

* In the pin configurations tab, change the DEBUG0 pinning to this: A screenshot of a computer

  AI-generated content may be incorrect.
* In the pin configurations tab, change SCI9 to this: A screenshot of a computer

  AI-generated content may be incorrect.
* Copy mcuboot\_ms/lab/lab\_resources/printf\_redirect folder to the RA6\_boot/src folder
* Add a SCI\_uart module and configure it like this: A screenshot of a computer

  AI-generated content may be incorrect.
* Set the linker misc option to --specs=nosys.specs: A screenshot of a computer

  AI-generated content may be incorrect.
* Remove call to initialise\_monitor\_handles if virtual console was used previously.

Import the tools and RA6\_boot\_scripts projects from the archive mcuboot\_ms\e2\_projects\mcuboot\_ms\_x1\_042425.zip

A screenshot of a computer

AI-generated content may be incorrect.