AN13386 RT600 Flash Loader For Custom Flash Device Rev. 1.0 — 10 February 2025

Application note

Document information

Information	Content
Keywords	AN13386, i.MX RT600, RT685, RT600, MIMXRT685, boot, flash loader
Abstract	The RT600 flash loader is a family of dual-core microcontrollers for custom flash device.



1 Introduction

The RT600 is a family of dual-core microcontrollers for embedded applications featuring an Arm[®] Cortex[®] - M33 CPU combined with a Cadence Xtensa HiFi4 advanced Audio Digital Signal Processor CPU. The RT600 provides up to 4.5 MB of on-chip SRAM (plus an additional 128 KB of tightly coupled HiFi4 ram) and several high-bandwidth interfaces to access off-chip flash. The FlexSPI flash interface supports two channels and includes a 32 KB cache and an on-the-fly decryption engine. The RT600 is designed to allow the Cortex-M33 to operate at frequencies of up to 300 MHz and the HiFi4 DSP to operate at frequencies of up to 600 MHz.

2 Boot features

Since the RT600 has no internal flash for code and data storage, images must be stored elsewhere for loading upon reset or the CPU can execute-in-place(XIP) from external memory. Images can be loaded into on-chip SRAM from external flash or downloaded via the serial ports (UART, SPI, I2C, or USB). The code is then validated, and boot ROM jumps to on-chip SRAM.

The evaluation kit for the RT600 uses an octal flash connected to port B as an option for booting in addition to the pSRAM connected to port A. This document describes the necessary steps to boot from port A using the MIMXRT685-EVK with different external memories.

3 Boot settings

Depending on the values of the OTP bits, ISP pins, and the image header type definition, the bootloader decides whether to download code into the on-chip SRAM or run from external memory. The bootloader checks the OTP bit settings first, and then the ISP pins. If bits [3:0] in OTP word BOOT_CFG [0] are not programmed (4b'0000), the boot source is determined by the states of the ISP boot pins (PIO1_15, PIO1_16, and PIO1_17) as shown in Table 1. The focus in this application note is boot mode: FLEXSPI BOOT PORT A.

Boot mode	ISP2 pin PIO1_17	ISP1 pin PIO1_16	ISP0 pin PIO1_15	Description
-	LOW	LOW	LOW	Reserved
SDIO 0 (SD CARD)	LOW	LOW	HIGH	Boot from an SD card device connected to SDIO 1 interface. The RT6xx looks for a valid image in the SD card device. If there is no valid image found, the RT6xx enters the ISP boot mode based on OTP DEFAULT_ISP_MODE bits (6:4, BOOT_CFG[0]).
FLEXSPI BOOT PORT B	LOW	HIGH	LOW	Boot from Quad or Octal SPI Flash devices connected to the FlexSPI interface 0 Port B. The RT6xx looks for a valid image in external Quad/Octal SPI Flash device. If there is no valid image found, the RT6xx enters recovery boot or ISP boot mode.
FLEXSPI BOOT PORT A	LOW	HIGH	HIGH	Boot from Quad/Octal SPI Flash devices connected to the FlexSPI interface 0 Port A. The RT6xx looks for a valid image in external Quad/Octal SPI Flash device. If there is no valid image found, the RT6xx enters recovery boot or ISP boot mode.
SDIO 0 (eMMC)	HIGH	LOW	LOW	Boot from an eMMC device conncted to SDIO O interface. The RT6xx looks for a valid image in the eMMC device. If there is no valid image found, the RT6xx enters the ISP boot

Table 1. Boot mode and ISP Downloader modes based on ISP pins

Boot mode	ISP2 pin PIO1_17	ISP1 pin PIO1_16	ISP0 pin PIO1_15	Description
				mode based on the value of OTP DEFAULT_ISP_MODE bits (6:4, BOOT_CFG [0]).
-	HIGH	LOW	HIGH	Reserved
SERIAL ISP (UART,SPI, I2C, USB-HID)	HIGH	HIGH	LOW	The Serial Interface (UART, SPI, and I2C,USB-HID) is used to program OTP, external Flash, SD or eMMC device.
111	HIGH	HIGH	HIGH	Serial Master boot (SPI Slave, I2C Slave, or UART, USB- HID) is used to download a boot image over the serial interface (SPI Slave, I2C Slave or UART, USB-HID)

 Table 1. Boot mode and ISP Downloader modes based on ISP pins...continued

4 Hardware modifications

This section provides an overview on replacing the pSRAM with with a flash device and changing ISP pins to select FlexSPI Port A.

4.1 Replacing the pSRAM with a flash device

The footprint of the evaluation board for port A supports the memories listed below; however, it is not limited to this list. Consider how the memory must be connected to choose which resistors should be populated.

- S26KS256SDDPBHV02
- S27KS0641DPBHI023
- APS6408L-OBM-BA
- MX25UM51345GXDI00

In this example, the MX25UM51345GXDI00 that is on port B is removed and placed on port A. Verify the memory signals that are needed by looking at the data sheet and comparing the signals to the EVK schematics. See below the necessary hardware changes.

The evaluation board has several 0 Ω resistors that can be added or removed depending on the device that is used. In this case, the original setting shows that R250 (corresponds to A5) and R247 (corresponds to C2) are not populated. It means that there is no connection to these pins. However, for this memory these pins are necessary as they provide the chip select signal and the ECC correction signal, so they must be added (shown in red). In addition, A3 in this memory is not connected (NC), it corresponds to R248 that must be removed (shown in purple). See Figure 1.

AN13386

RT600 Flash Loader For Custom Flash Device







4.2 Changing ISP pins to select FlexSPI Port A

As mentioned previously, if the PRIMARY_BOOT_SRC bits in OTP are not set, the i.MX RT600 reads the status of the ISP pins to determine the boot source. Therefore, to boot from port A instead of port B, the ISP switches on the evaluation board must be changed to:

ISP0 -> High. SW5 pin 1 is off to create a pull-up.

ISP1 -> High. SW5 pin 2 is off to create a pull-up.

ISP2 -> Low. SW5 pin 3 is on to create a pull-down.





5 Flash loader modifications

For the debuggers to be able to load the SDK examples correctly, a flash loader is used. It contains various configurations for the external memory in use. In addition, the flash loader is configured to use Port B. To download and debug your application from the correct flash device on Port A, change the flash loader source code. The steps for flash loader modifications for MCUXpresso, IAR, and Keil are listed below.

5.1 Characteristics and register settings

To understand how the characteristics of the device are translated into the register settings, look at the following characteristics from the Macronix MX25UM51345G Octal Flash data sheet.

Several characteristics are needed from the general and software features of the device. For example, the general features state that the device supports a single-bit structure as well as an 8-bit structure that can be applied to the number of bits in the data pad for flash access. Additionally, there is support for DTR and the maximum frequencies at which it can run depending on the mode chosen. There is also a configurable dummy cycle number for the OPI operation and that SFDP is supported on this device.

GENERAL Supports Serial Peripheral Interface Mode 0 Single Power Supply Operation -1.65 to 2.0 volt for read, erase, and program operations 512Mb: 536,870,912 x 1 bit structure or 67,108,864 x 8 bits (Octa I/O mode) structure 512Mb: 536,870,912 x 1 bit structure or 67,108,864 x 8 bits (Octa I/O mode) structure - Protocol Support - Single I/O and Octa I/O - Support DTR (Double Transfer Rate) Mode - Latch-up protected to 100mA from -1V to Vcc +1V Fast frequency support - Support clock frequency up to - Single I/O mode: 133MHz - Octa I/O mode: 133MHz - Octa Peripheral Interface (OPI) available - Equal Sectors with 4K byte each, or Equal Blocks with 64K byte each - Any Block can be erased individually Programming : - 256byte page buffer - Octa Inut/Output page program to enhance program performance Typical 100,000 erase/program cycles 2 0 years data retention	 SOFTWARE FEATURES Input Data Format SPI: 1-byte command code OPI: 2-byte command code OOPI Output Data Format: Byte mode data sequence Advanced Security Features Block lock protection The BP0-BP3 and T/B status bits define the size of the area to instructions Advanced Sector Protection (Solid and Password Protect) Additional 8K bit security OTP Features unique identifier Factory locked identifiable, and customer lockable Command Reset Program/Erase Suspend and Resume operation Electronic Identification JEDEC 1-byte manufacturer ID and 2-byte device ID Support Serial Flash Discoverable Parameters (SFDP) mode

Figure 7. Characteristics from the Macronix MX25UM51345G Octal Flash data sheet

Although these are not all the characteristics to set the necessary register bits, you can understand how they can or cannot be configured. Examine the CONFIG_OPTION0 register.

Note: See <u>Section 6</u>, for the register options.

Option	Description						
device_type	 For the Macronix Octal Flash devices, there are two options: 4- Macronix Octal DDR 5- Macronix Octal SDR Both of these options are possible according to the data sheet of the devices. Choose the setting appropriately for your application. In this example, the SDR setting is chosen. 						
	24-BGA, 68-WLCSP Octa I/O STR (MHz)						
	Octa I/0 DTR (MHz)						
query_pad	The query pad is the necessary number of pads to be able to read the SFDP or MID. SFDP is supported on this device, for more details of this communication see section 11-1 of the data sheet of MX25UM.						
	The sequence of issuing RDSFDP instruction in SPI is CS# goes low \rightarrow send RDSFDP instruction (5Ah) \rightarrow send 3 address bytes on SI pin \rightarrow send 8 dummy cycles \rightarrow read SFDP code on SO \rightarrow to end RDSFDP operation can use CS# to high at any time during data out.						
	SFDP in SPI is a JEDEC standard, JESD216D. The sequence of issuing RDSFDP instruction in OPI/DOPI mode: CS# low→ send RDSFDP instruction (5Ah/A5h) → send 4 address bytes on SIO pin → send 20 dummy cycles → read SFDP code on SIO [7:0]→ to end RDSFDP operation can use CS# to high at any time during data out.						
	According to the data sheet, it is possible to configure it for serial or octal communication. However, since SPI is a JEDEC standard, this paper uses the first setting for the 1-bit structure.						
	query_pad 19:16 Data pads during Query command (read SFDP or read MID) 0 - 1 2 - 4 3 - 8						
	Figure 8. Data pad options provided for query command						
cmd_pad	The CMD pad is the bit structure used for flash access. For this device, the 1-bit structure and the 8-bit structure are available. This example uses the octal setting.						
	cmd_pad 15:12 Data pads during Flash access command						
	0 - 1 2 - 4 3 - 8						
	Figure 9. Data pad options provided for flash access command						
quad_mode_setting	In Quad mode, flash transmits/receives data on 4 Data pin. This device does not support quad mode; therefore, the setting is 0.						
misc_mode	Miscellaneous mode allows experimental settings, these settings are not recommended for a product and must be 0. If any setting applies to your device, review the data sheet.						
max_freq	You can configure the frequency using the serialClkFreq file of the FlexSPI Flash Configuration block from the user manual. See <u>Section 6</u> . In this example, option 1 indicates SDR mode 24 MHz in normal boot mode.						

The settings chosen are written in the CONFIG_OPTION0 and CONFIG_OPTION1 within the source code. The steps for each IDE are described below and the same setting is applied to all three.

CONFIG_OPTION0 = 0xC0503001

CONFIG_OPTION1 = 0x00000000

Note: See <u>Section 6</u>, for OTP settings.

5.2 Steps for the flash loader modifications for MCUXpresso

Note: Currently, the following steps are based on MCUXpresso + LPC-Link2. SEGGER's JLINK is not supported for the custom driver using MCUXpresso.

1. Open MCUXpresso IDE and select the "Import project(s) from file system..." option from the Quickstart panel.



Figure 10. Quickstart panel in MCUXpresso IDE

- 2. In the Project archive (zip), browse for the following path where the flash loader drivers are located: C:\nxp\LinkServer_24.9.75\Examples\Flashdrivers\NXP\iMXRT
- 3. Here you find several zip folders for the i.MXRT families. Select the iMXRT6xxB_FlexSPI_SFDP.zip.

Name	
iMXRT5xx_FlexSPI_SFDP.zip	
iMXRT6xxB_FlexSPI_SFDP.zip	
iMXRT1020_QSPI.zip	
🔋 iMXRT1050_EcoXiP_Flash.zip	
🔋 iMXRT1050_HyperFlash.zip	
🔢 iMXRT1050_QSPI.zip	

Figure 11. Flash loader zip folder selection

4. Once the zip folder has been selected, click **Next** and select both projects: iMXRT6xxB_FlexSPI_SFDP andLPCXFlashDriverLib.

NXP Semiconductors

AN13386

RT600 Flash Loader For Custom Flash Device

iMXRT6xxB_FlexSPI_SFDP (iMXRT6xxB_FlexSPI_SFDP/) IPCXFlashDriverLib (LPCXFlashDriverLib/)	Select All Deselect All Refresh
Options Copy projects into workspace Hide projects that already exist in the workspace	
Working sets Add project to working sets Working sets:	New V Select

Figure 12. Import window

5. The iMXRT6xxB_FlexSPI_SFDP project has some presets ready that can be chosen from. Find these options using the **drop-down** list in the **Build** icon when selecting the project.

<u> </u>	
📑 🕶 🗐 👘 🛉 👟 🔦	- 🗟 💛 🖓 🖳 🔌 🕨 II 🖷 M Z. 🗇 .e 🗟 🕱 🕪 🗓 🖷 Z. 🗞
눱 Project E 🕱 🛛 🔀 P 🗹	1 Debug (Debug build)
	2 MIMXRT600_FlexSPI_A_MXIC_OPI (Macronix Octal SDR on FlexSPI_A)
✓	3 MIMXRT600_FlexSPI_A_MXIC_OPI_S (Macronix Octal SDR on FlexSPI_A (Secure))
> 🏀 Project Setting	4 MIMXRT600_FlexSPI_A_SFDP_QSPI (SFDP SDR QSPI on FlexSPI_A)
> 🗿 Includes	5 MIMXRT600_FlexSPI_A_SFDP_QSPI_S (SFDP SDR QSPI on FlexSPI_A (Secure))
> 🔑 flashdriver > 🏳 romcode	6 MIMXRT600_FlexSPI_B_MXIC_OPI (Macronix Octal SDR on FlexSPI_B)
> 29 test	7 MIMXRT600_FlexSPI_B_MXIC_OPI_S (Macronix Octal SDR on FlexSPI_B (Secure))
> 🗁 linkscripts	8 MIMXRT600_FlexSPI_B_SFDP_QSPI (SFDP SDR QSPI on FlexSPI_B)
LS_preconnect	9 MIMXRT600_FlexSPI_B_SFDP_QSPI_S (SFDP SDR QSPI on FlexSPI_B (Secure))
📄 readme.txt 📃	

Figure 13. Drop-down list for preset configuration

There are settings ready and available for both ports A and B using the Macronix Octal flash device, as well as flash devices that support SFDP.

6. Open the FlashConfig.h file and scroll down to the enabled section for this option. It is presented as an if-else statement that enables or disables the configuration based on the chosen device. Replace the value for CONFIG OPTION0 and CONFIG OPTION1 with the setting chosen in section 5.1.

5	lif (MIMXRT600_FlexSP)	I_A_MXIC_OPI) defined (MIMXRT600_FlexSPI_A_MXIC_OPI_S)
6	. –	
6		
6	0xC : tag	
6	0 : Option size - 1	1 => 0 not used
6	5 : Macronix Octal	SDR
6	0 : 1 data pad duri	ing query
6	0 : 1 data pad duri	ing flash access (3 = 8)
6	0 : cmd_pad	
	<pre>0 : quad_mode_setti</pre>	ing
	0 : misc_mode	
	1 : max_freq	
7		
		nged to single bit mode, not octal
	efine CONFIG_OPTION0	0xC0503001
7		
7		
	0x0 : Single Flash co	onnected to Port A
7	C	
7	efine CONFIG_OPTION1	0X0000000

Figure 14. Config options with new setting written

7. Before building the project, select "Release_SectorHashing" for the LPCXFlashDriverLib project and build.

RT600 Flash Loader For Custom Flash Device



8. Build the iMXRT6xxB_FlexSPI_SFDP project, the "builds" folder is created. It contains the CFX file necessary to boot from port A with the settings chosen for the Macronix Octal flash device.



Figure 16. Built custom CFX file

9. To test that this flash loader works, import the gpio_led_output example in the same workspace that flash loader drivers are in. In the Advanced Settings window of the SDK Import wizard, select the driver from the workspace option as seen below.

RT600 Flash Loader For Custom Flash Device

<complex-block></complex-block>							\Box \times
<complex-block></complex-block>							TEG.
<complex-block></complex-block>							
<complex-block></complex-block>	Advanced Setting	IS					^
<complex-block></complex-block>	▼ C/C++ Library Settings						
<complex-block></complex-block>	Set library type (and hosting v	variant) Redlib (noho	ost-nf)	~			
<complex-block></complex-block>	Redlib: Use floating point ve	ersion of printf		Nev	libNano: Use floating point vers	sion of printf	
<complex-block></complex-block>		than string based pri	Intf MCUXpresso IDF	New	libNano: Use floating point vers	sion of scanf	
<complex-block><complex-block></complex-block></complex-block>		library "printf"			6		
<complex-block></complex-block>		handler Li	ingerver hash unver				
<complex-block></complex-block>		SP-D16 (Hard AR					~
<complex-block></complex-block>							
<complex-block></complex-block>							
<complex-block></complex-block>			Flash driver MIMXR1000_FlexSPI_b	S_MXIC_OPI.ctx V Brow	se project Browse workspa	ace	~
<complex-block></complex-block>							_
<complex-block></complex-block>					OK Cance	el	
<complex-block></complex-block>	Link application to RAM			Hone			~
<complex-block></complex-block>	▼ Memory Configuration						
<complex-block></complex-block>	Memory details						
NAM BAAM BAAMAN BAAM BAAMAN BAAM BAAM BAAM BAAMAN BAAMAN BAAMAN BAAMAN	Default LinkServer Flash Driver	r				Br	owse
							÷.
<complex-block></complex-block>						MIMXRT600_FlexSPI_B_MX	
<complex-block></complex-block>							
<complex-block></complex-block>			Collin Line Dolate		Investigation of the second	• Countr	
<complex-block></complex-block>	Add Hash Add KAM		Split Join Delete		Import Merge Expor	denerate	~
Image: Construction of Priority Program Construction of Priority Prior Construction of Prior Pri	ettings	from th	a huilda fald	lor	< Back	Next > Finish	Cancel
Movement Market instantion Image: stantiand GMU Compiler Image: stantiand	The created	from tr	ie builds fold	ier.			
Windsaccd Settings • C/C++ Library Settings • Library type (and hotting variant) • Library type (and hotting variant) • Library type (and hotting variant) • Meditive Leading soint variants • Meditive Leading variant	🔀 SDK Import Wizard						
Set Under Configuration • MCU Linker • MCU Linker • MCU Linker • Memory Configuration Memory Configuration MAM USE Configuration Mamory Configuration </td <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>							
• C(C++ Likzay Setting) • Setting Vipe (and hotting valant) • Bedlik: Use doating point version of prind! • Bedlik: Use doating point version of prind! • MMXRET00. Ples.SPL_A.MOC_OPIC. • Indiversions HardFault handler • Indiversions Project Type None • Indiversions Project Type None • Used Setting Point type Point Setting Point type Point						NK	
Set library type (and hosting variant) Bibli (blochester) Bedlib: Use floating point version of printf Bibli (Sub floating point version of printf) Bedlib: Use character rather than string based variant: Bibli (Sub floating Point Version of printf) Bedlib: Use character rather than string based variant: Bibli (Sub floating Point Version of printf) Include seminost Hadrid value handler Bibli (Sub floating Point Version of printf) Include seminost Hadrid value handler Bibli (Sub floating Point Version Printf) NEXT Compiler Bibli (Sub floating Point Version Printf) Include seminost Hadrid value handler Bibli (Sub floating Point Version Printf) Version Printfield Bibli (Sub floating Point Version Printf) Include seminost Hadrid value handler Bibli (Sub floating Point Version Printf) Version Printfield Bibli (Sub floating Point Version Printf) Include seminost Printfield Bibli (Sub floating Point Version Printfield Pointfield Pointfiel	Advanced Setting	15	X LinkServer flash	driver	- - x	NXP	
Set library type (and hosting variand) Reality (lonkook-m)	Advanced Setting	15			x	NF	
Image: Use floating point version of print Image: Redit: Use floating point intro bases concert Image: Redit: Use floating point type Image: Redit: Use floating point type Image: Redit: Use floating point Image: Redit: R	▼ C/C++ Library Settings		Select the elements	from the tree:	×	NF	
Redirect SDK 'PPINITF' to C library 'print' Indicuipants' Indicate seminost HardFault handler Indicate seminost HardFault handler Indit HardFault handler Indicate s	▼ C/C++ Library Settings		Select the elements > MXRT6xxB_ > Select the elements > MXRT6xxB_ > Select the elements	from the tree: rriverLib FlexSPI_SFDP	×	NP	
Include seminor Hardhaut hander Indiscripts Include seminor Hardhaut hander Indiscripts Include seminor Hardhaut hander Indiscripts Set Floating Point type [ProS.SP.D10 (Hard AB] India from the seminor Hardhaut hander MCU Compler India from the seminor Hardhaut hander WCU Compler Float driver ToutZone Project Type None Image standard WCU Linker Image standard Unick seminor Hardhaut hander Image standard McU Linker Image standard Inick seminor Hardhaut hander Image standard Memory Configuration Image standard Memory dealis Image standard Use All Inick Standard Generate Standard Standard Add Flash Add RAM Spitt Delete Import. Merge. Export. Generate. Image standard	C/C++ Library Settings Set library type (and hosting v Redlib: Use floating point ve	variant) Redlib (noho	Select the elements > ⋸ LPCXFlashD > ⋸ iMXRT6xx8 > ⊵ settings > 쓸 MMXRT6 > ☺ builds	from the tree: triverLib FlexSPL_SFDP 600_FlexSPL_A_MXIC_OPI	pint vers	sion of printf	
Vector Size Diver Vector Vector Vector Vector Vector Vector <	C/C++ Library Settings Set library type (and hosting v Redlib: Use floating point ve Redlib: Use character rather	variant) Redlib (noho ersion of printf than string based pri	Select the elements > Control	from the tree: riverLib FlexSPL_SFDP 600_FlexSPL_A_MXIC_OPI (RT600_FlexSPLA_MXIC_OPI.cfx	pint vers	sion of printf	
Set Floating Point type [PAS-SP-D16 (Hard AB Flash Driver IndU C Compaler IndU C 200 (std: gm/97) Flash Driver Plash driver Wrong selection. Please select a file with extension '.cfr' Wrong selection. Please select a file with extension '.cfr' Memory Configuration Memory dealls Default indSerer Flash Driver Plash QSPL [LASH Flash Diver Flash QSPL [LASH Flash Diver Flash QSPL [LASH Flash Diver RAM SRAM RAM SAM RAM Cod0000 0.480000 MIMXXF1600, FlexSPL B.JMX V Add Flash Add RAM Spit Join Delete Import. Merge Export. Generate v	C/C++ Library Settings Set library type (and hosting v Redlib: Use floating point ve Redlib: Use character rather Redirect SDK "PRINTF" to C	variant) Redlib (nohe ersion of printf than string based nri library "printf"	Select the elements Select the elements Select the elements MCRTGroeg MMRK Select the elements MCRTGroeg Select the elements S	from the tree: friverLib FlexSPL_SPDP 600_FlexSPL_A_MXIC_OPI (RT600_FlexSPL_A_MXIC_OPI.cfx er ts	pint vers	sion of printf	
Lenguage standard GNU C99 (std:sgnid9) Fish driver wrkspace. TrustZone Project Type None Wrong selection. Please select a file with extension '.dx' C Mcu Linker C K Cancel Link application to RAM C K Cancel Monory details C K Cancel Default LinkServer Flash Driver Fish Ox00000 Flash CQSPUTLSH Flash Add Flash Add RAM Spile Emin	C/C++ Library Settings Set library type (and hosting v Redlib: Use floating point ve Redlib: Use character rather Redirect SDK "PRINTF" to C Include semihost HardFault	variant) Redlib (nohe ersion of printf than string based nri library "printf"	set-nf) set-nf	from the tree: friverLib FlexSPL_SPDP 600_FlexSPL_A_MXIC_OPI (RT600_FlexSPL_A_MXIC_OPI.cfx er ts] Dirt vers	sion of printf sion of scant X	
TurstZone Project Type None • MCU Linker Wrong selection. Plesse select a file with extension ".cfr." Cancel Cancel Cancel • Memory Configuration Memory Configuration Default LinkServer Flash Driver Type Name Alas Location Size Driver Flash Ox40000 MAM VALUE BRAM Spit Spit Spit Spit Spit Spit	C/C++ Library Settings Set library type (and hosting v Redlib: Use floating point ve Redlib: Use character rather Redlib: Use character rather Redirect SDK "PRINTF" to C Include semihost HardFault Hardware settings	rariant) Redlib (nohc ersion of printf than string based nir library "printf" handler Li	set-nf) set-nf	from the tree: friverLib FlexSPL_SPDP 600_FlexSPL_A_MXIC_OPI (RT600_FlexSPL_A_MXIC_OPI.cfx er ts] Dirt vers	sion of printf sion of scant X	
TustZone Project Type None • MCU Linker	C/C++ Library Settings Set library type (and hosting v Redlik: Use floating point ve Redlik: Use character rather Redirect SDK "PRINTF" to C Include semihost HardFault Hardware settings Set Floating Point type FPVs-S	Variant) Redlib (nohc ersion of printf than string based nri library "printf" i handler SP-D16 (Hard ABI	Select the elements > Control	from the tree: friverLib FlexSPL_SPDP 600_FlexSPL_A_MXIC_OPI (RT600_FlexSPL_A_MXIC_OPI.cfx er ts] Dirt vers	sion of printf sion of scant X	
Will Linker Ink application to RAM • Memory Configuration Memory dealls Default Linkerer Flash Driver Image: State of the	C/C++ Library Settings Set Library type (and hosting v Redlik: Use floating point ve Redlik: Set Floating Point Ve Set Floating Point Ve FNATWATE SetTing Set Floating Point Ve PA-SetTing MCU C Compiler	varianti) Redlib (nohc ersion of printf than string based nri library "printf" handler SP-D16 (Hard AB	Select the elements Select the elements > Constant > Constant	from the tree: friverLib FlexSPL_SPDP 600_FlexSPL_A_MXIC_OPI (RT600_FlexSPL_A_MXIC_OPI.cfx er ts	j bint vers	sion of printf Sion of scart DE	Ŷ
	OfC++ Library Settings Set Library type (and hosting v Redlik: Use floating point ve Redlik: Use floating point ve Redlik: Use character rather Redlik: Use character Redlik: Use character rather Redlik: Use character Redli	ariant) Redlib (nohn ersion of printf than string based and library "printf" handler SP-D16 (Hard AB SP-D16 (Hard AB	Select the elements Select th	from the tree: Invertible PIESPISEDP 800_FIESPIA_MOIC_OPI INTROD_FIESPIA_MOIC_OPI.cfx er s	aint ver aint series workspa	sion of printf Sion of scart DE	Ŷ
Memory details Default LinkServer Flash Driver Type Name Alias Location Size Driver Plash Odd000000 Mix00000 Ox800000 Mix00000 Mix00000 Mix00000 Add Flash Odd0140000 Ox40140000 Ox401000 Ox40000 Ox40140000 Ox40000 Merce	OfC++ Library Settings Set library type (and hosting v Redlik: Use character rather Redlik: Use character Redlik:	ariant) Redlib (nohn ersion of printf than string based and library "printf" handler SP-D16 (Hard AB SP-D16 (Hard AB	Select the elements Select th	from the tree: Invertible PIESPISEDP 800_FIESPIA_MOIC_OPI INTROD_FIESPIA_MOIC_OPI.cfx er s	aint ver aint series workspa	sion of printf Sion of scart DE	Ŷ
Memory details Default LinkServer Flash Driver Type Name Alias Location Size Driver Plash Odd000000 Mix00000 Ox800000 Mix00000 Mix00000 Mix00000 Add Flash Odd0140000 Ox40140000 Ox401000 Ox40000 Ox40140000 Ox40000 Merce	C/C++ Library Settings Set library type (and hosting v Redlik: Use floating point ve Redlik	ariant) Redlib (nohn ersion of printf than string based and library "printf" handler SP-D16 (Hard AB SP-D16 (Hard AB	sst-nf) sst-nf) int MCLXpress lash Driver Flash driver	from the tree: invertab FlesSPL_SEDP 600_FlesSPLA_MAIC_OPI IRT600_FlesSPLA_MAIC_OPI.cfx er is	ion '.cfx'	sion of printf sion of scanf X IDE	× ×
Default LinkServer Flash Driver Browsen Type Name Alass Location Size Driver If and a start of the s		ariant) Redlib (nohn ersion of printf than string based and library "printf" handler SP-D16 (Hard AB SP-D16 (Hard AB	sst-nf) sst-nf) int MCLXpress lash Driver Flash driver	from the tree: invertab FlesSPL_SEDP 600_FlesSPLA_MAIC_OPI IRT600_FlesSPLA_MAIC_OPI.cfx er is	ion '.cfx'	sion of printf sion of scanf X IDE	× ×
Type Name Alias Location Size Driver Flash QSPLFLASH Flash 0x800000 0x800000 MMXRT600, FlexSPLB_MX Image: Splat Sp	C/C++ Library Settings Set library type (and hosting v Redilt: Use floating point ve Redilt	ariant) Redlib (nohn ersion of printf than string based and library "printf" handler SP-D16 (Hard AB SP-D16 (Hard AB	sst-nf) sst-nf) int MCLXpress lash Driver Flash driver	from the tree: invertab FlesSPL_SEDP 600_FlesSPLA_MAIC_OPI IRT600_FlesSPLA_MAIC_OPI.cfx er is	ion '.cfx'	sion of printf sion of scanf X IDE	× ×
Fisch OSPLFLASH Fisch 0x000000 0x000000 MMMSRT600, Fisc.SPLB_MX. RAM SRAM RAM 0x00000 0x400000 0x400000 RAM USB_RAM RAM 0x40140000 0x4000 Add Flash Add RAM Split Join Delete Import_ MergeExportGenerate	C/C++ Library Settings Set library type (and hosting v Redilt: Use floating point ve Redilt: Use floating point ve Redilt: Use floating point ve Redilt: Use floating Point type Redilt: Vse floating Point type Phys-S MCU Compiler Language standard GNUG MCU Linker Link application to RAM Memory Configuration Memory Configuration	variant) Redlib (nohc ersion of printf than string based an library 'printf' handler Library 'printf' SP-D16 (Hard AB SP-D16 (Hard AB S9 (-std=gnu99)	sst-nf) sst-nf) int MCLXpress lash Driver Flash driver	from the tree: invertab FlesSPL_SEDP 600_FlesSPLA_MAIC_OPI IRT600_FlesSPLA_MAIC_OPI.cfx er is	ion '.cfx'	sion of printf sion of canf	
RAM USB_RAM RAM2 0x40140000 0x4000 Add Flash Add RAM Split Isin Delete Import Merge Export Generate v	C/C++ Library Settings Set library type (and hosting v Redili: Use floating point ve Include semihost HardFault Hardware setting: Set Floating Point type [Pu/S-5 MCU Compiler Language standard GNUG MCU Linker Link application to RAM Memory Configuration Memory Configuration Memory Configuration	rariant) Redlib (nohc ersion of printf than string based on library 'printf' handler Lin SP-D16 (Hard AB SP-D16 (Hard AB S9 (-std=gnu99)	sst-nf) sst-nf) ant hCUXpres ash Driver Rash driver Rash driver Composition	from the tree: triverLib FIesSPL_SEDP 600_FIesSPLA_MXIC_OPI KR1500_FIesSPLA_MXIC_OPI.cfx et 15 15 15 10 10 10 10 10 10 10 10 10 10	int vers int vers int vers ion '.ch' Cancel	sion of printf isian af agent IDE	
Add Flash Add RAM Split Ioin Delete Import Merge Export Generate v ? < Back		rariant) Redlib (nohc ersion of printf than string based noi library 'printf' handler library 'printf' library 'printf' libra	sst-nf) sst-nf) int MCUXpres intServer fl lash Driver Flash driver Alias Flash Alias Flash	from the tree: Invertibi Fles.SPL_SEDP 500_Fles.SPL_A_MXIC_OPI IXTEOD_Fles.SPL_A_MXIC_OPI_cfx 5 5 6 0 0 0 0 0 0 0 0 0 0 0 0 0	bint vers bint v	sion of printf dama degand IDE el	
(?) Cancel		rariant) Redlib (noho ersion of printf than string based and library "printf" handler SP-D16 (Hard ABI S9 (-std= gnu99) r Name QSP [FLASH SRAM	sst-nf) sst-nf) inti-	from the tree: twierLib PiesSP1_SDP 800_FiesSP1_A_MXIC_OPI RTR00_FiesSP1_A_MXIC_OPI_cfx er er a n. Please select a file with exten OK Location 0x800000 0x800000	int veri int	sion of printf dama degand IDE el	
	C/C++ Library Settings Set library type (and hosting v Redilt: Use floating point ve Redilt: Use floating point ve Redilt: Use floating point ve Redilt: Use floating Point type Redilt: Use floating Point type Redilt: Vse floating Redilt: Vse floa	rariant) Redlib (noho ersion of printf than string based and library "printf" handler SP-D16 (Hard ABI S9 (-std= gnu99) r Name QSP [FLASH SRAM	Select the elements Sot-Inf) est inf inf int int int int int int int int	from the tree: twierLib PiesSP1_SDP 800_FiesSP1_A_MXIC_OPI RTR00_FiesSP1_A_MXIC_OPI_cfx er er a n. Please select a file with exten OK Location 0x800000 0x800000	bint vers bint v	sion of printf sion of ganf X DE BE DE D D Criver MIMKRT600_PlexSPL_B_MX	
	C/C++ Library Settings Set library type (and hosting v Redilt: Use floating point ve Redilt: Use floating point ve Redilt: Use floating point ve Redilt: Use floating Point type Redilt: Use floating Point type Redilt: Vse floating Redilt: Vse floa	rariant) Redlib (noho ersion of printf than string based and library "printf" handler SP-D16 (Hard ABI S9 (-std= gnu99) r Name QSP [FLASH SRAM	Select the elements Sot-Inf) est inf inf int int int int int int int int	from the tree: twierLib PiesSP1_SDP 800_FiesSP1_A_MXIC_OPI RTR00_FiesSP1_A_MXIC_OPI_cfx er er a n. Please select a file with exten OK Location 0x800000 0x800000	bint vers bint v	sion of printf sion of ganf X DE BE DE D D Criver MIMKRT600_PlexSPL_B_MX	
	C/C++ Library Settings Set library type (and hosting v Redilt: Use floating point ve Redilt: Use floating point ve Redilt: Use floating point ve Redilt: Use floating Point type Redilt: Use floating Point type Redilt: Vse floating Redilt: Vse floa	rariant) Redlib (noho ersion of printf than string based and library "printf" handler SP-D16 (Hard ABI S9 (-std= gnu99) r Name QSP [FLASH SRAM	Select the elements Sot-Inf) est inf inf int int int int int int int int	from the tree: twierLib PiesSP1_SDP 800_FiesSP1_A_MXIC_OPI RTR00_FiesSP1_A_MXIC_OPI_cfx er er a n. Please select a file with exten OK Location 0x800000 0x800000	bint vers bint v	sion of printf sion of ganf X DE BE DE D D Criver MIMKRT600_PlexSPL_B_MX	
		rariant) Redlib (noho ersion of printf than string based and library "printf" handler SP-D16 (Hard ABI S9 (-std= gnu99) r Name QSP [FLASH SRAM	Select the elements Sot-Inf) est inf inf int int int int int int int int	from the tree: twierLib PiesSP1_SDP 800_FiesSP1_A_MXIC_OPI RTR00_FiesSP1_A_MXIC_OPI_cfx er er a n. Please select a file with exten OK Location 0x800000 0x800000	int veri int veri int veri int veri int veri int veri int veri vorkpa int vorkpa int	sion of printf diamate and IDE see	
n driver select		rariant) Redlib (noho ersion of printf than string based and library "printf" handler SP-D16 (Hard ABI S9 (-std= gnu99) r Name QSP [FLASH SRAM	Select the elements Sot-Inf) est inf inf int int int int int int int int	from the tree: twierLib PiesSP1_SDP 800_FiesSP1_A_MXIC_OPI RTR00_FiesSP1_A_MXIC_OPI_cfx er er a n. Please select a file with exten OK Location 0x800000 0x800000	int veri int veri int veri int veri int veri int veri int veri vorkpa int vorkpa int	sion of printf diamate and IDE see	

11. Click Finish to close the SDK Import wizard. Next, open the flash_config.c source file in the flash_config folder of the gpio_led_output project. Change the sflashAlSize = 0 to the BOARD_FLASH_SIZE macro and replace the sflashBlSize with 0.

RT600 Flash Loader For Custom Flash Device



Figure 19. Modify board flash size to match port A

12. Rebuild the project and run. You have successfully booted from port A.

5.3 Steps for the flash loader modifications for IAR

- 1. For IAR, open the flash loader project from the following path in your PC: C:\Program Files\IAR Systems\Embedded Workbench 9.0\arm\src\flashloader\NXP \FlashIMXRT600_EVK_FLEXSPI
- 2. To keep the changes locally, copy the flash loader project folder to your workspace.
- 3. In the FlashIMXRT600_EVK_FLEXSPI.c source file, replace the values for configOption.option0.U and configOption.option1.U with the setting discussed in section 5.1.



4. Build this project. The FlashIMXRT600_EVK_FLAXSPI.out file is generated in the Output folder of the project.

Files
🗉 🌒 FlashIMXRT600_EVK_FLEXSPI - FlashIMXRT600_EVK_FLEXSPI
Here Hereich
⊣⊕ 🛋 rom_api
HE STASHIMART600_EVK_FLEXSPLc
🛏 🖹 readme.txt
느무 💼 Output
🖹 FlashIMXRT600_EVK_FLEXSPI.map
└── 🗋 FlashIMXRT600_EVK_FLEXSPI_import_lib.o

Figure 21. Built custom driver out file

5. Locate the IAR path:

C:\Program Files\IAR Systems\Embedded Workbench 9.0\arm\config\flashloader\NXP The .flash, .board, and .mac files are stored here for different NXP devices. Make a copy of the FlashIMXRT600_EVK_FLEXSPI.board and FlashIMXRT600_EVK_FLEXSPI.flash files. Add them to your application project folder, in this case the gpio_led_output folder. In this example, the copies are renamed:

- FlashIMXRT600 EVK FLEXSPI v2.flash
- FlashIMXRT600_EVK FLEXSPI v2.board

6. Open the .flash file and replace the path where your local .out file is stored from step 4.



Figure 23. Pointing .board file to custom .flash file

- 8. Save your changes and open the gpio_led_output example for the IAR IDE.
- 9. Select the project and open the project options. Locate the **Debugger > Download tab** and select **override default .board file.**

Category: General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator CADI CMSIS DAP GDB Server I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET TI XDS	Factory Settings Setup Download Suppress download Suppress download Use flash loader(s) Override default board file \$TOOLKIT_DIR\$\config\flashloader\NXP\FlashIMX\$} Edt Perform mass erase before flashing
TI XDS	OK Cancel

10. Find the gpio_led_output project path and select the newly modified .board file and click OK.

xr	t685 > driver_examples > gpio > led_output > iar	
—	Name	
	MIMXRT685Sxxxx_cm33_ram.icf	
	MIMXRT685Sxxxx_cm33_flash.icf	
	🔮 gpio_led_output.eww	
	gpio_led_output.ewp	
	gpio_led_output.ewd	
	FlashIMXRT600_EVK_FLEXSPI_v2.flash	
	FlashIMXRT600_EVK_FLEXSPI_v2.board	
-		
Figure 25. Select custom .board file		

11. Rebuild the project and run. You have successfully booted from port A.

5.4 Steps for the flash loader modifications for KEIL

- 1. For KEIL, open the flash loader project provided in the SW package.
- 2. In the FlashPrg.c source file, replace the values for CONFIG_OPTION0 and CONFIG_OPTION1 with the setting from section 5.1.

PlashPrg.c* 25 * 26 *
26 *
27 -
28 戸/* History:
29 * Version 1.00
30 * Initial release
31 */
32 L
33 #include <stdbool.h></stdbool.h>
34 #include "/FlashOS.H" // FlashOS Structures
35 #include <string.h></string.h>
<pre>36 #include "bl_api.h" 37 #include "cmsis compiler.h"</pre>
37 #Include "Cmsis_Compiler.n"
39 /** local definitions **/
40
41 <u>#define FLASH BASE ADDR 0x08000000</u>
42 #define CONFIG OPTIONO 0xc0503001
43 #define CONFIG OPTION1 0x00000000
Figure 26. Config option new setting written

3. Build the project inside the workspace.

4. A MIMXRT6XX_EVK_FLEXSPI.FLM file is generated in the **Output** folder of the project. Change the name to new MIMXRT6XX_EVK_FLEXSPI.FLM to identify it later on.

Name	Date modified	Туре
MIMXRT6XX_EVK_FLEXSPI	5/26/2021 1:39 PM	File folder
FlashOS.H	6/23/2020 12:02 PM	H File
new_MIMXRT6XX_EVK_FLEXSPI.FLM	5/26/2021 1:39 PM	FLM File

Figure 27. Rename custom FLM file

- 5. Copy and paste this file in the following path:
 - $C:\Evil_v5\ARM\Flash$
- 6. Select the project and open the **Options for Target**.... Locate the **Utilities** tab.

ᇶ gpio_led_output debug	
	Options for Target 'gpio_led_output debug' Alt+F7
🕀 🛄 board	Add Group
🕀 🧰 doc 🕀 📴 utilities 👘	Manage Project Items
	Open Map File
III III MIMART685S	Open Build Log
🕀 🧰 component/ua	Rebuild all target files
😑 🦢 flash_config 🔛	Build Target F7
🗄 🗋 flash_confi	Show Include File Dependencies
Figure 28. Options for project	
7. Click the settings and open the Target d	river setup dialog box.
🐰 Options for Target 'gpio_led_ou	tput debug' X
Device Target Output Listing	Jser C/C++ (AC6) Asm Linker Debug Utilities
Configure Flash Menu Command -	
We Target Driver for Flash Pr	ogramming 🛛 🔽 Use Debug Driver
Use Debug Drive	r Settings 🛛 🗹 Update Target before Debugging
Init File:	Edt
C Use External Tool for Flash Pr	ogramming
Command:	
Arguments:	
🗖 Bun Independe	nt
Configure Image File Processing (F Output File:	CARM):
Ouput me.	source
Image Files Root Folder:	Generate Listing
	OK Cancel Defaults Help
Figure 29. Settings dialog box for debug driv	er

Note: See <u>Section 6</u>, for .init file modifications.

- 8. The two loader paths that are shown are available from the predetermined settings. In this example, these two are removed and a new one with the modified flash loader is created.
- 9. Click the **Add** button to find the modified flash loader.

CMSIS-DAP ARMv8-M Target Driver Setup	×
Debug Trace Flash Download Pack	
Download Function Image: Fase Full Chip Image: Program RAM for Algorithm Image: Program ming Algorithm Image: Program ming Algorithm Start: Image: Program ming Algorithm	
Programming Augustum Description Device Size Device Type Address Range MIMXRT6XX FLEXSPI 64M Ext. Rash SPI 08000000H - 0BFFFFFH MIMXRT6XX FLEXSPI 64M Ext. Rash SPI 18000000H - 1BFFFFFFH	
Start: Size:	
Add	
OK Cancel Help	
arget Driver Setup	

10. Find the driver created and click **Add**.

Description	Flash Size	Device Type	Origin	^
MIMXRT6XX FLEXSPI	64M	Ext. Flash SPI	Device Family Package	
MIMXRT6XX FLEXSPI	64M	Ext. Flash SPI	Device Family Package	
AM29x128 Flash	16M	Ext. Flash 16-bit	MDK Core	
K8P5615UQA Dual Flash	64M	Ext. Flash 32-bit	MDK Core	
LPC18xx/43xx MX25V8035F		Ext. Flash SPI	MDK Core	
LPC18xx/43xx S25FL032 SP		Ext. Flash SPI	MDK Core	
LPC18xx/43xx S25FL064 SP		Ext. Flash SPI	MDK Core	
LPC407x/8x S25FL032 SPIFI	4M	Ext. Flash SPI	MDK Core	
LPC5460x MT25QL128 SPIFI	16M	Ext. Flash SPI	MDK Core	
M29W640FB Flash	8M	Ext. Flash 16-bit	MDK Core	
MIMXRT105x EcoXiP Flash	4M	Ext. Flash SPI	MDK Core	
MIMXRT6XX FLEXSPI	128M	Ext. Flash SPI	MDK Core	
RC28F640J3x Dual Flash	16M	Ext. Flash 32-bit	MDK Core	
S25FL128S_V2C	16M	Ext. Flash SPI	MDK Core	~
S29GL064N Dual Flash	16M	Ext. Flash 32-bit	MDK Core	>
				· ·
C:\Keil_v5\ARM\flash\new_MIM		ELEVODI ELM		
C. (Kell_V5 (A) (M VIAsi View_Min		_ LEXST I.I EM		
	Add	Cancel		

Figure 31. Adding custom flash programming algorithm

- 11. To save the project changes, click **OK**.
- 12. Rebuild the project and download. You have successfully booted from port A.

6 Required options, settings, and modifications

It gives required information on Option 0 and Option 1 definition, FlexSPI flash configuration block, FlexSPI boot configurations in OTP, settings of the shadow register, and LUT Section from .init file that may need to be modified.

1. Option0 definition in table 1004 of the RT6xx User manual.

Table 2. Table 1004. Option0 def	finition
----------------------------------	----------

Field	Bits	Description
tag	31:28	The tag of the config option, fixed to 0x0C
option_size	27:24	Size in bytes = (Option Size + 1) * 4
		It is 0 if only option0 is required.
device_type	23:20	Device Detection Type
		0 - Read SFDP for SDR commands
		1 - Read SFDP for DDR Read commands
		2 - HyperFLASH 1V8
		3 - HyperFLASH 3V
		4 - Macronix Octal DDR
		5 - Macronix Octal SDR
		6 - Micron Octal DDR
		7 - Micron Octal SDR
		8 - Adesto EcoXiP DDR
		9 - Adesto EcoXiP SDR
query_pad	19:16	Data pads during Query command (read SFDP or read MID)
		0- 1
		2-4
		3-8
cmd_pad	15:12	Data pads during Flash access command

Field	Bits	Description
		0-1
		2-4
		3-8
quad_mode_setting	11:8	Quad Mode Enable Setting
		0 - Not configured
		1 - Set bit 6 in Status Register 1
		2 - Set bit 1 in Status Register 2
		3 - Set bit 7 in Status Register 2
		4 - Set bit 1 in Status Register 2 vis 0x31 command
		This setting the flash to be configured into QPI mode. User code must reset the flash
		into SPI mode, the ROM does not do this automatically.
		Note: This field will be effective only if device is compliant with JESD216 only (9 longword SFDP table)
misc_mode	7:4	Miscellaneous Mode
		0 - Not enabled
		1 - Enable 0-4-4 mode for High Random Read performance
		3 - Data Order Swapped mode (for MXIC OctaFlash only)
		5 - Select the FlexSPI data sample source as internal loop back, more details please refer
		FlexSPI usage
		6 - Config the FlexSPI NOR flash running at stand SPI mode
		Note: Experimental feature, do not use in products, keep it as 0.
max_freq	3:0	Max Flash Operation speed
		0 - Don't change FlexSPI clock setting
		Others – See fuse map of FlexSPI clock setting

Table 2. Table 1004. Option0 definition ...continued

2. Option 1 definition in table 1005 of the RT6xx User manual.

Table 3. Table 1005.Option1 definition

Field	Bits	Description			
flash_connection	31:28	Flash connection option:			
		Single Flash connected to port A			
		- Parallel mode			
		2 - Single Flash connected to Port B			
drive_strength	27:24	The Drive Strength of FlexSPI Pads			
dqs_pinmux_group	23:20	The DQS pin mux Group Selection			
pinmux_group	19:16	The pin mux group selection			
status_override	15:8	Override status register value during device mode configuration			
dummy_cycles	7:0	Dummy cycles for read command			
		0 - Use detected dummy cycle			
		Others - dummy cycles provided in flash data sheet			

3. Flash frequency in the FlexSPI flash configuration block. Table 997 in RT6xx User manual.

serialClkFreq 0x046 1 Flash Frequency. In Normal boot mode[BOOT_CFG[0]:bit7==0] SDR mode: 1 - 24 MHz 2 - 48 MHz DDR mode: 1 - 48 MHz In High speed boot mode mode[BOOT_CFG[0]:bit7==1] SDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz 6 - 120 MHz 7 - 133 MHz 8 - 166 MHz 9 - 200 MHz 9 - 200 MHz DDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz 5 - 100 MHz 6 - 120 MHz 7 - 133 MHz 8 - 166 MHz 9 - 200 MHz DDR mode: 1 - 30 MHz 5 - 100 MHz 6 - 100 MHz 7 - 133 MHz 8 - 166 MHz 9 - 200 MHz 1 - 30 MHz 5 - 100 MHz 7 - 133 MHz 8 - 166 MHz 9 - 200 MHz 1 - 30 MHz 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz 5 - 100 MHz 5 - 100 MHz 5 - 100 MHz 6 - 120 MHz 7 - 133 MHz 8 - 166 MHz 9 - 200 MHz 1 - 30 MHz 5 - 100 MHz 5 - 100 MHz 7	Field	Offset	Size in bytes	Description
SDR mode: 1 - 24 MHz 2 - 48 MHz DDR mode: 1 - 48 MHz In High speed boot mode mode[BOOT_CFG[0]:bit7==1] SDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz 6 - 120 MHz 7 - 133 MHz 8 - 166 MHz 9 - 200 MHz 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz 6 - 120 MHz 7 - 133 MHz 8 - 166 MHz 9 - 200 MHz 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz		0040	-	
1 - 24 MHz 2 - 48 MHz DDR mode: 1 - 48 MHz In High speed boot mode mode[BOOT_CFG[0]:bit7==1] SDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz 6 - 120 MHz 7 - 133 MHz 8 - 166 MHz 9 - 200 MHz DDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 3 - 60 MHz 4 - 80 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz	serialCikFreq	UXU46	1	
2 - 48 MHz DDR mode: 1 - 48 MHz In High speed boot mode mode[BOOT_CFG[0]:bit7==1] SDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz 6 - 120 MHz 7 - 133 MHz 8 - 166 MHz 9 - 200 MHz DDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 3 - 60 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz				
DDR mode: 1 - 48 MHz In High speed boot mode mode[BOOT_CFG[0]:bit7==1] SDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz 6 - 120 MHz 7 - 133 MHz 8 - 166 MHz 9 - 200 MHz DDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz				
1 - 48 MHz In High speed boot mode mode[BOOT_CFG[0]:bit7==1] SDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz 6 - 120 MHz 7 - 133 MHz 8 - 166 MHz 9 - 200 MHz DDR mode: 1 - 30 MHz 2 - 50 MHz 5 - 100 MHz 6 - 120 MHz 7 - 133 MHz 8 - 166 MHz 9 - 200 MHz DDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz 5 - 100 MHz				
In High speed boot mode mode[BOOT_CFG[0]:bit7==1] SDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz 6 - 120 MHz 7 - 133 MHz 8 - 166 MHz 9 - 200 MHz DDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz				
SDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz 6 - 120 MHz 7 - 133 MHz 8 - 166 MHz 9 - 200 MHz DDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 3 - 60 MHz 3 - 60 MHz 5 - 100 MHz 5 - 100 MHz				
1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz 6 - 120 MHz 7 - 133 MHz 8 - 166 MHz 9 - 200 MHz DDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz				
2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz 6 - 120 MHz 7 - 133 MHz 8 - 166 MHz 9 - 200 MHz DDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz				
3 - 60 MHz 4 - 80 MHz 5 - 100 MHz 6 - 120 MHz 7 - 133 MHz 8 - 166 MHz 9 - 200 MHz DDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz				
4 - 80 MHz 5 - 100 MHz 6 - 120 MHz 7 - 133 MHz 8 - 166 MHz 9 - 200 MHz DDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz				
5 - 100 MHz 6 - 120 MHz 7 - 133 MHz 8 - 166 MHz 9 - 200 MHz DDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz				
6 - 120 MHz 7 - 133 MHz 8 - 166 MHz 9 - 200 MHz DDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz				
7 - 133 MHz 8 - 166 MHz 9 - 200 MHz DDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz				
8 - 166 MHz 9 - 200 MHz DDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz				
9 - 200 MHz DDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz				
DDR mode: 1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz				
1 - 30 MHz 2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz				
2 - 50 MHz 3 - 60 MHz 4 - 80 MHz 5 - 100 MHz				
3 - 60 MHz 4 - 80 MHz 5 - 100 MHz				
4 - 80 MHz 5 - 100 MHz				
5 - 100 MHz				
6 - 120 MHz				5 - 100 MHz
				6 - 120 MHz
7 - 133 MHz				
8 - 166 MHz				8 - 166 MHz
9 - 200 MHz				9 - 200 MHz

 Table 4. Table 997. FlexSPI flash configuration block

4. The boot ROM is set to find devices that support 3B read by default. In some devices, the commands are different. Therefore, the OTP fuses must reflect the correct device.

 Table 5.
 Table 998. FlexSPI boot configurations in OTP

Field Name	Enum Name	Description	Offset	Width	Value
FLEXSPI_FLASH_ TYPE		Define typical Serial NOR Flash types	4	3	
	QSPI_ADDR_3B	Device supports 3B read by default			000b
		Reserved			001b
	HYPER_1V8	HyperFlash 1V8			010b
	HYPER_3V3	HyperFlash 3V3			011b
	OSPI_DDR_MXIC	MXIC Octal DDR			100b
	OSPI_DDR_ MICRON	Micron Octal DDR			101b
		Reserved			110b
		Reserved			111b

For development purposes, use the OTP shadow registers. It identifies the type of memory used while the device remains powered. If a power-on-reset occurs, this setting must be reconfigured. For example, if you use a hyper flash in which the 3B read command is not supported, set the shadow register at the end of the main function as shown below.



Figure 32. Writing to shadow register

5. If the specific memory requires, the .init file may be modified with respect to the LUT. Refer to the user manual to provide the correct settings of the LUT. In this application note example, it is not necessary since the LUT reflects the settings needed for the QSPI flash. Shown below is the section of the .init file that may need to be modified.

// Config look up ta	able
_WDWORD(0x40134018,	0x5AF05AF0);
_WDWORD(0x4013401C,	0x2);
_WDWORD(0x40134200,	0x08200413);
_WDWORD(0x40134204,	0x00002404);
_WDWORD(0x40134208,	0x0);
_WDWORD(0x4013420c,	0x0);

Figure 33. LUT Section from .init file

7 Conclusion

This application note describes how to modify the flash loader source code step by step to boot from port A using the MIMXRT685-EVK. For more information, refer to "RT6xx User Manual".

8 References

- RT6xx User manual (document: UM11147)
- MX25UM51345G Datasheet

9 Revision history

Table 6 summarizes the revisions to this document.

Table 6. Revision history

Revision number	Date	Substantive changes
AN13386 v.1.0		Updated <u>Section 5.2</u> Changed the "Appendix A" to <u>Section 6</u>
0	01 September 2021	Initial release

AN13386 Application note

RT600 Flash Loader For Custom Flash Device

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at <u>PSIRT@nxp.com</u>) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

 $\ensuremath{\mathsf{NXP}}\xspace$ B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners. **NXP** — wordmark and logo are trademarks of NXP B.V.

AN13386

RT600 Flash Loader For Custom Flash Device

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Cadence — the Cadence logo, and the other Cadence marks found at <u>www.</u> <u>cadence.com/go/trademarks</u> are trademarks or registered trademarks of Cadence Design Systems, Inc. All rights reserved worldwide. IAR — is a trademark of IAR Systems AB.

J-Link — is a trademark of SEGGER Microcontroller GmbH.

RT600 Flash Loader For Custom Flash Device

Contents

1	Introduction	2
2	Boot features	2
3	Boot settings	2
4	Hardware modifications	3
4.1	Replacing the pSRAM with a flash device	3
4.2	Changing ISP pins to select FlexSPI Port A	5
5	Flash loader modifications	6
5.1	Characteristics and register settings	6
5.2	Steps for the flash loader modifications for	
	MCUXpresso	8
5.3	Steps for the flash loader modifications for	
	IAR	12
5.4	Steps for the flash loader modifications for	
	KEIL	14
6	Required options, settings, and	
	modifications	16
7	Conclusion	19
8	References	19
9	Revision history	19
	Legal information	20

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

© 2025 NXP B.V.

All rights reserved.

For more information, please visit: https://www.nxp.com

Document feedback Date of release: 10 February 2025 Document identifier: AN13386