Application Note

Adesto

Application Note AN106

Using EcoXiP on the NXP i.MX RT1050 EVKB Board

www.adestotech.com

This publication contains proprietary information which is subject to change without notice and is supplied “as is”, without any warranty of any kind.

Revision History

Revison Number	Date	Tasks
A0	September 06, 2018	i.MX RT1050 EcoXiP Flash Memory Device Application Note initial release
A1	June 18, 2019	i.MX RT1050 EcoXiP Flash Memory Device Application Note reformatting

Adesto

3600 Peterson Way, Santa Clara, CA 95054 USA | Phone: +1 408 400 0578 | www.adestotech.com | e-mail: contact@adestotech.com

Table of Contents

3600 Peterson Way, Santa Clara, CA 95054 USA | Phone: +1 408 400 0578 | www.adestotech.com | e-mail: contact@adestotech.com

1. Introduction

This document discusses the hardware and software requirements for configuring the NXP i.MX RT1050 EVKB board with an Adesto EcoXiP Flash device. An overview of hardware modifications is provided in Section 2, along with a more detailed description of the necessary hardware changes in Appendix A. From a software perspective, Sections 3 - 5 contain information on configuring the EcoXiP Flash, loading the code into the Flash using a Flash loader from various tool chains (three are provided), and executing the code. The Adesto EcoXiP Flash device conforms to the new JEDEC xSPI1 and JESD251 specifications and is the only Octal-SPI device designed specifically for XiP, providing better cost and power savings than its competitors. Figure 1 shows a photo of a modified i.MX RT1050 EVKB board with an Adesto EcoXiP Flash device.

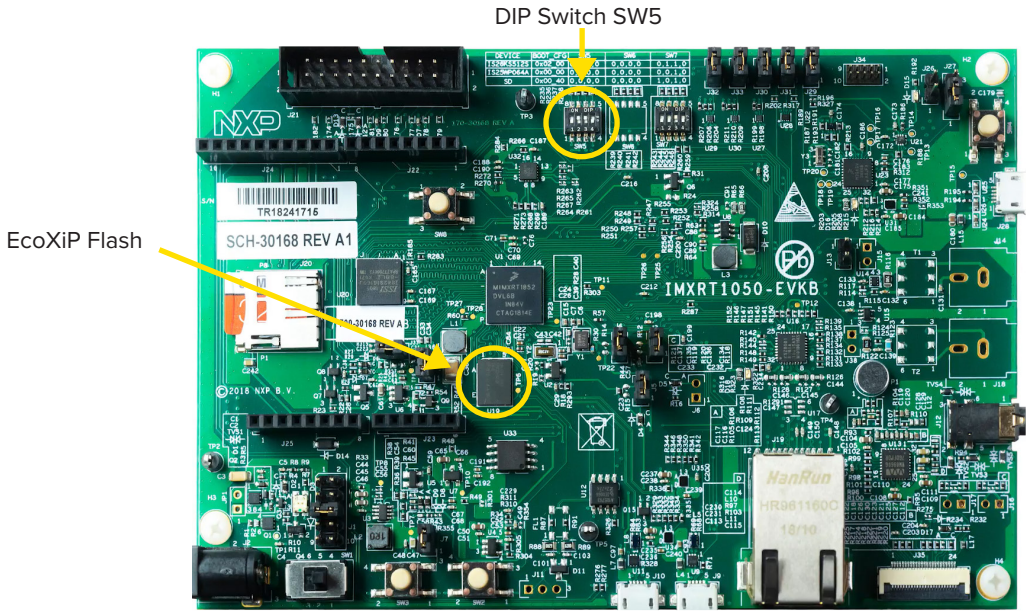

Figure 1. NXP i.MX RT1050 EVKB Board with EcoXiP Flash

The NXP i.MX RT1050 crossover processor family has been designed without embedded Flash. Rather, it relies on external Flash devices such as Adesto's EcoXiP to store code and data. Its highly optimized Flash host controller (called FlexSPI) enables direct execution of code from external Flash. This architecture, known as execute-in-place or XiP, provides multiple benefits in terms of cost and power consumption and works well with advanced serial Flash devices such as EcoXiP which supports Octal-SPI interface, double data rate (DDR) and high clock speeds to support high-speed execution.

The i.MX RT1050 can be configured to boot directly from external Flash. Generating a Flash-bootable image requires users to include a couple of data blocks at the top of the image, one of which is product-dependent (includes parameters which vary from Flash product to Flash product). The i.MX RT1050 Software Development Kit (SDK) includes many example projects and now most example projects include one configuration or target in which the system boots and executes from external Flash.

Once a bootable image is built, it must be downloaded to the target and in the case of Flash memory be programmed into the Flash device. During development, programming can be done directly from the development tools environment, specifically from the debugger. Development tools often support Flash loader extensions (sometimes called Flash drivers or Flash algorithms) to enable that. Typically each Flash manufacturer requires its own extension or plug-in for programming. This is true for the Adesto EcoXiP Flash device.

Adesto

3600 Peterson Way, Santa Clara, CA 95054 USA | Phone: +1 408 400 0578 | www.adestotech.com | e-mail: contact@adestotech.com

This application note examines the process of building a Flash-bootable image for i.MX RT1050, programming the Flash, and running it. It specifically shows how to do all of this using the EcoXiP Flash. This application note refers to the NXP MCUExpresso SDK version 2.4.0.

2. Hardware Modifications

This section provides an overview of the hardware modifications performed on the NXP evaluation board to facilitate its use with the Adesto EcoXiP Flash memory device. If you have already received a modified IMXRT1050-EVKB board from Adesto, this section is meant to provide a brief overview of the hardware modifications performed. If you received an original unmodified IMXRT1050-EVKB board and need to perform the required hardware modifications, or if you are just interested in understanding how the original board was modified, refer to Appendix A.

A modified version of the NXP evaluation board for the i.MX RT1050 uses an EcoXiP Flash. The following modifications were applied to an original IMXRT1050-EVKB board:

- Replace original Flash device in U19 with the Adesto EcoXiP Flash device.
- Populate a 4-position DIP switch in SW5 (originally unpopulated). Set switches 2 and 3 to the ON position to extend RT1050 boot ROM hold time (indicates the amount of time delay from a Flash reset to the first Flash access). As an alternative to populating the DIP switch, wires can be soldered to jumper pins 2 and 7 and pins 3 and 6 on SW5. This would accomplish the same amount of delay as described in Appendix A.1.1, Enable Extension of Reset Hold Time Using Switch SW5.
- Set the i.MX RT1050 eFuse which enables the boot ROM to reset the Flash device each time the i.MX RT1050 is reset.

3. Boot / XiP Settings

In order to boot from the EcoXiP device, it is necessary to include a suitable Serial NOR Configuration Block in the image.

The Serial NOR Configuration Block occupies the first 512 bytes of the Flash memory. It provides information to the i.MX RT1050 boot ROM on how to configure the FlexSPI host controller and the Flash device with desired parameters for optimal operation. The boot ROM initially reads from the Flash using a universal read command (opcode 03h) at a low speed. Once it reads the Serial NOR Configuration Block and reconfigures the system accordingly, it jumps to the application program entry point. At that point the system starts executing code from Flash. For more information, refer to the *i.MX RT1050 Processor Reference Manua*l, Chapter 8 (System Boot).

The i.MX RT1050 SDK supports this execute-in-place (XiP) mode. Many example projects have one configuration which supports this mode. For example, users of the IAR tools will find in many cases a project configuration called ‘Debug’ which builds the code to run from TCM (internal SRAM) and next to it another configuration called ‘flexspi_nor_debug’ which builds the code to run from Flash in XiP mode. That project configuration uses a C file named evkbimxrt1050_flexspi_nor_config.c to define the Serial NOR Configuration Block. This file can be found in the ‘xip’ source group of the project. Once compiled, the file is linked to offset 0 in the flash memory as this is the location where the boot ROM expects the Serial NOR Configuration Block to be located.

The version of *evkbimxrt1050_flexspi_nor_config.c* which comes with i.MX RT1050 SDK is not suitable for EcoXiP. As such, Adesto provides a version of this file which works with EcoXiP. This file is attached to this application note. Note that the .c file name should not be changed. When the project is rebuilt, a bootable image is created that allows the i.MX RT1050 processor to boot from the EcoXiP Flash device. Please note that this source file works for all supported tool chains.

Adesto

3600 Peterson Way, Santa Clara, CA 95054 USA | Phone: +1 408 400 0578 | www.adestotech.com | e-mail: contact@adestotech.com

After replacing the *evkbimxrt1050_flexspi_nor_config.c* file in the XiP folder, open your project with your development tool, select the project configuration designed for XiP and build the project to produce a bootable image. Refer to Section 5 for more information on how to load the bootable image and run it.

Try generating an EcoXiP-bootable image for the 'hello_world' example of the SDK (the "hello world" project can be found in the following path under the SDK root folder: *boards\evkbimxrt1050\demo_apps\hello_world*).

The following screen shot shows the hello_world example using the IAR Workbench.

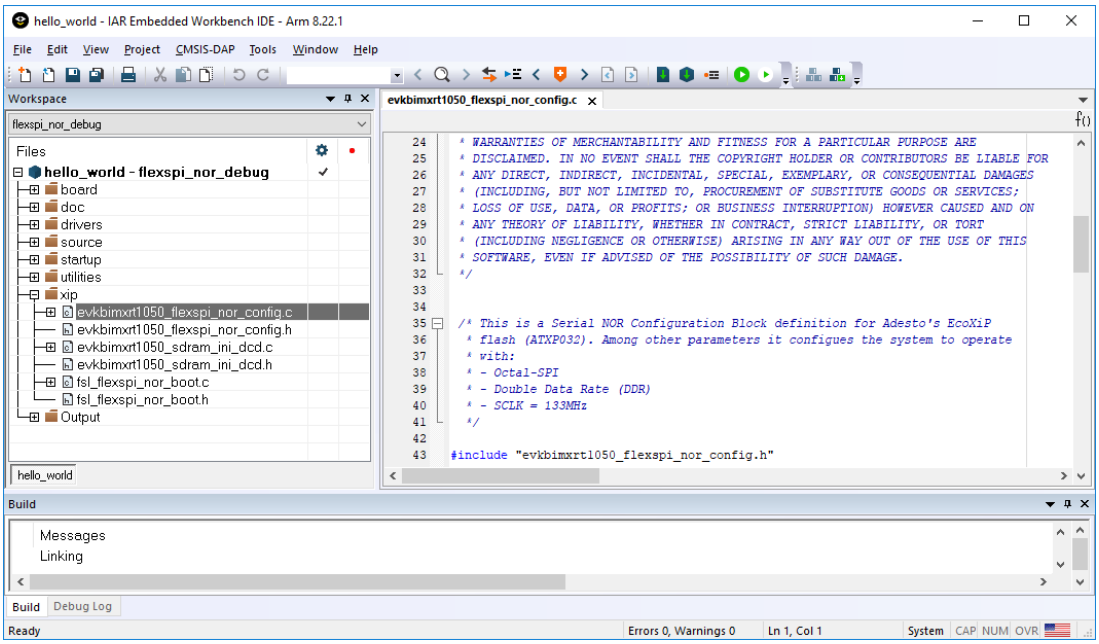

Figure 2. Hello World Example - IAR Workbench

4. Loading Code into the EcoXiP Flash

Once a bootable image has been created it has to be loaded into Flash memory. During the development phase this can be done as part of loading a program to target memory from the debugger. The debugger typically has a Flash loader extension or plug-in which can program an image into Flash memory in case the program's image (or part of it) is linked to the Flash memory address range. Different Flash products require different Flash loaders. The user has to let the debugger know which Flash loader to use based on the target Flash memory being used in the system.

Flash loaders for EcoXiP are available for the following major tool chains:
•IAR Embedded WorkBench — Version 8.30.1 and up
•Keil MDK — Version 5.25.2 and up
•MCUXpresso IDE — Version 10.2.0 and up

The configuration required to support each tool chain is described in the following subsections.

Adesto

3600 Peterson Way, Santa Clara, CA 95054 USA | Phone: +1 408 400 0578 | www.adestotech.com | e-mail: contact@adestotech.com

4.1 IAR Embedded Workbench

The IAR debugger has a minor issue related to verification after Flash loading, but this issue can easily be resolved using the procedure below. This procedure initially shows how to set up an EcoXiP Flash loader without verification, and then how to enable verification.

In the IAR project option go to Debugger and select the Download tab. Check “Use flash loader(s)”, then check “Override default. board file”. In the box underneath enter the following:

```
$TOOLKIT_DIR$\config\flashloader\NXP\FlashIMXRT1050_EVK_FlexSPI_EcoXiP.board
```

You can also browse to find this file using the 3-dot small box to the right of the text box as shown in the following screen shot.

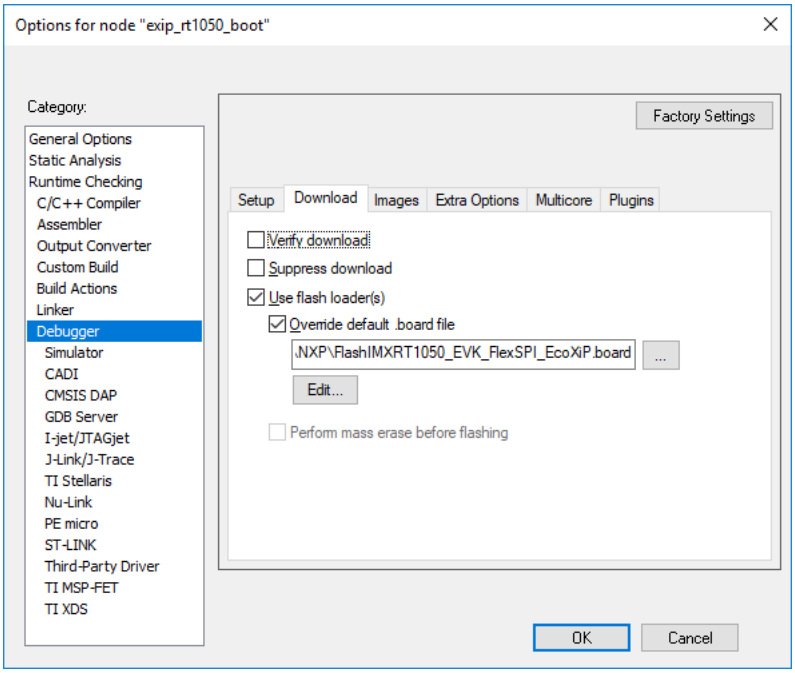

Figure 3. IAR Embedded Workbench Debugger

The above should work without the “Verify download” option (as you can see above it’s not checked so far).

To enable verification, one has to use one of the following two options to work around the aforementioned IAR issue.

Option 1:
Enable a macro file to be called after reset. This file re-initializex the FlexSPI host controller and a couple of other things in the MCU. To apply, first copy the attached *EcoXiP_init.mac file* to the folder where the IAR project is located (same folder where *.eww* and *.ewp* files are located). Then in the *IAR project* option, go to Debugger and select the Setup tab. Check “Use macro file(s)”. In the box enter *$PROJ_DIR$\EcoXiP_init.mac* as shown in the following screen shot.

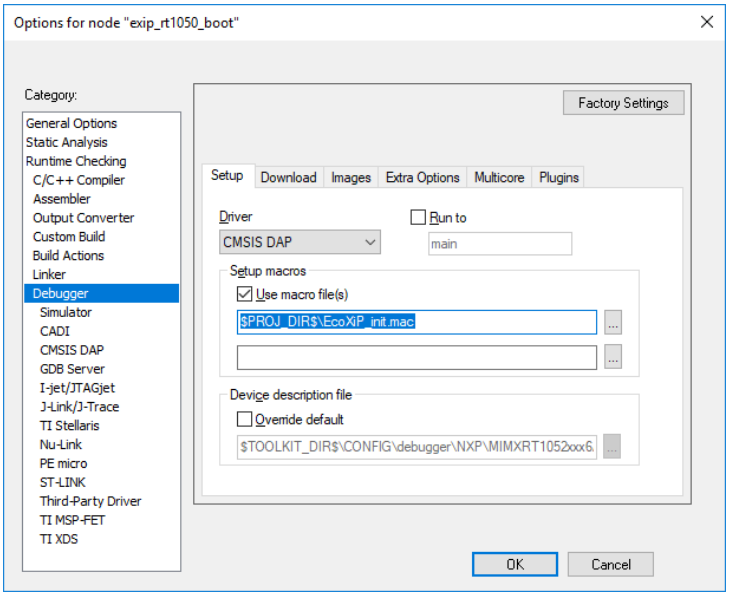

Figure 4. IAR Embedded Workbench Verification Enable - Option 1

Option 2: Change the reset style so that the debugger will not reset the MCU before verification. To apply this option go to *CMSIS DAP* in the *IAR project* option. Click on the *Setup* tab and in the Reset pulldown menu select “*Software*” as shown in the screen shot below.

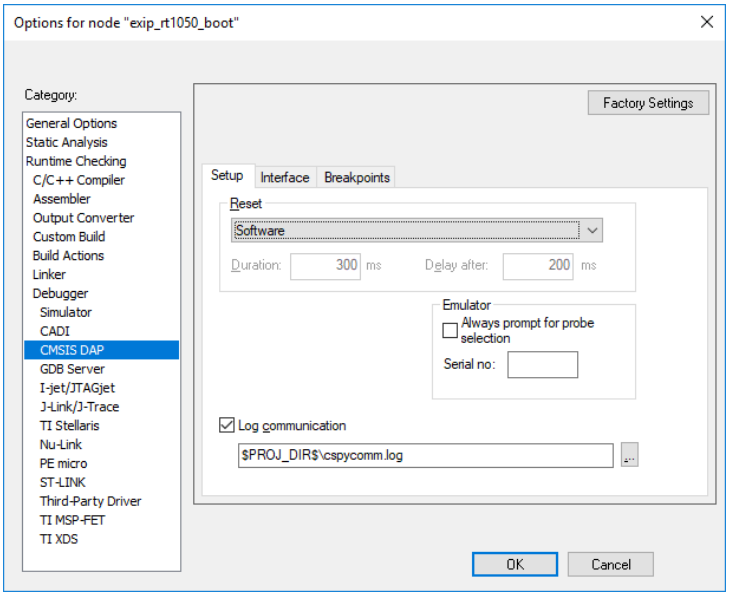

Figure 5. IAR Embedded Workbench Verification Enable - Option 2

3600 Peterson Way, Santa Clara, CA 95054 USA | Phone: +1 408 400 0578 | www.adestotech.com | e-mail: contact@adestotech.com

After applying the settings according to one of the options described above, check "*Verify download*" as shown in Figure3 above. Now you can use the EcoXiP Flash loader with verification.

To load the program image, click the *Download* and *Debug* buttons.

4.2 Keil MDK

Note: If you have MDK version 5.26 or higher, continue to the next paragraph, otherwise carefully read the remainder of this paragraph. At the time this document was released, the latest MDK version was 5.25. This version includes an EcoXiP Flash loader (algorithm in the MDK terminology) which supports i.MX RT1050 rev A0. Due to an incompatible change done by NXP in i.MX RT1050 rev A1, this MCU revision requires a small change in the Flash loader. The EcoXiP Flash loader which supports i.MX RT1050 rev A1 is available and will be released in MDK version 5.26. In the meantime, if you have MDK version 5.25 you can replace the Flash loader with the one provided with this application note. Simply copy the file *MIMXRT105x_ECOXIP_4MB_SEC.FLM* to the ARM\Flash folder under the MDK installation root folder, overwriting the original file.

In the Keil MDK project options, select the *Debug* tab, then click on the *Settings* button on the far right as shown in the following screen shot.

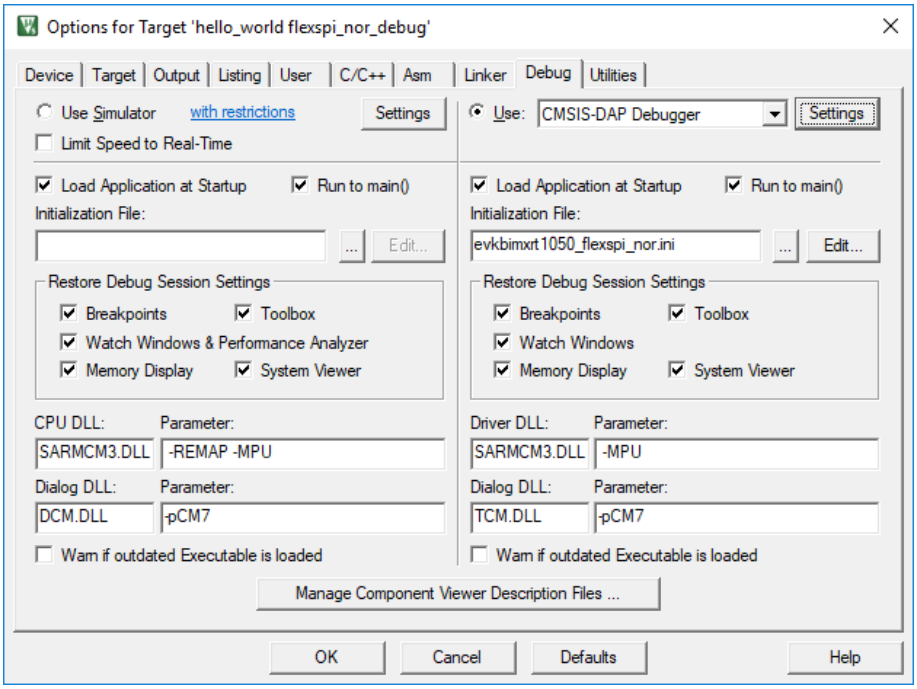

Figure 6. Keil MDK - Project Options

When the *Driver Setup* window pops up, select the *Flash Download* tab as shown in the following screen shot.

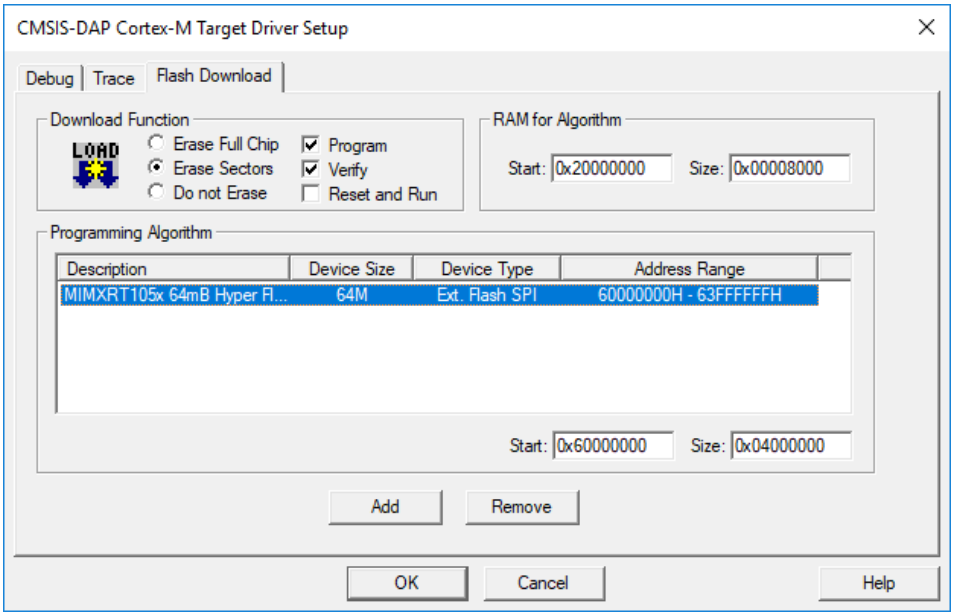

Figure 7. Keil MDK - Driver Setup

In the *Programming Algorithm* list, if there is already an existing algorithm for HyperFlash, click on *Remove* to remove it. Now click *Add*. A window with a list of Flash algorithms pops up. Select the algorithm called "*MIMXRT105x EcoXiP Flash*" and click *Add* as shown in the following screen shot.

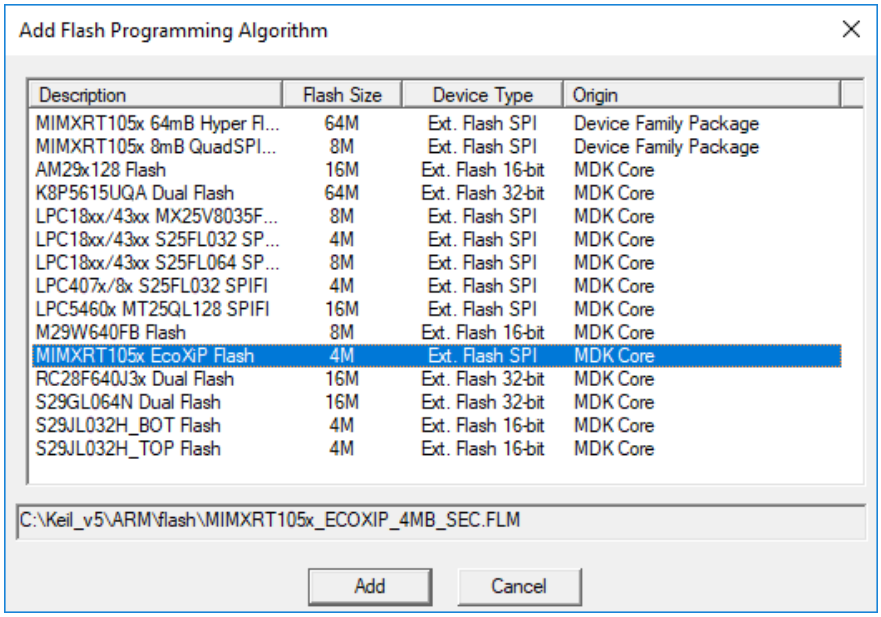

Figure 8. Keil MDK - List of Flash Programming Algorithms

Adesto

3600 Peterson Way, Santa Clara, CA 95054 USA | Phone: +1 408 400 0578 | www.adestotech.com | e-mail: contact@adestotech.com

Select the EcoXiP Flash algorithm (only one visible) and click OK as shown in the following screen shot.

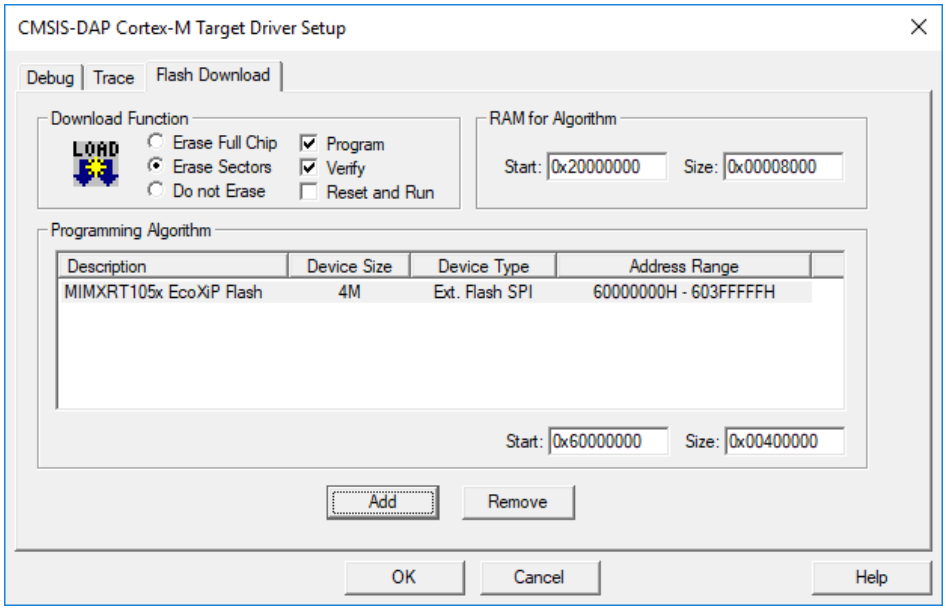

Figure 9. Keil MDK - Select EcoXiP Flask

The program can now be loaded into the EcoXiP Flash. To load the image, click on the *Load* button.

4.3 MCUXpresso IDE

When importing the SDK example into the MCUXpresso IDE, an Advanced Settings window is displayed as shown in Figure 10 that includes a memory configuration box at the bottom. This is where you can set up the EcoXiP Flash driver.

Note: If the example project has already been imported you can still modify the memory configuration by opening the project properties dialog box and under C/C++ Build by selecting 'MCU Settings'.

Adesto

3600 Peterson Way, Santa Clara, CA 95054 USA | Phone: +1 408 400 0578 | www.adestotech.com | e-mail: contact@adestotech.com

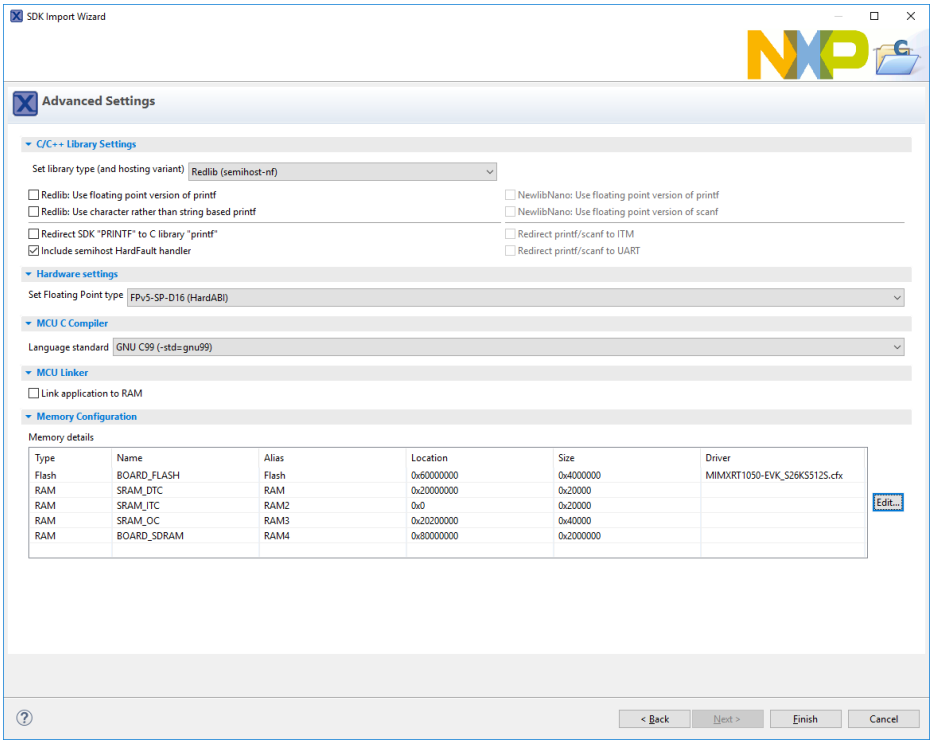

Figure 10. Advanced Settings Display Window

Click on the *Edit* button on the right side of the *Memory Details* box in Figure 10. The following screen shot is displayed.

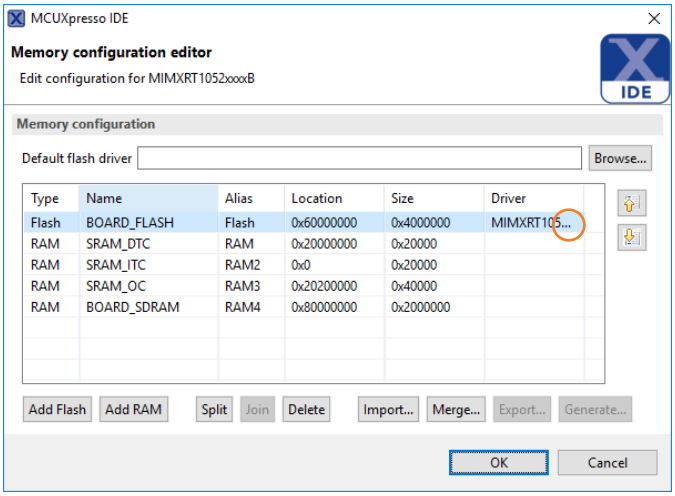

Figure 11. MCUExpresso IDE Memory Configuration Editor

There will be one Flash entry listed. On the right hand side in the Driver column above, double-click on the 3 dots to the right of the driver's name as noted by the orange circle in Figure 11. The following screen shot is displayed.

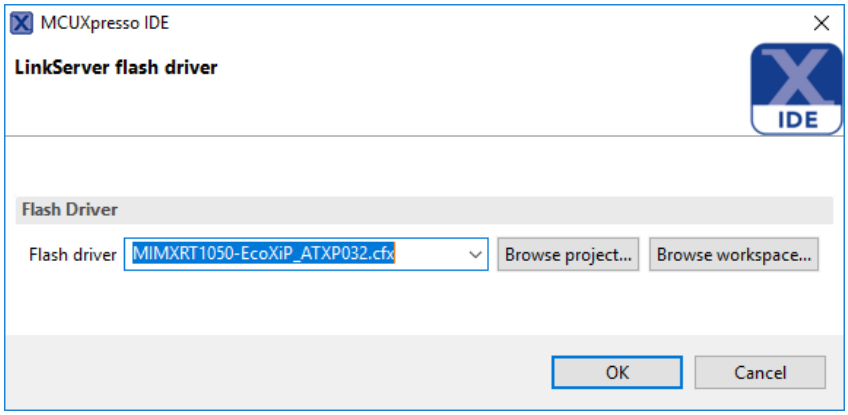

Figure 12. MCUExpresso IDE LinkServer Flash Driver

From the Flash Driver pull-down menu, select the EcoXiP Flash driver: *MIMXRT1050-EcoXiP-ATXP032.cfx*, as shown in the above screen shot. Click *OK* to return to the Memory Configuration Editor. Click *OK* to exit the editor.

The debugger is now ready to load the program. To load the image, click on the *Debug* button.

5. Boot / Run

After the program image has been loaded into Flash memory, the program can be executed in one of two ways:

- Stay in the debug session and run the application program from the debugger
- Exit the debug session and run the application program stand alone

To run the application program stand-alone, make sure that the board is set up as follows:

- On DIP switch SW7, set switch 3 to the ON position, and switches 1, 2, and 4 to the OFF position. This configures the boot ROM to boot from the EcoXiP Flash device.
- On DIP switch SW5, set switches 2 and 3 to the ON position, and switches 1 and 4 to the OFF position. If no DIP switch is installed in SW5, two wires must be placed, one connecting pads 2 and 7 and another connecting pads 3 and 6 (see Appendix A). This configures the hold time (the time between Flash reset and the first Flash access).

Once the board is set up correctly, either press the reset button (SW3) or power cycle the board. In both cases the application program automatically runs from the EcoXiP Flash.

In the case of the “hello world” example project the following message appears in the display window.

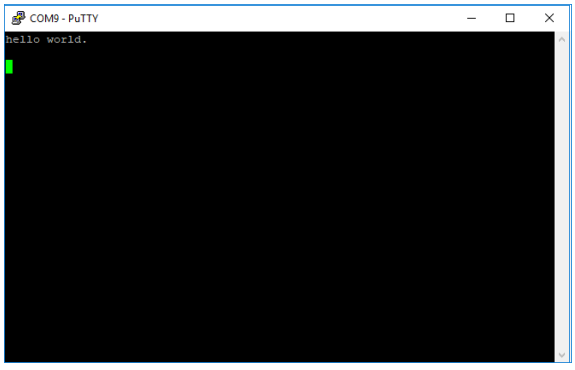

Figure 13. Hello World Display

3600 Peterson Way, Santa Clara, CA 95054 USA | Phone: +1 408 400 0578 | www.adestotech.com | e-mail: contact@adestotech.com

APPENDIX A. REWORK INSTRUCTIONS

The following steps are used to rework the NXP board to work with the Adesto EcoXiP Flash device.

A.1 Replace Flash Device

The first step is to replace the Flash component at U19 on the board (most likely a HyperFlash device) with the Adesto EcoXiP ATXP032 Flash.

Note that both devices come in a 24-pin BGA package.

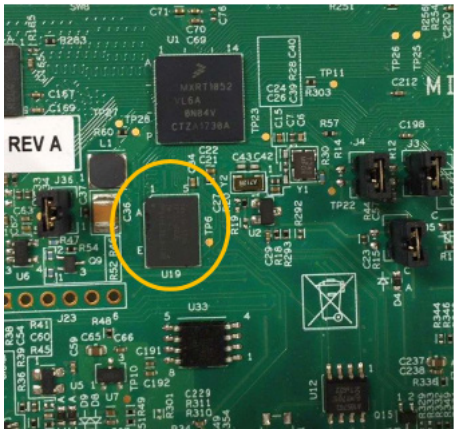

Figure 14. Replace U19 with Adesto EcoXiP Flash Device

A.1.1 Enable Extension of Reset Hold Time Using Switch SW5

When the board is shipped from NXP, location SW5 is not populated. However, for synchronization purposes, it is recommended that the EcoXiP Flash device be reset at the same time at the i.MX RT1050 processor. This requires modification to the pins associated with device SW5 on the board.

Pins 2 and 7, and pins 3 and 6 of SW5 must be connected to configure the appropriate amount of delay to allow the EcoXiP Flash to stabilize before being accessed by the i.MX RT1050 processor.

There are two methods that can be used to connect pins 2 and 7, and pins 3 and 6 at location SW5.

- Use wires to jumper pins 2 and 7, and pins 3 and 6 to manually configure the appropriate i.MX RT1050 processor pins to a logic HIGH.
- Install a 4-position DIP switch at location SW5 and set switches 2 and 3 to the ON position.

To solder wires between the DIP switch pins in order to manually set them to the ON position, refer to Section A.1.2.

To add a DIP switch to location SW5, refer to Section A.1.3.

Adesto

3600 Peterson Way, Santa Clara, CA 95054 USA | Phone: +1 408 400 0578 | www.adestotech.com | e-mail: contact@adestotech.com

A.1.2 Jumper Pins at Location SW5

To manually configure the appropriate processor pins to a logic HIGH, wires must be soldered between pins 2 and 7, and pins 3 and 6 of SW5 as shown in Figure 15.

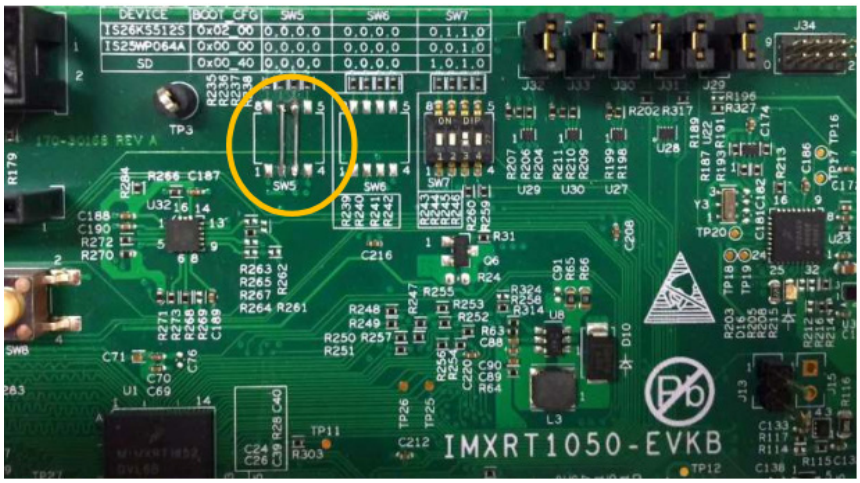

Figure 15. Using Wires to Jumper Select Pins at Location SW5

A.1.3 Add a DIP Switch at Location SW5

As an alterative to connecting wires between selected pins at location SW5 as described in the previous subsection, a DIP switch can be added, and then switches 2 and 3 set to the ON position.

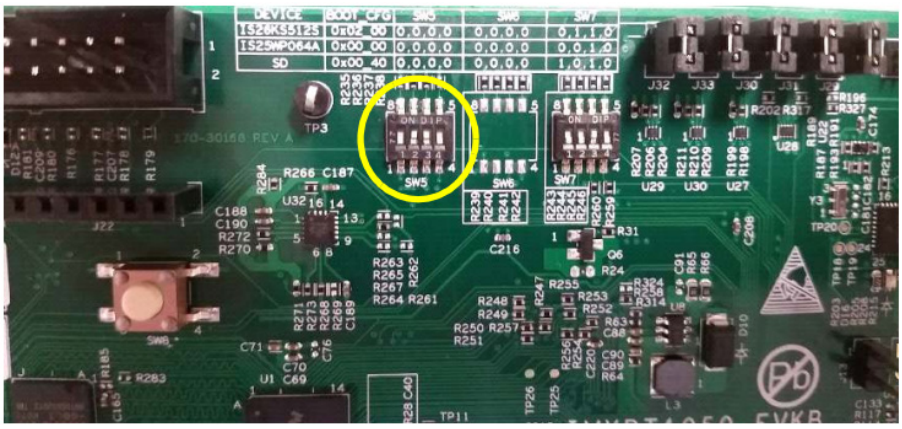

Figure 16. Adding a DIP Switch at Location SW5

Adesto

3600 Peterson Way, Santa Clara, CA 95054 USA | Phone: +1 408 400 0578 | www.adestotech.com | e-mail: contact@adestotech.com

A.2 JEDEC Reset and eFUSE Programming

To ensure that the i.MX RT1050 MCU and the EcoXiP Flash never lose synchronization with each other, it is best if the EcoXiP Flash is reset each time the MCU is reset. To achieve this, the i.MX RT1050 boot ROM option must be enabled to automatically apply a JEDEC reset to the EcoXiP Flash right after the MCU reset.

A.2.1 JEDEC Reset

A JEDEC reset is a sequence of signals sent by a host MCU to the Flash device which results in a reset event occurring on the Flash device. The sequence is transmitted by the host over two of the SPI interface pins. The EcoXiP Flash supports the JEDEC reset option as an alternative to applying a reset on its RESET input pin.

To enable an automatic JEDEC reset by the i.MX RT1050 boot ROM, it is necessary to set an eFuse of the i.MX RT1050, specifically bit 6 in the OTP register located at offset 0x6E0 in the eFuse array of the i.MX RT1050 processor.

A.2.2 Setting the eFuse

Setting an eFuse is a quick and easy procedure done with the help of NXP's MfgTool (manufacturing tool program). This tool comes as part of NXP's "Flashloader i.MX-RT1050" package which can be downloaded from here.

https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/i.mx-applications-processors/i.mx-rt-series/i.mx-rt1050-crossover-processor-with-arm-cortex-m7-core:i.MX-RT1050?&tab=Design_Tools_Tab

On the above NXP web page, scroll down to select *Flashloader i.MX_RT1050* and click *Download*.

Once the package is downloaded, the *MfgTool* program can be found in the following path under the package root folder: `Tools\mfgtools-rel`. The program's name is `MfgTool2.exe`. Before running the program, copy the attached `boot_image.sb` file to the following folder under the package root folder: `Tools\mfgtoolsrel\Profiles\MXRT105X\OS` Firmware. This is an input script for *MfgTool* and its only purpose is to set the above eFuse to 1.

Once the file is copied, connect power to the NXP I.MX RT1050 EVKB board to power as normal by connecting a USB cable from a PC host to location J28 on the NXP board. In addition, connect a USB cable from that PC host to location J9 on the NXP board (this is the communication channel of the MfgTool program). Set all DIP switches on SW7 to the OFF position, then push the SW3 button to reset the system.

Invoke the MfgTool program by double-clicking on the `MfgTool2.exe` file as shown below.

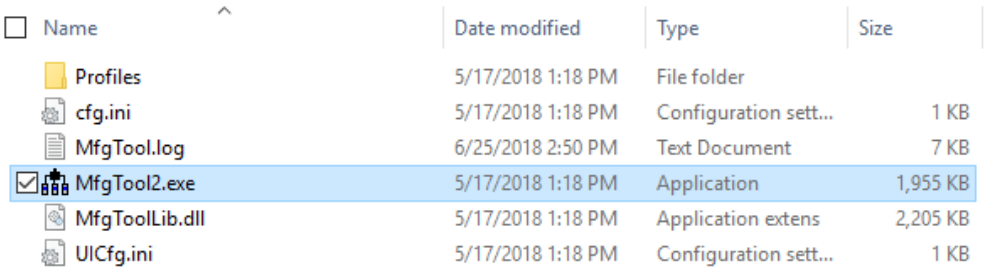

Figure 17. Invoke the MfgTool Program

The *MfgTool* window is displayed as shown below.

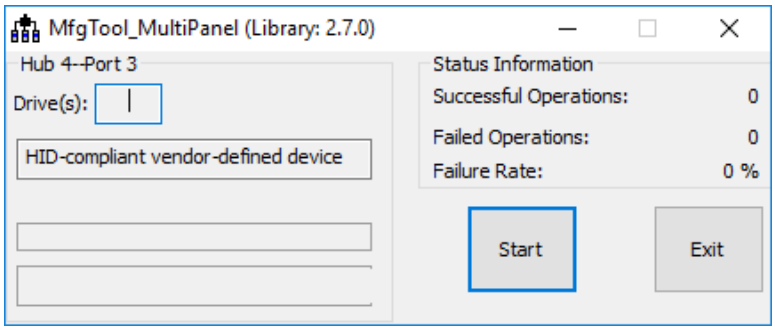

Figure 18. MfgTool Main Window

In the above screen shot, the message: *HID-compliant vendor-defined* device is displayed.

Click Start. The operation should complete instantly and green bars should be displayed at the bottom. Once these bars are visible, click *Stop* and then *Exit*. At this point the JEDEC Reset eFuse is set.

3600 Peterson Way, Santa Clara, CA 95054 USA | Phone: +1 408 400 0578 | www.adestotech.com | e-mail: contact@adestotech.com

Additional Information

Adesto Technologies Corporation (NASDAQ:IOTS) is a leading provider of innovative application-specific semiconductors and embedded systems for the IoT era. The company's technology is used by more than 2,000 customers worldwide who are creating differentiated solutions across industrial, consumer, medical and communications markets. With its growing portfolio of high-value technologies, Adesto is helping its customers usher in the era of the Internet of Things.

Visit: www.adestotech.com

3600 Peterson Way, Santa Clara, CA 95054 USA | Phone: +1 408 400 0578 | www.adestotech.com | e-mail: contact@adestotech.com

AN106A1_06/2019