Boot the i.MX RT1180 and Test the Boot Time from Different Devices
Rev. 1.0 — 4 March 2025 Application note

Document information

Information	Content
Keywords	AN14589, i.MX RT1180, boot ROM, boot time, FlexSPI
Abstract	This document describes the i.MX RT1180 system boot, i.MX RT1180 booting from different boot devices, use of the MCUXpresso secure provisioning tool, and the method to test the boot time from different boot devices.



1 Introduction

The i.MX RT1180 crossover MCU of NXP is a dual-core device that features the Arm Cortex-M7 (CM7) and Cortex-M33 (CM33) cores. The CM7 and CM33 cores can operate at a speed of up to 800 MHz and 240 MHz respectively with 1.5 MB on-chip random-access memory (RAM).

The i.MX RT1180 supports the following features:

- Multiple protocols
- Bridging communications between the real-time Ethernet and the Industry 4.0 systems
- Advanced security with the integrated EdgeLock Secure Enclave

The i.MX RT1180 MCU includes an integrated Gbit/s time sensitive networking (TSN) switch and EtherCAT slave controller (ESC). These features make it suitable for industrial and automotive communication applications. The MCU supports the use of the MCUXpresso ecosystem, which has the following features:

- Software development kit (SDK)
- A choice of IDEs
- · Secure provisioning and configuration tools to enable rapid development

The i.MX RT1180 supports booting from various boot devices. For more details, see Section 2.8.

This document outlines the following details:

- i.MX RT1180 system boot
- i.MX RT1180 booting from different booting devices
- Use of the MCUXpresso secure provisioning tool
- Method to test the boot time from different booting devices

2 i.MX RT1180 boot overview

2.1 Boot feature

The boot process starts with any reset when the hardware reset logic triggers the CM33 boot core to execute from the on-chip boot ROM. To determine the boot flow behavior of the device, the boot ROM uses the state of the internal register bit field SBMR2 [BOOT_MODE] and various fuse settings. This register reflects the status of the Boot mode pins.

2.2 Serial downloader

The boot ROM code allows the downloading of the programs that the device runs. For example, the provisioning program can use a serial connection to provide a new image to the boot device. Usually, the provisioning program is downloaded to the internal RAM and allows programming of the boot devices, such as the FlexSPI NOR flash. The boot ROM serial downloader uses a low-power serial peripheral interface (LPSPI), low-power universal asynchronous receiver/transmitter (LPUART), or a high-speed USB in a non-stream mode connection. Table 1 lists the serial downloader peripherals PinMux on MIMXRT1180-EVK.

Peripheral	Instance	Port (IO function)	PAD
LPUART	1	LPUART1_TX	GPIO_AON_08
		LPUART1_RX	GPIO_AON_09
LPSPI	1	LPSPI1_SCK	GPIO_AON_04
		LPSPI1_PCS0	GPIO_AON_05

 Table 1. Serial downloader peripherals PinMux

Derinherel	Instance	Port (IO function)	DAD
Peripheral	Instance	Port (IO function)	PAD
		LPSPI1_SDO	GPIO_AON_06
		LPSPI1_SDI	GPIO_AON_07
USB	1	USB1_DN	USB1_DN
		USB1_DP	USB1_DP

Table 1. Serial downloader peripherals PinMux...continued

2.3 External memory configuration data

The external memory configuration data (XMCD) allows the boot ROM code to configure the following memory:

- Synchronous dynamic random-access memory (SDRAM) connected to the smart external memory controller (SEMC) controller
- HyperRAM/APMemory pseudostatic random-access memory (PSRAM) via the FlexSPI controller, from an external program image residing on the boot device

The XMCD aims to simplify the external RAM enablement, and it is a replacement for the legacy device configuration data (DCD).

2.4 Secure boot

A key feature of the i.MX RT1180 boot ROM is to perform a secure boot. The signing process can be done through a secure provisioning tool, which NXP provides. The signatures are then included as a part of the final program image. If configured, the signatures are verified via the public keys included in the program image.

In addition to support the digital signature verification to authenticate the program images, the encrypted boot is also supported. The encrypted boot can be used to prevent the cloning of the program image directly off the boot device. It can be performed on all the boot devices supported on the chip.

2.5 Boot ROM features

The following are the main features of the boot ROM:

- Booting from CM33
- · Booting from various boot devices
- Recovery boot once primary boot failed
- External RAM expansion using XMCD
- Serial downloader via USB, universal asynchronous receiver/transmitter (UART), and LPSPI
- Encrypted boot via advanced encryption standard-cipher block chaining (AES-CBC)
- Digital signature based advanced high assurance boot (AHAB)
- Encrypted execute-in-place (XIP) booting on serial NOR via FlexSPI interface powered by on-the-Fly AES decryption (OTFAD) and inline encryption engine (IEE)

The boot ROM supports the following primary boot devices:

- Serial NOR flash via FlexSPI
- Serial NAND flash via FlexSPI
- SLC RAWNAND flash via SEMC
- SD or embedded multi-media card (eMMC) via uSDHC

2.6 Boot mode selection

The combination of the following settings decides the Boot mode:

AN14589

- ROM API
- Fuse
- BOOT_MODE pins

Figure 1 shows the Boot mode determination.



2.7 Boot mode pin settings

The i.MX RT1180 has three Boot mode pins. The selection of the Boot mode depends on the binary value stored in the internal register field SBMR2 [BOOT_MODE]. Sampling the BOOT_MODE[2:0] pin inputs on the rising edge of the POR_B initializes the Boot mode. After sampling these inputs, their subsequent state does not affect the contents of SBMR2 [BOOT_MODE]. Table 2 shows the Boot mode pin definitions.

BOOT_MODE[2:0]	Boot type
000	Boot from internal fuses
001	Serial downloader
010	eMMC 8 bits via an uSDHC2
011	SD 4 bits via an uSDHC1
100	Serial NOR flash with JESD216 via FlexSPI1, primary group, PortA
101	Serial NAND flash with 2K page via FlexSPI1, primary group, PortA
110	Infinite loop modes
111	Test mode

Table 2. Boot mode pin definitions

Note: Burn the BOOT_CFG fuse for different boot devices.

2.8 Boot devices

The i.MX RT1180 supports various boot devices:

- Serial NOR flash via FlexSPI interface
- Serial NAND flash via FlexSPI interface
- · NAND flash with SEMC interface on CS0 with bus width 8 bits or 16 bits
- · SD/eMMC via uSDHC interface supporting high capacity cards
- Serial NOR/electrically erasable programmable read-only memory (EEPROM) via LPSPI

2.9 Boot image

The i.MX RT1180 boot image can also be called a container-based image. Table 3 shows the top-level view of the boot image.

Table 3. The Top-level view of the boot image

- -

viemory configuration block			
Container 1	Container header		
	Image array entry		
	Signature block		
	Padding to 1 kB (0x400) alignment		
Container 2 (optional)	Container header		
	Image array entry		
	Signature block		
	Padding to 1 kB (0x400) alignment		
Padding to 8 kB (0x2000) alignment			
NXP images (optional)	NXP firmware		
User or OEM images	Application data (optional)		
	CM7 image (optional)		
	CM33 image		

The memory configuration block is a data region from the beginning of the boot device memory, to the start of the container-based image. The memory configuration block has the different size and layout for the different boot devices.

The container consists of the following items:

- Container header
- Image array entry
- Signature block

The container header contains the following essential information about the entire container image, which the ROM uses:

- · Size of the container
- Number of images
- · Fuse version
- · SW version, and so on

The image array entry contains the following information:

- Image offset
- Image size
- · Load address, and so on

The signature block contains the following information:

- SRK table offset
- Certificate offset
- Signature offset, and so on

2.10 MCUXpresso secure provisioning tool

The MCUXpresso secure provisioning tool is a GUI-based application, which simplifies the generation and provisioning of bootable executables on NXP MCU devices. The graphical interface provides a streamlined development flow, making it simpler to prepare, flash and fuse images, while using and providing access to the existing utilities.

To achieve the advance scripting, use the command-line interface. To achieve a more advanced secure provisioning flow, modify the tool-generated scripts. <u>Figure 2</u> shows the MCUXpresso secure provisioning tool block diagram.



Figure 2. MCUXpresso secure provisioning tool block diagram

3 Boot i.MX RT1180 from different boot devices

This section explains the process to use the MCUXpresso secure provisioning tool to generate the bootable image, which can boot from different boot devices. These boot devices can be QSPI flash, NAND flash, eMMC, SD card, and SPI flash.

3.1 Boot from Quad SPI flash via FlexSPI

1. Open the SDK demo and build the XIP image via IAR. Figure 3 shows the IAR configuration.

AN14589

NXP Semiconductors

AN14589

Boot the i.MX RT1180 and Test the Boot Time from Different Devices

Filee	1 - /*
nies Riebollo world d	\sim 2 \sim Copyright (c) 2013 -
	enio_chijj = nexspi_nor •
Options for node "hello Category: General Options Static Analysis Runtime Checking	_world_demo_cm33" ×
C/C++ Compiler Assembler Output Converter Custom Build Linker Build Actions Debugger Simulator CADI CMSIS DAP E2/E2 Lite GDB Server G+LINK I-jet J-Link/J-Trace TI Stellaris Nu-Link	Generate additional output Output format: Motorola S-records Output file Override default hello_world_demo_cm33.srec
ST-LINK Third-Party Driver	

Figure 3. XIP image IAR configuration

• To build an image for booting on FlexSPI instance2, ensure that you link file that requires modification. Change the address for m_flash_start to FlexSPI2 address (0x40000000).

/* CM33	3 use la	ast 2M bytes	space of	sdram	and hyperram for
define	symbol	m_sdram_star	rt	=	isdefinedsymbol
define	symbol	m_sdram_size	Э	=	isdefinedsymbol
define	symbol	m_hyperram_s	start	=	isdefinedsymbol
define	symbol	m_hyperram_s	size	=	isdefinedsymbol
define	symbol	m_flash_star	rt	=	0x04000000;
define	symbol	m_flash_size	Э	=	0x00800000;

Figure 4. Flash start address change

• Burn the FlexSPI instance2 boot fuse.

DOOT OF ONT	YOR INSTANCE	0 - FlexSPI1	0
BOOT_CFG2[7]	XSPI_INSTANCE	1 - FlexSPI2	

Figure 5. FlexSPI instance fuse

To create a "new workspace" in the MCUXpresso secure provisioning tool, perform the following steps:
 a. Open the MCUXpresso secure provisioning tool.

- b. Select "choose file" -> "new workspace".
- c. Select the MIMXRT1189 option button.

New Workspace	×
Workspace: C:\lxtdoc\project\RT1180_Boot	∽ Browse
Series Processor O KW45xx/K32W1xx O MIMXRT1165 O LPC55Sxx/NHS52Sxx MIMXRT1166 O MC56F818xx MIMXRT1166 O MC56F818xx MIMXRT1171 O MC56F818xx MIMXRT1172 O MCX N94x/N54x/N23x MIMXRT1172 O MCY MIMXRT1173 O MWCT2xD2 MIMXRT1173 O RW61x MIMXRT1175 O I.MX RT10xx MIMXRT1176 I.MX RT11xx MIMXRT1181 O I.MX RT5xx MIMXRT1182 O MC56F817xx/6xx MIMXRT1189 O MC56F817xx/6xx MIMXRT1189	

Figure 6. Create a "new workspace" in the MCUXpresso secure provisioning tool

- 3. To build the image via the MCUXpresso secure provisioning tool, perform the following steps:
 - a. Choose the image generated by IAR.
 - b. Click the **Build image** button.

NXP Semiconductors

AN14589

Boot the i.MX RT1180 and Test the Boot Time from Different Devices

🐓 Build image 🔺 Write image 🦄	PKI management		
Source executable image:	C:\lxtdoc\2_SDK\SDK_2_15_001_MIMXRT1180-EVK\boards\evkmimxrt	1180\demo_apps\h€ ∨ Browse	✤ Build image
Start address:	0x2800B000 Application image; XIP: yes	 Additional images 	Generated files:
Use custom output file path:	bootable_images\hello_world_demo_cm33.bin	~ Browse	<u>* application ahab.</u> <u>* build image win.b</u>
ELE firmware:		✓ Browse	<u>* unsigned MIMXRT11</u> <u>* unsigned MIMXRT11</u>
Versions:	Image version: 0 🗸 Dual image boot	 Firmware versions 	<u>* write parameters.</u>
XMCD: None ~	🖌 Edit	✓ Browse	Build script hooks:
Authentication key:			<u>pre build win. bat</u> build win. bat
DEK key:		🖎 Random	
AHAB encryption algorithm:	V Key id: 0		
🗸 OTFAD encryption 🖌 IE	encryption VOTP configuration		
Log			Detach
Stature of the countries Second	A		
Status of the operation. Success.	A new workspace is created. U. Lixtacolprojectlalliou_boot		

Figure 7. Build image via MCUXpresso secure provisioning tool

- 4. To write the image via the MCUXpresso secure provisioning tool, perform the following steps:
 - a. Ensure that the MIMXRT1180-EVK board is in the serial Download mode SW5 (0001).
 - b. Click the Write image button.

NXP Semiconductors

AN14589

Boot the i.MX RT1180 and Test the Boot Time from Different Devices

CUXpresso Secure Provisioning Tool version 9.0 - C:\htdoc\project/RT1180_Boot - × *** rise tools Help *** *********************************				
Bile Target Tools Help MINUTETING via UDBB bod Unsigned from FlexSPI NOR - simplified LC OEM Open Dbg None Build image V Write image PKI management Botable image to be writen Use built image Image path: bootable; images/hello, world_demo_cm33.bin Write ange path: bootable; images/hello, world_demo_cm33.bin V Frite size Write anges/hello Additional required files Title Path Unsign flashloader (bb., bootable; images/unsigned_MIMXKT1199_flashloade., Flashloader (bootable RAM image) used if processor is unsecured Parameters (json) config/write_parameters.json Parameters needed in write and fuses to be burnt by write script (or shadow registers) Advanced Start flashloader (Chetable; images/unsigned_MIMXKT1189_flashloade, flashloader_boot device Og Write a bootable image to the target boot device O the ab	🔀 *MCUXpresso Secure Pro	ovisioning Tool version 9.0 - C:\lxtdoc\project\RT1180_Boot		– 🗆 X
MMMXRT1189 vis USB Boot Unsigned from FlesSPI NOR-simplified LC: OEM Open Dbg: None Build image & Write image P KII management Bootable image to be writem Image path: bootable_images/unsigned If the image winkbat Write image winkbat If the image winkbat Write script hooks: vist is winkbat Description Table Path Description Unsigned flashloader (bin. bootable_images/unsigned_MMXRT1189_flashloade Flashloader (bootable RAM image) used if processor is unsecured Write a bootable image to the target boot device Parameters needed in write and fuses to be burnt by write script (or shadow registers) Vrite a bootable image to the target boot device Start flashloader: Create manufacturing package	File Target Tools Help			
Build image Write image PKI management Botable image to be written Image path:	MIMXRT1189 via USB E	loot Unsigned V from FlexSPI NOR	- simplified LC: OEM Open Dbg: None	
Bootable image to be written Use built image Image path: bootable_images\hello_world_demo_cm33.bin	🗸 Build image 🦆 Write	image 🖌 PKI management		
Use built image Image path: bootable_images\hello_world_demo_cm33.bin	Bootable image to be writt	en		**
Image path: bootable_images\hello_world_demo_cm33.bin Browse Write script hooks: Additional required files Title Path Description Unsigned flashloader (bi bootable_images\unsigned_MIMXRT1189_flashloade Flashloader (bootable RAM image) used if processor is unsecured write parameters (json) configs\write_parameters.json Parameters needed in write and fuses to be burnt by write script (or shadow registers) Advanced	🗹 Use built image			V Write image
Additional required files Title Path Unsigned flashloader (bin. bootable_images\unsigned_MIMXRT1189_flashloade Flashloader (bootable RAM image) used if processor is unsecured write parameters (json) configs\write_parameters.json Parameters needed in write and fuses to be burnt by write script (or shadow registers) Advanced	Image path: boo	table_images\hello_world_demo_cm33.bin	V Browse	<u>^write image win.bat</u>
Additional required files Title Path Unsigned flashloader (bim. bootable_images\unsigned_MIMXRT1199_flashloade Flashloader (bootable RAM image) used if processor is unsecured write parameters (json) configs\write_parameters.json Parameters needed in write and fuses to be burnt by write script (or shadow registers) Advanced				Write script hooks:
Additional required files Title Path Description Unsigned flashloader (Jsi bootable_images/unsigned_MIMXRT1189_flashloade Flashloader (bootable RAM image) used if processor is unsecured write parameters (json) configs/write_parameters.json Parameters needed in write and fuses to be burnt by write script (or shadow registers) Advanced				write win. bat
Title Path Description Unsigned flashloader (.bi bootable_images\unsigned_MIMXRT1189_flashloade Flashloader (bootable RAM image) used if processor is unsecured write parameters (.json) configs\write_parameters.json Parameters needed in write and fuses to be burnt by write script (or shadow registers) Advanced	Additional required files			
Unsigned flashloader (bin. bootable_images/unsigned_MIMXRT1189_flashloader Flashloader (bootable RAM image) used if processor is unsecured write parameters (json) configs/write_parameters.json Parameters needed in write and fuses to be burnt by write script (or shadow registers) Advanced	Title	Path	Description	
write parameters (json) configs\write_parameters.json Parameters needed in write and fuses to be burnt by write script (or shadow registers) Advanced	Unsigned flashloader (.bi	bootable_images\unsigned_MIMXRT1189_flashloade	Flashloader (bootable RAM image) used if processor is un	secured
Advanced Start flashloader Create manufacturing package og Write a bootable image to the target boot device Detach mprinage successed Write a bootable image to the target boot device Detach "C'luttoo'project KT1180_Boot\bootable_images/unsigned_UDUKT1189_flashloader_ahab_bootable.yaml" -o "C'luttoo'project KT1180_Boot\bootable_images/unsigned_UDUKT1189_flashloader_bin "Success. (Mtrged image: C'luttoo'project KT1180_Boot\bootable_images/unsigned_UDUKT1189_flashloader_bin created.) mprinage succeseded ### Create AMB image ### mprinage succeseded ### Create AMB image ### mprinage succeseded ### ESULT of the soript "building image": Success (return code = [0]SUCCESS) Status of the operation: Success: Building image tatus of the last operation. Success: Building image	write parameters (.json)	configs\write_parameters.json	Parameters needed in write and fuses to be burnt by write	script (or shadow registers)
og Write a bootable image to the target boot device Detach Ampinange successed Ampinange su	< Advanced Start flashloader	Create manufacturing package		>
og Detach mapimage successes Building image taus of the last operation: Success: Building image taus of the last operation		Write a bootable	image to the target boot device	
<pre>map:mage successed map:mage successed map:mape:mage successed map:mage successed map:mape:mage successed map:mage successed map:ma</pre>	.og			Detach
tatus of the last operation: Success: Building image	nxpimage subcedud nxpimage utils binary-im. "C:\lxtdoc\project\RT1180 Success. (Merged image: / nxpimage succeeded ### Create AHAB image ##	age merge -o "C:\lxtdoc\project\RT1180_Boot\configs D_Boot\bootable_images\unsigned_MDMRT189_Flashload C:\lxtdoc\project\RT1180_Boot\bootable_images\unsign #	unsigned_MUMRT1189_flashloader_ahab_bootable.yaml″ er.bin″ ed_MUMRRT1189_flashloader.bin created.)	-•
	nxpimage ahab export — Success. (AHAB: C:/lxtdo nxpimage succeeded ### RESULT of the script Status of the operation:	"C:\lxtdov\project\XII180_Boot\configs\application_ c/project/XII180_Boot/bootable_images/hello_world_de `Building image`: Success (return code = [0]SUCCESS Success: Building image	hab.yaml mo_om33.bin created.))	v

Figure 8. Write image via MCUXpresso secure provisioning tool

- 5. To debug message on the debug console, perform the following steps:
 - a. Switch to the internal Boot mode SW5 (0100).
 - b. Press the power reset (POR) button on the MIMXRT1180-EVK, "hello world" can be seen on the serial debug console.



Figure 9. debug message on debug console

3.2 Boot from FlexSPI NAND flash

- 1. Because the FlexSPI NAND flash cannot support the XIP directly, to build the non-XIP image, perform the following steps:
 - a. Build the image.

- b. Change flexspi_nor_debug to debug.
- c. Figure 10 shows the settings of the Output Converter.

14	Workspace	hello_world.c x main() 1 /* 2 * Copyright (c) 2013 - 3 * Copyright 2016-2017 I 4 * All rights reserved. 5 * SPDY-Licence-Identif: X
	Category: General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Custom Build Linker Build Actions Debugger Simulator CADI CMSIS DAP E2/E2 Lite GDB Server G+LINK I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET	I33.srec

Figure 10. Build the non-XIP image

- 2. To test the FlexSPI NAND flash, perform the following steps:
 - a. Switch to the serial Download mode.
 - b. Click boot memory configuration and select FlexSPI NAND.
 - c. Click the $\ensuremath{\text{Test}}$ button. If the connection is correct, the test is successful.

Boot the i.MX RT1180 and Test the Boot Time from Different Devices

Boot Memory Configuration										×
Boot memory type	Boot memory config	juration paramet	ters							
O FlexSPI NOR - simplified	NAND option0			FCB option			Image copies			
FlexSPI NOR - complete FCB	Flash size:	1Gb	\sim	Search count:	1	\sim	Block index:	Block count:	Resulting word:	
FlexSPI NAND	Has multiplanes:	1 plane	~	Search stride:	64 pages	~	2	8	0x00020008	
O eMMC	Description									
O SD card	Pages per block:	64	~	Address type:	Byte address	~				
O XSPI NOR - simplified	Page size:	2KB	\sim	Image copies:	1	\sim				
O ASPI NUK - complete FCB	Max frequency:	60MHz	~							
	Has option1:	No	~							
O SEMC NAND										
Onchip RAM										
	Resulting word:	0xC0010023		Resulting word:	0xC1000003					
MT29F1G01ABAFD12 V Apply										
	NAND option1									
User configuration	Manufacturer id:	0								
Srowse										
Load Save	Resulting word:									
Protected area										
Enable protection	Comment									
Start address: 0x00000000 ~	The configuration	is loaded from t	the pred	efined template SDI	HC eMMC 8GB U	SDHC2				^
Size: 0x00001000 🗸	The configuration	is loaded from t	the pred	efined template SDI	HC SD-card 8GB U	JSDHC1				
Comment/reason:										~

Figure 11. FlexSPI NAND flash configuration in MCUXpresso secure provisioning tool

3. To debug the NAND flash debug message on the debug console, perform the following steps:

- a. Switch the Boot mode to 101 (serial NAND flash with 2K page via FlexSPI1, primary group, PortA).
- b. Press the POR button on the MIMXRT1180-EVK, "hello world" can be seen on the serial debug console.





3.3 Boot from eMMC

- 1. The eMMC boot image is a non-XIP image. Build the image similar to FlexSPI NAND flash. For more details, see <u>Section 3.2</u>.
- 2. To configure the eMMC, perform the following steps:
 - a. Set the Boot mode to serial download.

- b. Open the MCUXpresso secure provisioning tool and click "Boot Memory Configuration".
- c. Select the SDHC eMMC 8GB USDHC2 and click the Apply button.
- d. Click the **Test** button. If the connection is correct, the test is successful.

Boot Memory Configuration					- 0	×
Boot memory type	Boot memory configuration p	arameters				
O FlexSPI NOR - simplified	eMMC option0			eMMC option1		
O FlexSPI NOR - complete FCB	Enable boot config:	No	\sim	Instance selection:	USDHC2	~
○ FlexSPI NAND	Enable boot ack:	No ACK	~	Enable 1.8V	Dirable	
● eMMC	Enable boot ack:	IND ACK	~	Enable 1.ov:	Disable	~
○ SD card	Reset boot bus conditions:	Reset to 1bit, SDR, Normal	\sim	Enable power cycle:	Disable	~
O XSPI NOR - simplified	Boot mode:	Normal	\sim	Power up time:	5 ms	1
O XSPI NOR - complete FCB	Bus width:	1bit SDR	~	PWR polarity:	Power down when RST lov	~
Onchip flash	Tincing interferen	NI I		DMD dawn tinsau	20	
O IFR flash	liming interface:	INORMAI	~	PWK down time:	20 ms	~
O SEMC NAND	Boot bus width:	1bit SDR, 4bit DDR	\sim			
O Onchip RAM	Enable boot partition:	Not enabled	\sim			
Predefined template	Partition access:	User data area	\sim			
SDHC eMMC 8GB USDHC2 V Apply	/					_
User configuration	Resulting word:	0xC0000000		Resulting word:	0x0000002	
✓ Browse.	Parameters					
Load Save	Memory size [bytes]:	0x1c8000000				
Protected area	Frank black size (buter).	0.200	=			
Enable protection	Erase block size [bytes]:	0x200				
Start address: 0x00000000	Comment					
Size: 0x00001000	The configuration is loaded	from the predefined template	SDHC eN	MMC 8GB USDHC2		^
Comment/reasons						~
Comment/reason:	Test the configuration					
	Test not test	ed yet				

Figure 13. eMMC configuration in MCUXpresso secure provisioning tool

- 3. To debug the eMMC debug message on the debug console, perform the following steps:
 - a. Build and write the image to the eMMC.
 - b. Switch the Boot mode from serial download to 010 (eMMC 8 bits via uSDHC2).
 - c. Press the POR button on the MIMXRT1180-EVK, "hello world" can be seen on the serial debug console.



3.3.1 eMMC boot configurations

The i.MX RT1180 platform has three Boot mode pins. To boot from eMMC, set the Boot mode pins[0:2] to 010. The eMMC with bus width 8 bits boots the i.MX RT1180 via uSDHC2 boot device, which has fixed instance and bus width. If you need other configurations like eMMC bus speed mode, burn the fuse.

The following are the options to boot from eMMC via uSDHC1:

- 4-bits mode
- 4-bits DDR mode
- 8-bits DDR mode

The following is the example of booting from eMMC with bus width 4 bits via uSDHC2:

- 1. Download the image to the eMMC via the MCUXpresso secure provisioning tool.
- 2. Burn the fuse BOOT_CFG0[2:0] to 2, which selects the eMMC.
- 3. Burn the fuse BOOT_CFG0[6] to 1, the selected fuse configuration is valid for ROM booting.
- 4. Burn fuse BOOT_CFG4[0] to 1, the selected uSDHC2 is used for the eMMC boot. If you boot from the fuse, the default eMMC bus width is 4 bits. <u>Table 4</u> shows the fuse configuration when booting from eMMC with bus width 4 bits via uSDHC2.

	j. en en e	
FUSE	Definitions	Settings
BOOT_CFG0[2:0]	BOOT_MODE_FROM_FUSE	eMMC
BOOT_CFG0[6]	BT_FUSE_SEL	Fuse configuration is valid for ROM booting
BOOT_CFG4[0]	USDHC_PORT	uSDHC2 is used for the eMMC boot

Table 4.	Fuse	configuration	when	booting	from	eMMC
	i use	configuration	WIICII	booting	nom	CIAIIAIC

- 5. Switch the Boot mode to 000 (boot from internal fuses).
- 6. Press the POR button on the MIMXRT1180-EVK and the boot is successful.

3.4 Boot from the SD card

- 1. The SD card cannot XIP. Build a non-XIP image first.
- 2. To configure the SD card, perform the following steps:
 - a. Set the Boot mode to serial download.
 - b. Open the MCUXpresso secure provisioning tool and click "Boot Memory Configuration".
 - c. Select the **SD card**.
 - d. Click the **Test** button. If the connection is correct, the test is successful.

Boot Memory Configuration			— D	\times
Boot memory type	-Boot memory configuration	n parameters		
O FlexSPI NOR - simplified	SD option			
O FlexSPI NOR - complete FCB	Instance selection:	USDHC1 ~		
O FlexSPI NAND	D	44.5		
⊖ eMMC	Bus width:	1bit ~		
SD card	Timing interface:	SDR12/Normal ~		
🔿 XSPI NOR - simplified	Enable power cycle:	No ~		
🔿 XSPI NOR - complete FCB	Device a clasifica	DCT I and the block of the		
Onchip flash	Power polarity:	KST low-disable V		
○ IFR flash	Power up time:	5ms 🗸		
○ SEMC NAND	Power down time:	20ms ~		
Onchip RAM				
Predefined template	Resulting word:	0xD0000001		
✓ Apply	Parameters			
User configuration	Memory size [bytes]:	0x3b5980000		
V Browse	Erase block size (bytes):	0x200		
Load Save				
Destanted even				
Frotected area				
	Comment			
Start address: 0x0000000 V	The configuration is load	led from the predefined templa	te SDHC eMMC	^
Size: 0x00001000 ~	The configuration is load	led from the predefined templa	te SDHC SD-card	
Comment/reason:	8GB USDHC1			~
^	Test the configuration			
× .	Test not t	tested vet		
		-		
			ОК С	ancel

Figure 15. SD card configuration in MCUXpresso secure provisioning tool

- 3. To debug the SD card debug message on the debug console, perform the following steps:
 - a. Build and write the image to the SD card.
 - b. Switch the Boot mode to 011 (SD 4 bits via uSDHC1).
 - c. Press the POR button on the MIMXRT1180-EVK, "hello world" can be seen on the serial debug console.



Figure 16. SD card debug message on debug console

The i.MX RT1180 platform has three Boot mode pins. To boot from the SD card, set the Boot mode pins[0:2] to 011. The SD card with bus width 4 bits boots the i.MX RT1180 via uSDHC1 boot device, which has fixed instance and bus width. If you need other configurations, burn the fuse.

3.5 Recovery boot

1. The recovery boot starts when the primary boot fails. Burn the RECOVERY_BOOT_EN fuse.

1 – Recovery boot is enabled	E	BOOT_CFG0[4]	RECOVERY_BOOT_EN	0 – Recovery boot is disabled	0
				1 – Recovery boot is enabled	

Figure 17. RECOVERY_BOOT_EN fuse

2. Build the non-XIP image via IAR. The LPSPI flash cannot XIP. Download the image via blhost as the MCUXpresso secure provisioning tool does not support LPSPI flash boot.

Ipspi_flash_boot_image.srec	10/9/2024 5:05 PM	SREC File	56 KB	
-----------------------------	-------------------	-----------	-------	--

Figure 18. IAR build non-XIP image

3. Add the container to the non-XIP image as follows:

- a. In the MCUXpresso secure provisioning tool, select the image to build.
- b. Click the **Build image** button.

NXP Semiconductors

AN14589

Boot the i.MX RT1180 and Test the Boot Time from Different Devices

✓ Build image ✓ Write image ✓ PK	management	ropen bog none			
Source executable image:	:\lxtdoc\3_case\76_RT1180_Boot_time_test\boot_t	ime_test_rt1180_cm33\Debug\lp	spi_fla ∨ Brows	e 🗸 Buil	d image
Select Source image					×
← → × ↑ 📙 « 3_case → 76_RT11	80_Boot_time_test > boot_time_test_rt1180_cm3	3 > Debug > 🗸 🧹	Search Debug	م ا	ahab. ys
Organize 🔻 New folder				==	MXRT1189
	Name	Date modified	Туре	Size	MXRT1189 eters.js
🖈 Quick access	Browselnfo	10/9/2024 5:05 PM	File folder		les
NXL66653	obj	9/29/2024 5:05 PM 9/29/2024 10:30 AM	File folder	קע נדג	oks:
🧊 3D Objects	boot_time_test_rt1180_cm33.srec	10/9/2024 2:55 PM	SREC File	56 KB	<u>.bat</u>
Desktop Decuments	Ipspi_flash_boot_image.srec	10/9/2024 5:05 PM	SREC File	56 KB	
Downloads					
b Music					
Pictures					
Videos					
ECL UR (\ncl cha ava com) (O)					Detach
Network					
File name: Ipspi_flas	h_boot_image.srec		✓ All Source in	nage (*.axf;*.bin;*.e ~	
			Open	Cancel	

Figure 19. Build image via MCUXpresso secure provisioning tool

4. Switch to the Write image page. Click the Create manufacturing package ... button.

Image path:	bootal	ole_images\lpspi_flash_boot_image.bin	~	Browse	write image win.bat Write script hooks: write win.bat
Additional required fil	es	D.1			
litle		Path	Description		
Unsigned flashloade	r (.bi	bootable_images\unsigned_MIMXRI1189_flashloade	Flashloader (bootable RAM image) used if	processor is uns	ecured
write parameters (.js	onj	configs\write_parametersJson	Parameters needed in Write and tuses to be	e burnt by write	script (or snadow régisters)
<					>

Figure 20. Create the manufacturing package

5. Find the image after adding the container in manufacturing package.

		- 3	+
Name	Date modified	Туре	Size
Ipspi_flash_boot_image.bin	10/9/2024 5:15 PM	BIN File	27 KB
unsigned_MIMXRT1189_flashloader.bin	10/9/2024 5:15 PM	BIN File	98 KB

 For more details, see "Layout for Serial NOR Flash/E2PROM" in *i.MX RT1180 Reference Manual* (document <u>IMXRT1180RM</u>). Add MCB to the image. In this case, add 1024 zeros before the image.

File	Edit	Sei	arch		ew	For	mat	Sc	ripts	Te	mpl	ates	De	ebug	Pr	roject	Tools	Win	dow	He	lp		
	~ 📂			,	5		1	1	‰	Þ	Î	Ľ) (ן ע	۶	Å₿	🤛 🔶	1	5	3	Hex	÷ ۲	Π
Startu	р	lps	oi_fla	ish_b	poot_	ima	ge.bi	in X															
	ð	1	2	3	4	5	6	7	8	9	А	В	С	D	Е	F	012345	678	9ABC	DEF			
0380	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00							
0390	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00							
03A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00							
03B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00							
03C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00							
03D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00							
03E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00							
03F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00							
0400	00	A0	00	87	00	00	00	00	00	00	00	01	90	00	00	00	· ·‡··						
0410	00	20	00	00	00	4A	00	00	00	00	FE	0F	00	00	00	00	J		.þ				
0420	00	00	FE	0F	00	00	00	00	13	00	00	00	00	00	00	00	þ						
0430	7E	EC	F3	10	BA	E2	CD	0A	EF	E7	5C	93	94	9B	15	C2	~10.°ä	1.1	ç\""	2.A			
0440	18	2B	2D	AC	24	B2	39	56	D2	8E	B9	BC	EE	C3	EE	5E	.+-¬\$²	9002	Z'%1	AI^			
0450	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00							
0460	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00							
0470	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00							
0480	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00							
0490	00	10	00	90	00	00	00	00	00	00	00	00	00	00	00	00							
04A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00							
04B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00							
0400	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00							
04D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00							

Figure 22. Add 1024 zeros before the image

7. Download the image to the LPSPI flash via blhost. Figure 23 shows the blhost command.

Blhost cmd burn image to spi flash

cd C:\Users\nxf90529\Desktop\blhost_2.6.7\bin\win blhost.exe -u 0x1fc9,0x014c -- get-property 1 blhost.exe -u 0x1fc9,0x014c -- load-image ../../flashloader/flashloader_utility.bin blhost.exe -u 0x15A2,0x0073 -- get-property 1 blhost.exe -u 0x15A2,0x0073 -- fill-memory 0x20000000 4 0xc1100000 blhost.exe -u 0x15A2,0x0073 -- fill-memory 0x20000004 4 0x00000000 blhost.exe -u 0x15A2,0x0073 -- fill-memory 0x20000004 4 0x00000000 blhost.exe -u 0x15A2,0x0073 -- fill-memory 0x110 0x20000000 blhost.exe -u 0x15A2,0x0073 -- flash-erase-region 0x0 0x20000 0x110 blhost.exe -u 0x15A2,0x0073 -- get-memory 0x0 ../../image/lpspi_flash_boot_image.bin 0x110 blhost.exe -u 0x15A2,0x0073 -- read-memory 0x0 0x400 0x110

Figure 23. Blhost command to download the image

- 8. To debug the eMMC debug message on the debug console, perform the following steps:
 - a. Ensure that there is no image in XIP flash.
 - b. Switch the Boot mode to 0100 (serial NOR flash).
 - c. Press the POR button on the MIMXRT1180-EVK and the boot is successful.

AN14589



4 Test the boot time from different boot devices

This section tests the boot time from different boot devices.

4.1 Method for testing the boot time

The boot time is measured from the point of 65 % of POR_B to the point of running to "SystemInit" of apps code.

The POR_B test point on MIMXRT1180-EVK is J92-2. Figure 25 shows the POR_B test point.



Figure 25. POR_B test point on MIMXRT1180-EVK

The general-purpose input/output (GPIO) in "systemInit" of apps code toggles so that the oscilloscope can capture it.

Figure 26 shows the end time test point.



4.2 The boot time for different boot devices

Test the Quad serial peripheral interface (SPI) flash boot time with the images size of 8626 bytes and 412122 bytes. The start time and the end time test points are the same as given in <u>Section 4.1</u>.

1. The boot time for the 8626 bytes image is about 18.9 ms.



Figure 27. Test the Quad SPI flash boot time for 8626 bytes

2. The boot time for the 412122 bytes image is about 26.3 ms.



Figure 28. Test the Quad SPI flash boot time for 412122 bytes

3. <u>Table 5</u> shows the boot time for other boot devices.

 Table 5. The boot time for different boot devices with different sizes

Boot devices	Image size	Boot time	Image size	Boot time
Quad SPI flash	8626 bytes	18.9 ms	412122 bytes	26.3 ms

Boot devices	Image size	Boot time	Image size	Boot time
FlexSPI NAND flash	8799 bytes	22.89 ms	368862 bytes	41.8 ms
eMMC 4 bits	8806 bytes	247 ms	368870 bytes	263 ms
eMMC 8 bits	8806 bytes	243 ms	368870 bytes	251 ms
SD card	8806 bytes	129 ms	368866 bytes	140.5 ms
LPSPI flash	12118 bytes	22.9 ms	132142 bytes	60.5 ms

Table 5. The boot time for different boot devices with different sizes ... continued

5 Conclusion

This document outlines the following details:

- i.MX RT1180 system boot
- i.MX RT1180 booting from the different boot devices
- Use of MCUXpresso secure provisioning tool
- · Method to test the boot time from different boot devices

For more details, see *i.MX RT1180 Reference Manual* (document <u>IMXRT1180RM</u>).

6 Acronyms

Table 6 lists the acronyms used in this document.

Term	Description
AES	Advanced encryption standard
АНАВ	Advanced high assurance boot
CBC	Cipher block chaining
DCD	Device configuration data
EEPROM	Electrically erasable programmable read-only memory
eMMC	embedded multi-media card
ESC	EtherCAT slave controller
GPIO	General-purpose input/output
IEE	Inline encryption engine
LPSPI	Low-power serial peripheral interface
LPUART	Low-power universal asynchronous receiver/transmitter
OTFAD	On-the-Fly AES decryption
POR	Power reset
PSRAM	Pseudostatic random-access memory
RAM	Random-access memory
SDK	Software development kit
SDRAM	Synchronous dynamic random-access memory
SEMC	Smart external memory controller

Table 6. Acronymscontinued		
Term	Description	
SPI	Serial peripheral interface	
TSN	Time sensitive networking	
UART	Universal asynchronous receiver/transmitter	
XIP	Execute-in-place	
XMCD	External memory configuration data	

7 References

Table 7 lists the references used to supplement this document.

Table 7. Related documentation/resources

Document	Link/how to access
<i>i.MX RT1180 Reference Manual</i> (document IMXRT1180RM)	IMXRT1180RM

8 Note about the source code in the document

Sample code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2025 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials must be provided with the distribution.
- 3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

9 Revision history

Table 8 summarizes the revisions to this document.

Table 8. Revision history

Document ID	Release date	Description
AN14589 v.1.0	04 March 2025	Initial public release

AN14589	
Application	note

Boot the i.MX RT1180 and Test the Boot Time from Different Devices

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.

AN14589

Boot the i.MX RT1180 and Test the Boot Time from Different Devices

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. EdgeLock — is a trademark of NXP B.V. IAR — is a trademark of IAR Systems AB.

Boot the i.MX RT1180 and Test the Boot Time from Different Devices

Contents

1	Introduction	2
2	i.MX RT1180 boot overview	2
2.1	Boot feature	2
2.2	Serial downloader	2
2.3	External memory configuration data	3
2.4	Secure boot	3
2.5	Boot ROM features	3
2.6	Boot mode selection	3
2.7	Boot mode pin settings	4
2.8	Boot devices	4
2.9	Boot image	5
2.10	MCUXpresso secure provisioning tool	6
3	Boot i.MX RT1180 from different boot	
	devices	6
3.1	Boot from Quad SPI flash via FlexSPI	6
3.2	Boot from FlexSPI NAND flash	10
3.3	Boot from eMMC	12
3.3.1	eMMC boot configurations	14
3.4	Boot from the SD card	15
3.5	Recovery boot	16
4	Test the boot time from different boot	
	devices	19
4.1	Method for testing the boot time	19
4.2	The boot time for different boot devices	20
5	Conclusion	21
6	Acronyms	21
7	References	22
8	Note about the source code in the	
	document	22
9	Revision history	22
	Legal information	23

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: 4 March 2025 Document identifier: AN14589