Application note

Migrating from STM32F469/479 line to STM32MP151, STM32MP153 and STM32MP157 lines

Introduction

The STM32MP151, STM32MP153 and STM32MP157 devices are part of the STM32 Arm® Cortex® MPUs subclass; they all feature a Cortex®-M4 and depending on the part number, either a single-core or a dual-core Cortex®-A7. These devices are referred to as *STM32MP15x* in this document.

The Cortex®-M4 inside the STM32MP15x devices is compatible (for STM32Cube package) with the STM32F469/479 line devices. This compatibility permits an easy migration from an STM32F469/479 design to a similar device from the STM32MP15x lines and to benefit from their significantly higher integration and their advanced peripherals without adding any additional complexity.

The STM32MP15x high performance Cortex®-A7 runs open operating systems like Linux, which provides rich connectivity and the support of a software community.

This application note provides information to facilitate the migration from an STM32F469/479 design towards an STM32MP15x design.

1 STM32MP15x lines overview

The STM32MP15x devices are part of the STM32MP1 Series. Depending on the device's part number, the system includes a Cortex®-M4 and either a single-core or a dual-core Cortex®-A7.

The full featured system (see the table below) is partitioned in:

- One MPU subsystem: dual Cortex-A7 with L2 cache
- One MCU subsystem: Cortex-M4 with associated peripherals clocked according to CPU activity

Table 1. Configuration of the lines of the STM32MP1 Series

Lines	Reference manual	Cortex-A7 configuration	Cortex-M4	GPU	DSI	FDCAN
STM32MP151	RM0441	Single-core	Yes	No	No	No
STM32MP153	RM0442	Dual-core	Yes	No	No	Yes
STM32MP157	RM0436	Dual-core	Yes	Yes	Yes	Yes

The STM32MP15x lines offer extra performance compared to the STM32F469/479 devices.

The STM32MP15x devices include a larger set of peripherals with advanced features and higher system integration compared to the STM32F469/479 devices such as:

- Dual-core Arm® Cortex®-A7 subsystem
- 3D graphic processing unit (GPU)
- External LPDDR2/LPDDR3/DDR3/DDR3L 16- or 32-bit interface
- Security related peripherals (ETZPC, TZC, BSEC, OTP)
- Gigabit Ethernet MAC interface (ETH1)
- FD controller area network (FDCAN1/FDCAN2) including one TTCAN
- Voltage reference buffer (VREFBUF)
- SPDIF receiver interface (SPDIFRX)
- High-definition multimedia interface - consumer electronics control (HDMI-CEC)
- Universal serial bus high-speed host with two ports (USBH)
- Two embedded USB 2.0 High-Speed PHY (USBPHYCTRL)
- Digital filter for sigma-delta modulators (DFSDM1)
- 8- to 16-bit analog-to-digital converters (ADC1/ADC2)
- DMAMUX extension to DMA1/DMA2
- Master direct memory access (MDMA)
- Hardware semaphore (HSEM)
- Low-power timer (LPTIM1/LPTIM2/LPTIM3/LPTIM4/LPTIM5)
- Digital temperature sensor (DTS)

This migration guide covers the migration from STM32F469/479 towards STM32MP15x devices. The new features present on the STM32MP15x lines but not already present on the STM32F469/479 line are not covered in this document (refer to the STM32MP15x lines reference manuals and datasheets for more details).

There are some features available on STM32F469/479 line that are not present in the SMT32MP15x lines. These features are listed below:

- DMA2D
- Internal Flash

This document applies to Arm®-based devices.

Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

2 Hardware migration

There is no exact compatible package between SM32F469/479 line and STM32MP15x lines, but a good candidate for migration towards an STM32MP15x device can be chosen by considering the following criteria:

- GPIO: minimum number of available GPIOs. Precise count should be done for each application use case.
- Size: the package size (from 10×10 to 18×18)
- PCB: the PCB technology cost (TFBGA pitch 0.5 or LFBGA pitch 0.8)
- Perf (performance): the DDR bus width (16- or 32-bit) which is linked to the maximum Cortex-A7 performances.

The table below presents a cross reference to assist the user to choose the closest migration candidate in number of GPIO, but also according to some different criteria priority (smallest package size or lowest PCB cost or best performances).

Table 2. Cross-selector based on GPIO count and package size

Product	GPIO	Package	Size (mm)	Ball pitch (mm)	Priority	Closer part	GPIO	Package	Size (mm)	Ball pitch (mm)	DDR bus width
STM32F4x9Vx	71	LQFP100	14×14	-	size	STM32MP15xxAD	98	TFBGA257	10×10	0.5	16
					pcb	STM32MP15xxAB	98	LFBGA354	16×16	0.8	16
					perf	STM32MP15xxAC	148	TFBGA361	12×12	0.5	32
					perf	STM32MP15xxAA	176	LFBGA448	18×18	0.8	32
STM32F4x9Zx	106	LQPF144	20×20	-	size	STM32MP15xxAD	98	TFBGA257	10×10	0.5	16
					pcb	STM32MP15xxAB	98	LFBGA354	16×16	0.8	16
					perf	STM32MP15xxAC	148	TFBGA361	12×12	0.5	32
					perf	STM32MP15xxAA	176	LFBGA448	18×18	0.8	32
STM32F4x9Ax	114	UFBGA169	7×7	0.5	size	STM32MP15xxAD	98	TFBGA257	10×10	0.5	16
					perf	STM32MP15xxAC	148	TFBGA361	12×12	0.5	32
	114	WLCSP168	4.89×5.69	0.4	size	STM32MP15xxAD	98	TFBGA257	10×10	0.5	16
					perf	STM32MP15xxAC	148	TFBGA361	12×12	0.5	32
STM32F4x9Ix	131	LQFP176	24×24	-	size	STM32MP15xxAC	148	TFBGA361	12×12	0.5	32
					pcb	STM32MP15xxAA	176	LFBGA448	18×18	0.8	32
	131	UFBGA176	10×10	0.65	size	STM32MP15xxAD	98	TFBGA257	10×10	0.5	16
					perf	STM32MP15xxAC	148	TFBGA361	12×12	0.5	32
STM32F4x9Bx	161	LQFP208	28×28	-	size	STM32MP15xxAC	148	TFBGA361	12×12	0.5	32
					pcb	STM32MP15xxAA	176	LFBGA448	18×18	0.8	32
STM32F4x9Nx		TFBGA216	13×13	0.8	size	STM32MP15xxAC	148	TFBGA361	12×12	0.5	32
					pcb	STM32MP15xxAA	176	LFBGA448	18×18	0.8	32

3 Boot modes selection

STM32F469/479 devices usually start executing code from embedded Flash memory or internal SRAM, the selection is done using boot pins:

- Boot from user Flash memory
- Boot from system memory (used to program the user Flash memory)
- Boot from embedded SRAM (mostly for debug phases)

The STM32MP15x devices always start from internal BootROM. Internal BootROM starts based on boot pins and on internal OTP fuses.

- Boot from external Flash:
  - SD-Card (SDMMC1)
  - *e*•MMC (SDMMC2)
  - SLC-NAND (FMC)
  - Serial NOR-Flash (QUADSPI)
  - Serial NAND Flash (QUADSPI)
- Boot from UART or USB OTG on High-Speed PHY port #2
  - Used to access the device from STM32CubeProgrammer (STM32CubeProg) for example program the external Flash or internal OTP fuses

Refer to STM32 wiki articles below:

- https://wiki.st.com/stm32mpu/index.php/Boot_chains_overview
- https://wiki.st.com/stm32mpu/index.php/STM32MP15_ROM_code_overview

4 Peripherals migration

This section presents a full view of the features and peripheral counts of STM32F469/479, STM32MP157, STM32MP153 and STM32MP151 lines, an analysis of peripheral cross compatibility between STM32F469/479 and STM32MP15x lines and a peripheral address mapping snapshot for the concerned products.

Table 3. STM32F469/479 line features and peripheral counts compared to STM32MP15x lines

Product		STM32F469/479 line												STM32MP157 line								STM32MP153 line								STM32MP151 line							
Peripherals		STM32F469Vx	STM32F479Vx	STM32F469Zx	STM32F479Zx	STM32F469Ax	STM32F479Ax	STM32F469Ix	STM32F479Ix	STM32F469Bx	STM32F479Bx	STM32F469Nx	STM32F479Nx	STM32MP157AAD	STM32MP157CAD	STM32MP157AAB	STM32MP157CAB	STM32MP157AAC	STM32MP157CAC	STM32MP157AAA	STM32MP157CAA	STM32MP153AAD	STM32MP153CAD	STM32MP153AAB	STM32MP153CAB	STM32MP153AAC	STM32MP153CAC	STM32MP153AAA	STM32MP153CAA	STM32MP151AAD	STM32MP151CAD	STM32MP151AAB	STM32MP151CAB	STM32MP151AAC	STM32MP151CAC	STM32MP151AAA	STM32MP151CAA
Flash memory in Kbytes		512 (only for STM32F469x)												Only external Flash [1] [2]																							
		1024																																			
		2048																																			
SRAM in Kbytes	System	384 (160+32+128+64)												448 (128+128+64+64+64) [3]																							
	Backup	4												4																							
FMC memory controller		Yes												Yes [4]																							
Quad-SPI		Yes												Yes																							
Ethernet		No						Yes						Yes		Yes				Yes				Yes						Yes							
Timers	General-purpose	10												10																							
	Advanced-control	2												2																							
	Basic	2												2																							
Random number generator		Yes												Yes																							
Communication interfaces	SPI / I2S	4/2 (full duplex)						6/2 (full duplex)						6/3 (full duplex)																									
	I2C	3												6																							
	USART/ UART	4/3						4/4						4/4																							
	USB OTG FS	Yes												Yes [6]																							
	USB OTG HS	Yes												Yes [7]																							
	CAN	2												2 (FDCAN)																No							
	SAI	1												4																							
	SDIO	Yes												Yes [8]																							
Camera interface		Yes												Yes																							

Product	STM32F469/479 line												STM32MP157 line								STM32MP153 line								STM32MP151 line							
Peripherals	STM32F469Vx	STM32F479Vx	STM32F469Zx	STM32F479Zx	STM32F469Ax	STM32F479Ax	STM32F469Ix	STM32F479Ix	STM32F469Bx	STM32F479Bx	STM32F469Nx	STM32F479Nx	STM32MP157AAD	STM32MP157CAD	STM32MP157AAB	STM32MP157CAB	STM32MP157AAC	STM32MP157CAC	STM32MP157AAA	STM32MP157CAA	STM32MP153AAD	STM32MP153CAD	STM32MP153AAB	STM32MP153CAB	STM32MP153AAC	STM32MP153CAC	STM32MP153AAA	STM32MP153CAA	STM32MP153AAD	STM32MP153CAD	STM32MP153AAB	STM32MP153CAB	STM32MP153AAC	STM32MP153CAC	STM32MP153AAA	STM32MP153CAA
MIPI-DSI host	Yes												Yes								No															
LCD-TFT	Yes												Yes																							
Chrom-ART Accelerator™ (DMA2D)	Yes												No, but 3D GPU								No															
Cryptography	Yes (only on STM32F479xx)												Yes (only on STM32MP15xCxx)																							
Hash	Yes (only on STM32F479xx)												Yes																							
GPIOs	71		106		114		131		161				98				148		176		98				148		176		98				148		176	
12-bit ADC	3												2 (up to 16-bits)																							
Number of ADC channels	14		20		24								17				22				17				22				17				22			
12-bit DAC	Yes												Yes																							
Number of DAC channels	2												2																							
Maximum Cortex-M4 frequency	180 MHz												209 MHz (10)																							
Operating voltage	1.7 to 3.6V												1.71 to 3.6V																							
Operating temperatures	Junction temperature: −40 to 125 °C												Junction temperature: −40 to 105 °C / −40 to 125 °C																							
Package	LQFP100	LQPF144	UFBGA169 / WLCSP168	LQFP176 / UFBGA176	LQFP208	TFBGA216	TFBGA257	LFBGA354	TFBGA361	LFBGA448	TFBGA257	LFBGA354	TFBGA361	LFBGA448	TFBGA257	LFBGA354	TFBGA361	LFBGA448																			

1. *Cortex-M4 only runs from internal SRAM. System could boot from external Flash on SD-Card, eMMC, Serial-NOR, Serial-NAND or 8-bit SLC-NAND.*
2. *Includes 1500 internal OTP bits available for user.*
3. *For Cortex-M4 (including 64 Kbytes of SRAM which could be retained in Stanbdy mode for early M4 wakeup). Cortex-A7 has in addition 256 Kbytes of SYSRAM.*
4. *NAND and PSRAM only, SDRAM on dedicated 16/32-bits DDR interface*

5. *Gigabit Ethernet*
6. *Only if OTG HS is not used.*
7. *Only if OTG FS is not used. Two embedded High-Speed PHYs (shared with Cortex-A7 USB host ports).*
8. *Only SDMMC3 instance with 4-bit data available for Cortex-M4 (three instances in total: 8 + 8 + 4 bits).*
9. *Usable by Cortex-A7 only.*
10. *In addition to Dual-Cortex-A7 up to 650 MHz (single Cortex-A7 on STM32MP151 line).*

4.1 STM32 product cross-compatibility

The STM32MP15x lines embed a set of peripherals which can be classified in three groups of compatibility:

Full : Full backward compatibility is ensured by the STM32CubeMP1 package, although some minor changes are possible.

Partial: Compatibility could be ensured with some modification in the user code due to major peripheral version changes.

None: The feature does not exist in the product, or the feature have to be migrated to the Cortex-A7 OpenOS side (for example display control or Ethernet connectivity).

Table 4. STM32F469/479 peripheral compatibility with STM32MP15x lines

Peripheral	Compatibility	STM32Cube MPU package Cortex-M4 - backward compatibility
ADC	Partial	Major peripheral changes
CAN	Partial	New TT-FDCAN IP. FDCAN only available on STM32MP153 and STM32MP157 lines
CRC	Full	Major version increase
CRYP	Full	CRYP only available on STM32MP15xC devices
DAC	Full	Major version increase
DCMI	Full	-
DMA	Partial	DMAMUX to be used to select requesters
DMA2D	No	Not present in STM32MP15x lines. Graphics not intended to be handled by Cortex-M4. GPU available for Cortex-A7 on STM32MP157 line
DSI	No	Display not intended to be handled by Cortex-M4. DSI only available on STM32MP157 line
ETH	No	Ethernet not intended to be handled by Cortex-M4
FMC	Partial	Only support for limited PSRAM-like parallel interface controls, such as secondary low-speed parallel RAM-based display or custom SoC (FPGA or else). Not possible for Cortex-M4 in case of Boot from NAND-Flash
HASH	Full	-
I2C	Partial	Major peripheral changes
IWDG	No	IWDG not intended to be used for Cortex-M4 in STM32MP15x lines
LTDC	No	Display not intended to be handled by Cortex-M4
OTGFS	Partial	Partial, using OTG peripheral in FS mode with embedded Full-Speed PHY
OTGHS	Partial	Partial, using OTG peripheral in HS mode with embedded High-Speed PHY port #2
QUADSPI	Partial	Major version increase. QUADSPI not intended to execute in place (XiP) for Cortex-M4, only possible as data storage if not used by Boot from Serial NOR-Flash or Serial NAND-Flash
RNG	Full	Major version increase
RTC	Partial	Major peripheral changes related to security and tamper
SAI	Full	Major version increase
SDMMC	Full	Using SDMMC3 + major version increase
SPI	Partial	Major peripheral changes
TIM1 to 8 and TIM12 to 14	Full	-
TIM9 to 11	Full	TIM9 to TIM11 could be remapped to TIM15 to TIM17
UART/USART	Full	Major version increase
WWDG	Full	Major version increase

4.2 Memory mapping

The peripheral address mapping has been changed in ST32MP15x lines compared to STM32F469/479 line.

The address mapping is irrelevant when using STM32CubeMP1 package for Cortex-M4.

The internal Cortex-M4 memory address mapping has been changed in STM32MP15x lines as there is no Flash, hence the Cortex-M4 code is always executed from internal SRAM.

Find below an extract of STM32F469/479 and STM32MP157 line memory maps.

Figure 1. STM32F469/479 memory map (focus view on internal memories)

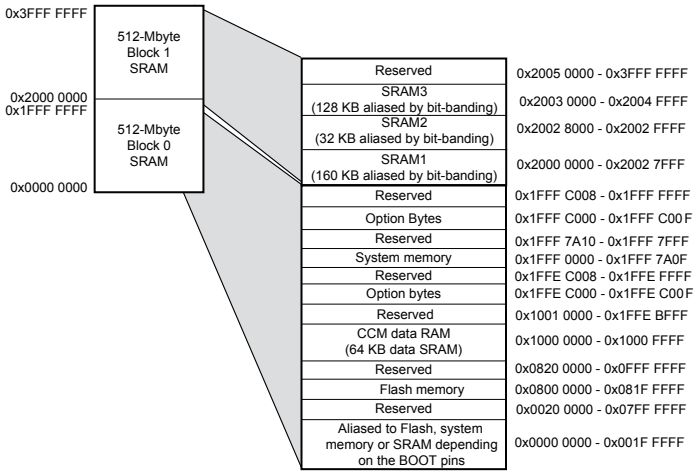
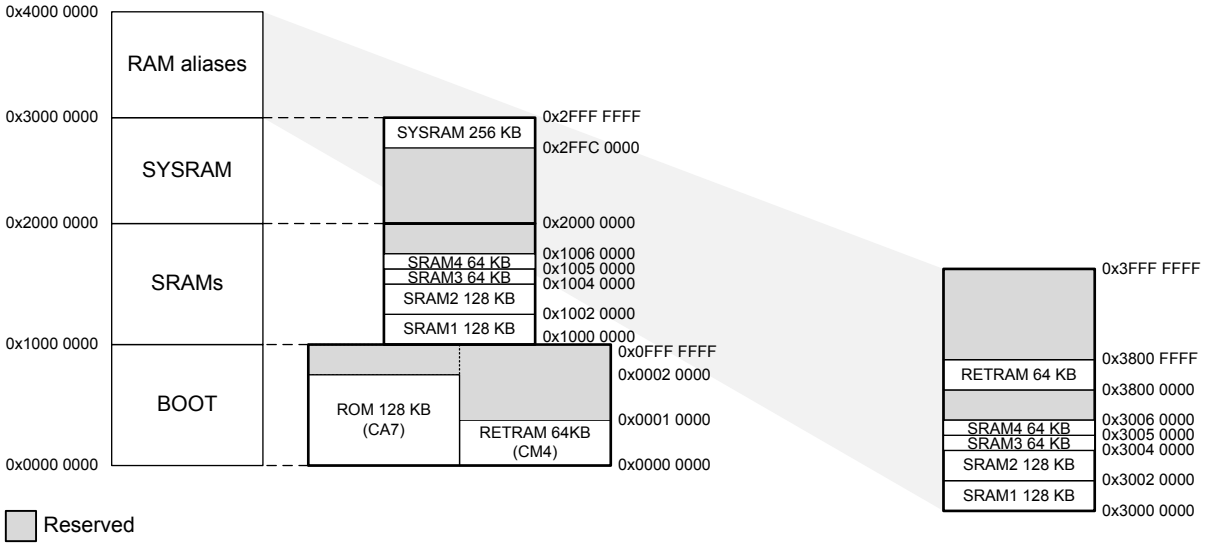


Figure 2. STM32MP157 line memory map (focus view on internal memories)



5 Application migration strategy

In addition to the hardware changes to be considered on the migration of STM32F469/479 devices towards STM32MP15x devices, some key STM32 MPU multi-core architecture concepts must be taken into consideration. Refer to the following article for more information:

- https://wiki.st.com/stm32mpu/index.php/Getting_started_with_STM32_MPU_devices

It is important to understand the hardware execution context principle for the migration, as each STM32MP15x internal peripheral is assigned to an execution context(s). A critical point to keep in mind is that some peripherals are shared, which implies that they can be used simultaneously by several contexts.

Once the hardware execution contexts are understood, the next step is to understand the assignment capabilities for the STM32MP15x devices. More information can be found on the article:

- https://wiki.st.com/stm32mpu/index.php/STM32MP15_peripherals_overview

The article mentioned above provides an overview of each peripheral's description as well as more information on the software components that allow each of them to be controlled in several execution contexts.

In order to migrate an existing application from STM32F469/479 line to STM32MP15x lines, three main aspects must be considered on the MPU platform:

- All the user interface management is controlled by Linux, running in the Arm® Cortex®-A7 non-secure hardware execution context. Migrating a user interface from a STM32Cube based firmware to a Linux based solution enables powerful graphical effects thanks to the presence of a GPU (graphics processing unit). The GPU creates the frames to be displayed via the LTDC and DSI controllers. On top of that, Linux also takes care of the high speed connectivity such as the USB, Ethernet or Flash interfaces. The Wiki article http://wiki.st.com/stm32mpu/index.php/Linux_application_frameworks_overview introduces the frameworks that are delivered as part of the STM32 MPU embedded software to support all the peripherals.
- All the real time and/or low power tasks might be run in the Arm Cortex-M4 execution context via an STM32Cube based firmware, and the corresponding peripherals should be assigned to this context. To better understand how to adapt the STM32F469/479 device firmware to run under this MPU coprocessor context refer to the following article:

  https://wiki.st.com/stm32mpu/index.php/STM32CubeMP1_development_guidelines .
- The communication between those two contexts is made via the inter process communication controller (IPCC), controlled by RPMsg on Linux side and OpenAMP on STM32Cube side. The peripheral and the respective frameworks are reachable from the following article:

  https://wiki.st.com/stm32mpu/index.php/IPCC_internal_peripheral .

Revision history

Table 5. **Document revision history**

Date	Version	Changes
19-Feb-2019	1	Initial release.

Contents

List of tables

List of figures