Migrating from STM32MP13x to STM32MP21x MPUs

Introduction

The STM32MP131, STM32MP133, and STM32MP135 devices are part of the STM32 Arm® Cortex® MPUs. They all feature a single Cortex®-A7 core.

These devices are referred to as STM32MP13x in this document.

The STM32MP211, STM32MP213, and STM32MP215 devices are part of the STM32 Arm® Cortex® MPUs. They all feature a Cortex®-A35 core, and a Cortex®-M33 core.

These devices are referred to as STM32MP21x in this document.

Similarly to the STM32MP13x lines with a Cortex®-A7 core, the high performance Cortex®-A35 core inside the STM32MP21x runs open operating systems such as Linux®. Both provide rich connectivity and the support of a software community. A system configuration migration between different hardware using different device trees is supported with OpenSTLinux.

This application note provides information to facilitate the migration from an STM32MP13x design towards an STM32MP21x one.

Table 1. Applicable products

Type	Product lines
Microcontrollers	STM32MP131, STM32MP133, STM32MP135 STM32MP211, STM32MP213, STM32MP215

1 General information

This document applies to STM32MP13x and STM32MP21x Arm®-based MPUs.

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

2 References

Table 2. **Referenced documents**

N°	Reference	Title
[1]	DS13483, DS13874, DS13875, DS13876, DS13877, DS13878	STM32MP13xA/D/C/Fxx datasheet
[2]	DS14556, DS14557	STM32MP21xA/D/C/Fxx datasheet
[3]	AN5474	Getting started with STM32MP13x lines hardware development
[4]	AN5489	Getting started with STM32MP2 series MPUs hardware development
[5]	RM0475	STM32MP13x reference manual
[6]	RM0506	STM32MP21x reference manual

3 STM32MP13x line overview

The STM32MP13x lines are part of the STM32MP1 series. The system includes a single Cortex®-A7with L2 cache.

For more information about the full-featured system of the STM32MP13x, see Table 3.

Table 3. Configuration of the STM32MP13x lines

Lines	Reference manual	Cortex®-A7 configuration	Other
STM32MP131	[5]	Single-core	1 ETH 1 ADC
STM32MP133		Single-core	2 ETH 2 CAN FDs 2 ADCs
STM32MP135		Single-core	2 ETH 2 CAN FDs 2 ADCs Display (LTDC) Camera module (DCMIPP)

4 STM32MP21x line overview

The STM32MP21x lines are part of the STM32MP2 series.
The full-featured system is partitioned in:

- An MPU subsystem based on a single Cortex®-A35 with a L2 cache
- An MCU subsystem: Cortex®-M33 with associated peripherals clocked according to the CPU activity

Table 4. Configuration of the STM32MP21x lines

Lines	Reference manual	Cortex®-A35 configuration	Cortex®-M33	Other
STM32MP211	[6]	Single-core	Yes	2 ETHs 2 CAN FDs Display (LTDC) Camera module (CSI-2 + DCMIPP)
STM32MP213		Single-core	Yes	2 ETHs 2 CAN FDs
STM32MP215		Single-core	Yes	1 ETH

The STM32MP21x lines offer a different set of peripherals compared to the STM32MP13x devices with some improved performance and features. See the list below.

Processing

- Single Cortex®-A35 core (up to 1.5 GHz) + Cortex®-M33 (at 400 MHz), compared to the single Cortex®-A7 core (up to 1 GHz) for the STM32MP13x
- External LPDDR4/DDR4/DDR3L with 16-bit interface at 800 MHz, compared to LPDDR2/LPDDR3 or DDR3/DDR3L with 16-bit interface at 533 MHz for the STM32MP13x

Connectivity

- Ethernet with TSN support
- One OCTOSPI, compared to one QUADSPI for the STM32MP13x

Memory

- Increased memory
  - 256-Kbyte CPU system, compared to 128 Kbyte for the STM32MP13x
  - 64-Kbyte MCU system added to 128-Kbyte MCU retention, compared to 32-Kbyte AHB SRAM for the STM32MP13x

Increased number of peripherals

- Four SAIs up to eight audio channels, compared to two SAIs up to four audio channels
- Eight MDF audio channels/filters, compared to four DFSDM input channels with two filters
- Four 32-bit general purpose timers, compared to two
- Six SPIs that include three I2Ss, compared to five SPIs that include four I2Ss
- Three I3Cs + three I2Cs, compared to five I2Cs
- A camera module CSI-2 interface and PSSI parallel interface shared with DCMPI/DCMIPP
- A lite-ISP with enhanced statistics in DCMIPP parallel camera interface

Security

- DDR on-the-fly encrypt/decrypt AES-256 with SCA protection, compared to DDR on-the-fly encrypt/decrypt AES-128
- Two SHA-512, compared to one
- Stronger SESIP 3 conformance (RSA/ECC private key store)
- Stronger secure boot encryption (AES-256, ECDSA P-384)
- Post-quantum LMS/XMSS signature support

Other

- FDCAN: CDC update, STNOTECH insertion
- DEBUG: mailbox for JTAG reopening, authentication under reset, DAP-AXI 33-bit addressing
- All three autonomous SPIs in LP/LPLV-Stopx, compared to only one
- Reduced number of GPIOs: 123/98, compared to 135

STM32MP21x 8x8 VFBGA225 package specificities

The 8x8 VFBGA225 package with 98 GPIOs on STM32MP21x has further limitations.

- Reduced number of peripherals:
  - No LPUART1
  - No TIM16, TIM17, LPTIM3, LPTIM4, LPTIM5
  - No CSI interface
- No USB2_HS signals (single lane USB)
- SDMMC2 is only 4 bits and is not a boot source
- SDMMC3 has no external level shifter control
- Only 8-bit FMC
- Only eight tampers (1 output, 7 inputs)
- No SAI4_D1 signal
- No WKUP4 signal
- No RTC_OUT2/RTC_LSCO signals
- FMC_NWE default boot pin changed (from PE14 to PD13)

5 Hardware migration

There is no compatibility between the STM32MP13x packages and the STM32MP21x packages. Alternate function multiplexing is different as well as ballout definition.

A good candidate for migration towards an STM32MP21x device can be chosen by considering the following criteria:

- Equivalent number of available GPIOs. A precise count must be done for each application use case.
- The package size
- The PCB technology cost (xFBGA 0.5 mm pitch or xFBGA 0.8 mm pitch)

The following table summarizes the key package parameters for STM32MP13x and STM32MP21x lines.

Table 5. Packages on STM32MP13x and STM32MP21x lines

Lines		STM32MP13x lines			STM32MP21x lines			
Product		STM32MP13xxAG	STM32MP13xxAF	STM32MP13xxAE	STM32MP21xxAS	STM32MP21xxAL	STM32MP21xxAR	STM32MP21xxAQ
Package		LFBGA289	TFBGA320	TFBGA289	VFBGA225	VFBGA361	VFBGA273	TFBGA264
	Size (mm)	9x9	11x11	14x14	8x8	10x10	11x11	14x14
	Thickness (max) (mm)	1.7	1.2	1.2	1.0	1.0	1.0	1.2
	Ball pitch (mm)	0.5	0.5	0.8	0.5	0.5	0.5	0.8
GPIOs		135	135	135	98	123	123	123

5.1 Power supply aspects

To ensure a better performance and power consumption, the STM32MP13x and STM32MP21x lines provide a separate supply voltage domain for Cortex®-A7, Cortex®-35, and the core domain.

When the activity requires it, these supplies could be overdriven to increase performance, lowered to reduce leakage, or even shut down.

Similarly to the STM32MP13x, for which it is recommended to use the STPMIC1 power supply management IC, it is recommended to use the STPMIC2L for the STM32MP21x.

More information about STPMICx can be found on www.st.com.

Table 6. Power supply comparison between STM32MP13x and STM32MP21x lines

STM32MP13x		STM32MP21x		Comments
Supply name	**Typical**	**Supply name**	**Typical**	
V_{DD}	1.8/3.3 V	V_{DD}	1.8/3.3 V	V_{DD} supply level on STM32MP21x must be carefully controlled within the following ranges: 1.71 V - 1.98 V (typ 1.8 V) 3 V - 3.6 V (typ 3.3 V) On STM32MP13x, the V_{DD} supply level can be set on the whole 1.71 V - 3.6 V range
V_{DDSD1}		V_{DDIO1}		Optional domain for SD card on SDMMC1
V_{DDSD2}		V_{DDIO2}		STM32MP13x: optional domain for SD card on SDMMC2

STM32MP13x		STM32MP21x		Comments
Supply name	Typical	Supply name	Typical	
	1.8/3.3 V		1.8/3.3 V	STM32MP21x: optional domain for eMMC on SDMMC2
N/A		V_{DDIO3}		STM32MP21x: optional domain for OCTOSPI port1
V_{DDCORE}	1.25 V (1)	V_{DDCORE}	0.82 V (1)	STM32MP21x: also V_{DDDCSI}.
V_{DDCPU}	1.25 V (1) 1.35 V (2)	V_{DDCPU}	0.8 V (1) 0.91 V (2)	STM32MP13x: for Cortex®-A7 STM32MP21x: for Cortex®-A35
N/A	0.8 V (3)	V_{08CAP}	0.8 V (3)	Backup regulator decoupling capacitance

1. *Support lowering.*
2. *In overdrive mode.*
3. *External LDO.*

Table 7. System power modes on STM32MP13x and STM32MP21x lines

Mode (STM32MP13x)	Mode (STM32MP21x)	V_{DD}		V_{DDCORE}		V_{DDCPU}	
		State	Control	State	Control	State	Control
Run	Run1	ON	-	ON	-	Overdrive	Hardware/software (1)
						ON	-
N/A(2)	Run2					OFF	PWR_CPU_ON
Stop	Stop1	ON	-	ON	-	Overdrive	Hardware/software (1)
						ON	-
N/A(3)	Stop2					OFF	PWR_CPU_ON
LP-Stop	LP-Stop1	ON	-	ON	-	Overdrive	Hardware/software (1)
						ON	-
N/A(3)	LP-Stop2					OFF	PWR_CPU_ON
LPLV-Stop	LPLV-Stop1	ON	-	Lowered	PWR_ON or PWR_LP	Lowered	PWR_CPU_ON or PWR_LP
LPLV-Stop2	LPLV-Stop2					OFF	PWR_CPU_ON
Standby	Standby1	ON	-	OFF	PWR_ON	OFF	PWR_CPU_ON
VBAT	VBAT1	OFF using hardware or software					

1. *For example, GPIO or I²C command to STPMIC.*
2. *On STM32MP13x, it is not possible to be in Run mode while the V_{DDCPU} voltage is OFF. On STM32MP21x, the Cortex®-M33 can be running when the V_{DDCPU} voltage is OFF.*
3. *On STM32MP13x, it is not possible to be in Stop mode while the V_{DDCPU} voltage is OFF.*

5.2 Alternate function multiplexing aspects

The STM32MP21x pinout and alternate function multiplexing (AFMux) are derived from STM32MP25x, which is not compatible with STM32MP13x.

It is not possible to directly port an application running on an STM32MP13x board to a STM32MP21x board. The board design must be reworked to fit STM32MP21x ballout and alternate function multiplexing.

6 Boot mode selection

The STM32MP13x and STM32MP21x devices always start from an internal boot ROM. The internal boot ROM starts based on the boot pins and on the internal OTP fuses.

- Boot from external flash memory with default OTPs
  - SD card (SDMMC1)
  - e•MMC (SDMMC2)
  - SLC-NAND (FMC)
  - Serial NOR or serial NAND flash memory
- Boot from UART or USB device
  - Used to access the device from the STM32CubeProgrammer (STM32CubeProg) to program the external flash memory or internal OTP fuses.

Default pins for the boot interface are different for the STM32MP13x and STM32MP21x lines.

Table 8. Bootloader interface on STM32MP13x and STM32MP21x

Interface	STM32MP13x	STM32MP21x
OCTOSPI1	N/A	OCTOSPI_P1_CLK PD0 OCTOSPI_P1_NCLK PD1 OCTOSPI_P1_DQS PD2 OCTOSPI_P1_NCS1 PD3 OCTOSPI_P1_IO0 PD4 OCTOSPI_P1_IO1 PD5 OCTOSPI_P1_IO2 PD6 OCTOSPI_P1_IO3 PD7 OCTOSPI_P1_IO4 PD8 OCTOSPI_P1_IO5 PD9 OCTOSPI_P1_IO6 PD10 OCTOSPI_P1_IO7 PD11
QUADSPI	QUADSPI_BK1_IO0 PF8 QUADSPI_BK1_IO1 PF9 QUADSPI_BK1_NCS PB2 QUADSPI_BK2_IO0 PH2 QUADSPI_BK2_IO1 PG10 QUADSPI CLK PF10	-
SDMMC1	SDMMC1_D0 PC8 SDMMC1_CK PC12 SDMMC1_CMD PD2	SDMMC1_D0 PE4 SDMMC1_CK PE3 SDMMC1_CMD PE2
SDMMC2	SDMMC2_D0 P14 SDMMC2_CK PE3 SDMMC2_CMD PG6	SDMMC2_D0 PE13(1) SDMMC2_CK PE14 SDMMC2_CMD PE15
UART4	UART4_TX PD6 UART4_RX PD8	No UART4 boot
UART5	UART5_TX PB13 UART5_RX PB5	UART5_TX PA0 UART5_RX PB15
UART7	UART7_TX PF7 UART7_RX PF6	N/A

Interface	STM32MP13x	STM32MP21x
UART8	UART8_TX PE1 UART8_RX PE0	N/A
USART2	N/A	USART2_TX PA4 USART2_RX PA8
USART3	USART3_TX PB10 USART3_RX PB12	N/A
USART6	USART6_TX PC6 USART6_RX PC7	USART6_TX PF5 USART6_RX PF4
USB	USB_DP2 USB_DM2	OTG_HSDM OTG_HSDM
FMC	FMC_D0 PD14 FMC_D1 PD15 FMC_D2 PD0 FMC_D3 PD1 FMC_D4 PE7 FMC_D5 PE8 FMC_D6 PE9 FMC_D7 PE10 FMC_D8 PE11 FMC_D9 PE12 FMC_D10 PE13 FMC_D11 PE14 FMC_D12 PE15 FMC_D13 PB8 FMC_D14 PD9 FMC_D15 PD10 FMC_CLE PD11 FMC_NCE PG9 FMC_NWE PD5 FMC_ALE PD12 FMC_NOE PD4 FMC_NWAIT PA9	FMC_D0 PE9 FMC_D1 PE6 FMC_D2 PE7 FMC_D3 PD15 FMC_D4 PD14 FMC_D5 PB13 FMC_D6 PD12 FMC_D7 PB14 FMC_D8 PB5 FMC_D9 PB6 FMC_D10 PB7 FMC_D11 PD13 FMC_D12 PB8 FMC_D13 PB9 FMC_D14 PB11 FMC_D15 PB10 FMC_CLE PE11 FMC_NCE1 PE12 FMC_NWE PE14 FMC_ALE PE8 FMC_NOE PE15 FMC_RNB PE13 FMC_NWE on PD13 on 8x8 package: only 8-bit interface

1. *SDMMC2 is not a boot source on VFBGA225 8x8 package*

Refer to STM32 wiki articles below:

- https://wiki.st.com/stm32mpu/wiki/Boot_chain_overview
- https://wiki.st.com/stm32mpu/wiki/STM32_MPU_ROM_code_overview

7 Peripheral migration

This section presents a full view of the features and peripheral counts of STM32MP13x and STM32MP21x lines. It also presents a peripheral address mapping snap shot for the concerned products.

7.1 Peripheral feature comparison

Table 9. Peripheral comparison between STM32MP13x and STM32MP21x

Product Line		STM32MP135	STM32MP133	STM32MP131	STM32MP215	STM32MP213	STM32MP211
CPU	Core	Cortex®-A7 FPU neon TrustZone®, single core			Cortex®-A35 FPU neon TrustZone®, single core		
	Cache size	32-Kbyte data cache			32-Kbyte data cache		
		32-Kbyte instruction cache			32-Kbyte instruction cache		
		128-Kbyte level 2 cache			128-Kbyte level 2 cache		
	Frequency	Up to 650 MHz			Up to 1200 MHz		
	Overdrive mode	Up to 1000 MHz			Up to 1500 MHz		
MCU	Core	-			Cortex®-M33 FPU TrustZone®		
	Cache size				16-Kbyte data cache		
					16-Kbyte instruction cache		
	Frequency				Up to 400 MHz		
Embedded SRAM	CPU system	128 Kbytes			256 Kbytes		
	Subsystem	32 Kbytes (AHB)			64 Kbytes (MCU) Tamper protected		
	MCU retention	-			128 Kbytes		
	Backup	8 Kbytes (Tamper protected)			8 Kbytes (Tamper protected)		
SDRAM addressing space	16-bit LPDDR	Up to 1 Gbyte (LPDDR2/3)			Up to 2 Gbytes (LPDDR4)		
	16-bit DDR	Up to 1 Gbyte (DDR3/DDR3L)			Up to 2 Gbytes (DDR3L) Up to 4 Gbytes (DDR4)		
SDRAM frequency	LPDDR frequency	LPDDR2/3 up to 533 MHz			LPDDR4 up to 800 MHz		
	DDR frequency	DDR3/DDR3L up to 533 MHz.			DDR3L/DDR4 up to 800 MHz.		
	On-the-Fly Encrypt/Decrypt	Yes, AES-128 with SCA protection encode/decode			Yes + AES-256 with SCA protection encode/decode		
Backup registers		128 bytes (32 x 32 bits, tamper protected)			512 bytes (128 x 32 bits, tamper protected)		
Timers	16-bit general-purpose	8 x 16 bits (6 securables) 2 x 32 bits			8 x 16 bits + 4 x 32 bits		
	Advanced- control	2			2		
	Basic	2			2		
	Low power	5 (2 securables)			5		

Product Line		STM32MP135	STM32MP133	STM32MP131	STM32MP215	STM32MP213	STM32MP211
Watchdog		2 (Independent, independent secure)			5 (4 x independent, 1x window)		
Communication interfaces	SPI/I^2S	5 (2 securable)/4(full duplex)			6/3 (full duplex)		
	I^2C	5 (3 securables)			3		
	I3C	-			3		
	USART + (LP)UART	4 (2 securables) + 4			4 + 4		
	SAI	2 (up to 4 audio channels)			4		
	USB host	USB 2.0 host (USBH), 2 ports, embedded hi-speed PHY			USB 2.0 host (USBH), 2 ports, embedded hi-speed PHY		
	USB OTG	USB 2.0 host/device (OTG), 1 port, embedded hi-speed PHY (shared with USBH port 2)			USB 2.0 host/device (OTG), 1 port, embedded hi-speed PHY (shared with USBH port 2)		
	SPDIFRX	4 inputs			4 inputs		
	FDCAN	2 (1 x TT-FDCAN), 10 Kbytes shared buffer		-	2 (1 x TT-FDCAN), 10 Kbytes shared buffer		-
SDMMC (SD, SDIO, e•MMC)		2 (8 + 8) (securable)			3 (8 + 8 + 4 bits)		
Serial-Flash	-	1 x QUADSPI (securable)			1 x OCTOSPI		
	On-the-Fly decryption	-			Yes		
FMC memory controller	PSRAM	4 × CS, up to 4 × 64 Mbytes			4 × CS, up to 4 × 64 Mbytes		
	NAND	2 × CS, SLC, BCH4/8			1 × CS, SLC, BCH4/8		
Ethernet		2 ports, R(G)MII, MII		1 port, R(G)MII, MII	2 ports, R(G)MII, MII		1 port, R(G)MII, MII
LCD-TFT parallel interface		Up to 24-bit 90 MHz pixel clock (up to 1080p30) 2 layers (including 1 secure)			Up to 24-bit 150 MHz pixel clock (up to 1080p60) No rotation, 2 layers (including 1 secure)	-	-
DMA		3 instances (1 secure), 33 physical channels in total			3 instances (1 secure), (TBD) physical channels in total		
Cryptography		DES/TDES, AES-256, PKA			DES/TDES, AES-256, PKA		
Hash		SHA-1, SHA-2 and SHA-3 (up to 512), HMAC			SHA-1, SHA-2 and SHA-3 (up to 512), MD5, HMAC		
True random number generator		Yes			Yes		
Fuses (one-time programmable)		3072 effective bits			12288 effective bits		
Camera interface	parallel	Up to 16 bits and up to 120 MHz.	-		Up to 16 bits and up to 120 MHz Path shared with CSI	-	

Product Line		STM32MP135	STM32MP133	STM32MP131	STM32MP215	STM32MP213	STM32MP211
Camera interface	CSI-2	-			2 x data lanes 2.5 Gbps each. Path shared with DCMIPP	-	
	ISP	-			Yes, basic ISP		
Digital filter		4 input channels with 2 filters (DFSDM)			4 input channels with 4 filters (MDF)		
ADC		2 x 12-bit ADCs			2 x 12-bit ADCs		
Number of ADC channels		19/18s			24		
Operating voltage		1.71 to 3.6 V			1.71 to 1.98 V or 3 to 3.6 V		
Junction temperature		-40 to 125 °C			-40 to 125 °C		

7.2 Memory mapping

The peripheral address mapping is different in STM32MP21x lines compared to STM32MP13x lines.
Find below STM32MP13x and STM32MP21x line memory maps.
More details can be found in the respective device reference manual.

Figure 1. STM32MP13x memory map

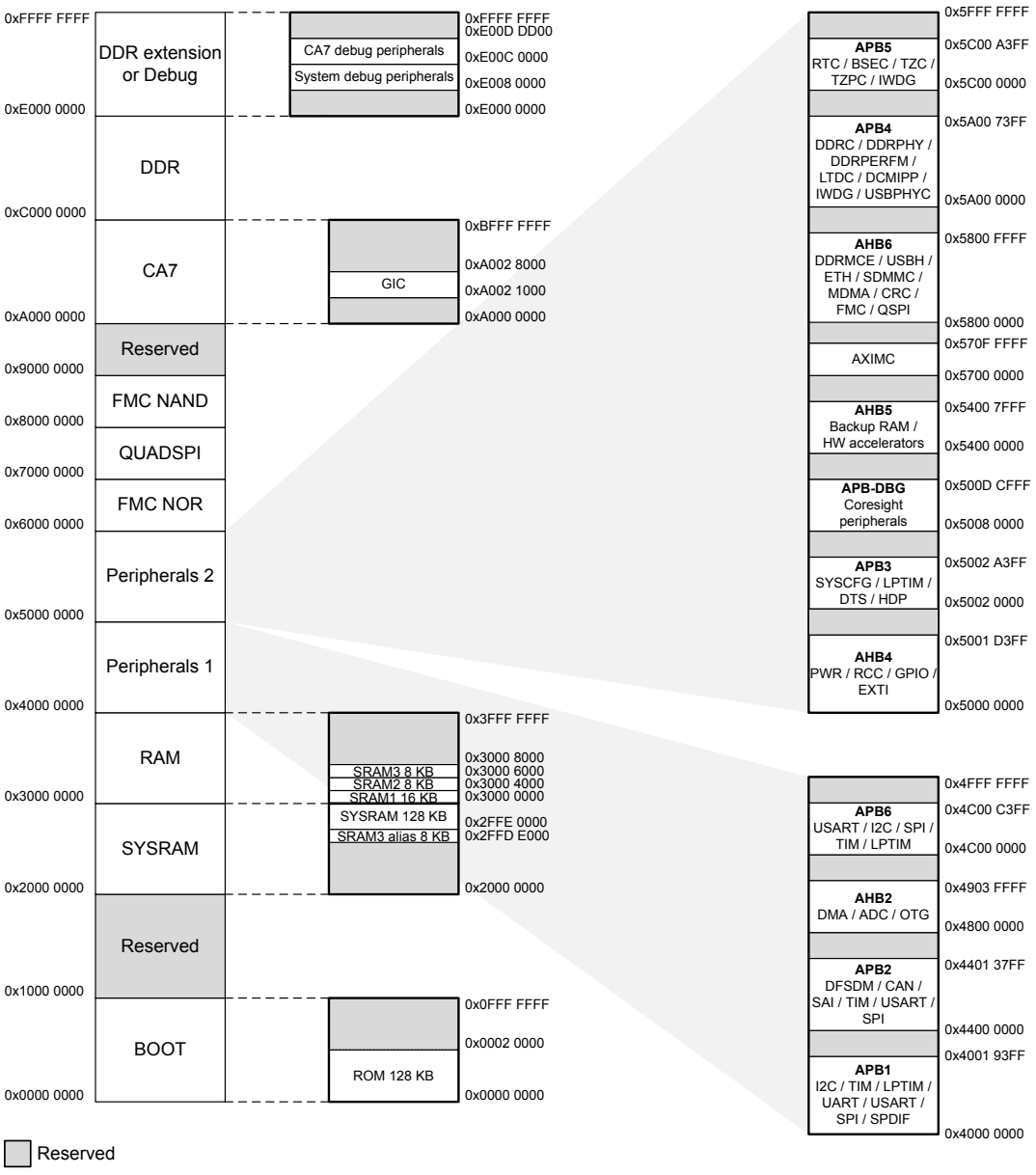
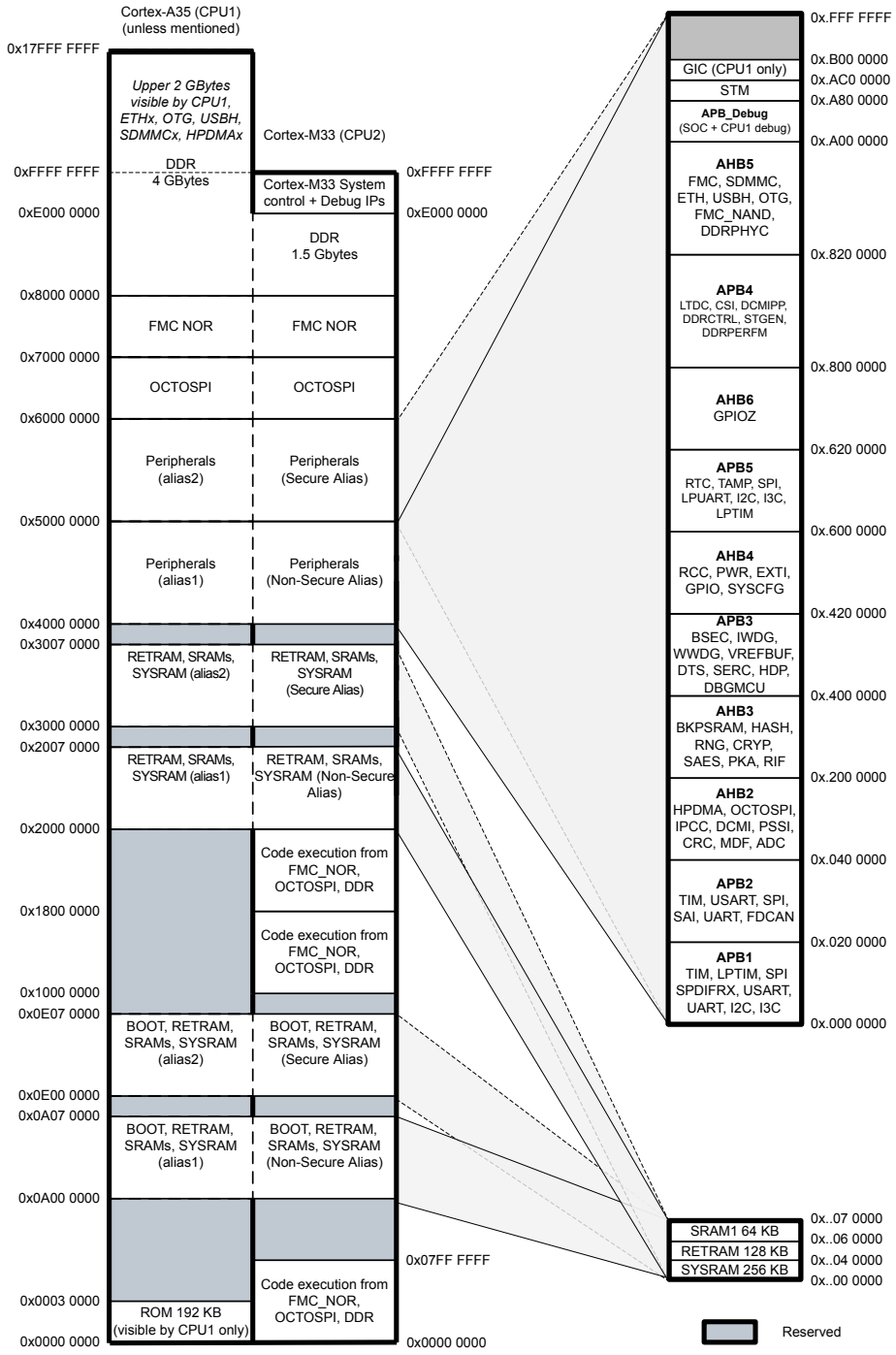


Figure 2. **STM32MP21x memory map**



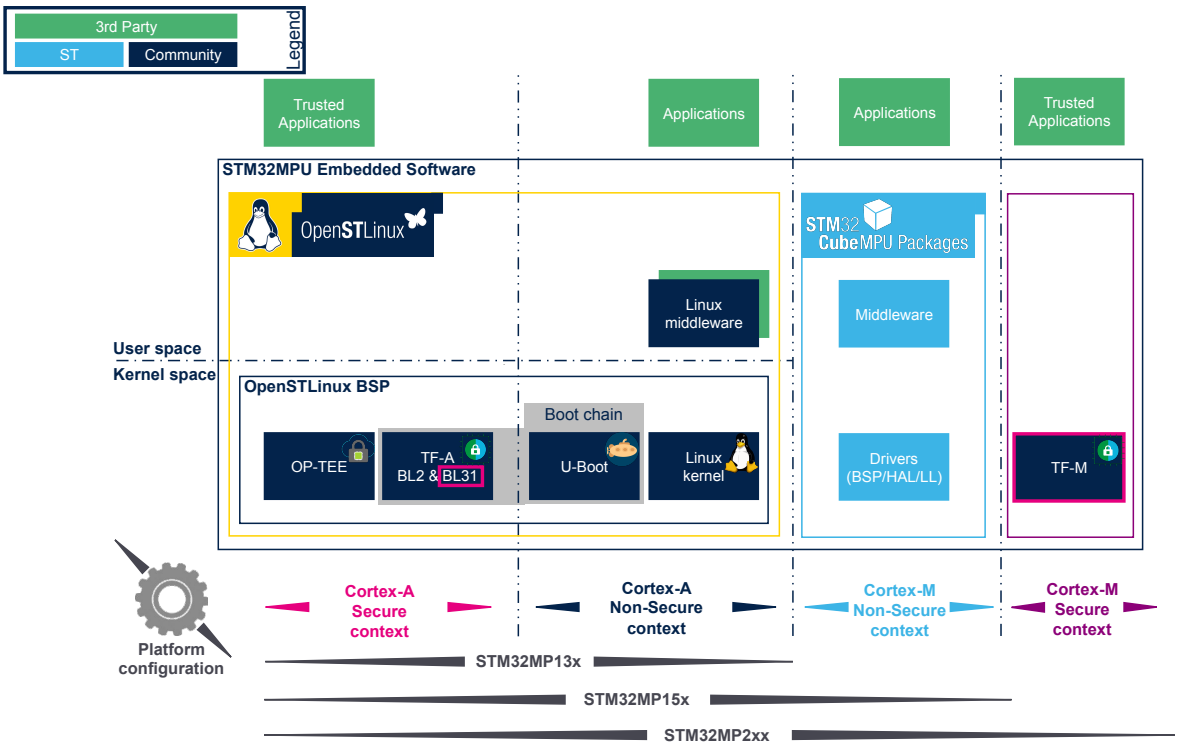
8 Application migration strategy

8.1 MPU embedded software: STM32MP13x versus STM32MP21x

STM32MP13x and STM32MP21x platforms are supported by the same embedded software delivery, which adapts automatically the compilation chain and the selected software components according to the machine selection.

Figure 3 shows all the software components part of STM32MPU embedded software delivery and their applicability per STM32 MPU lines (STM32MP15x, STM32MP13x and STM32MP2xx). New dedicated STM32MP2 series software components are highlighted in red.

Figure 3. STM32MPU embedded software

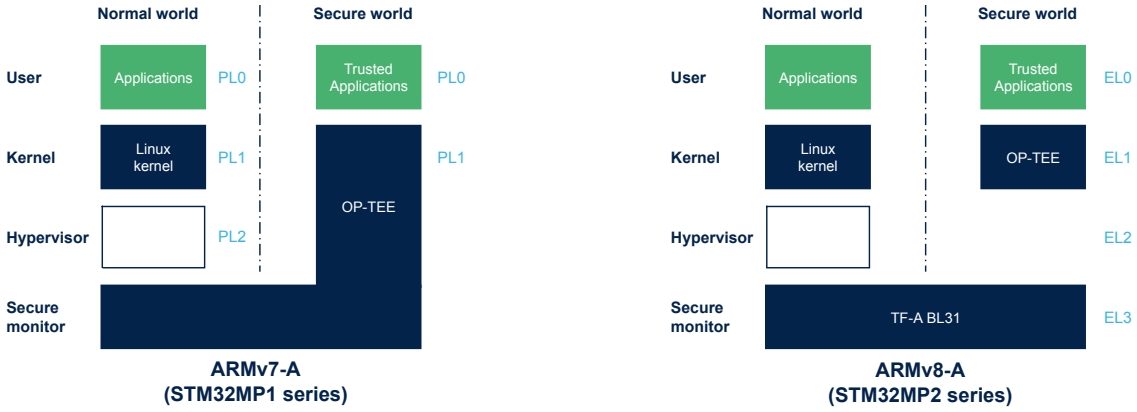


One of the main differences between the STM32MP1 series and the STM32MP2 series is the Cortex®-A Arm® architecture:

- STM32MP1 embeds a Cortex®-A7 cluster based on the Armv7-A 32-bit architecture.
- STM32MP2 embeds a Cortex®-A35 cluster based on the Armv8-A 64-bit or 32-bit architecture.

Armv8-A architecture introduces the notion of execution levels (EL), which is different from Armv7-A privilege levels (PL).

Figure 4. Armv7-A vs Armv8-A execution level and associated software



On Armv7-A secure monitor, functionalities are managed at secure PL1 level. This is also used for secure OS functionality.

On Armv8-A, the new architecture introduces the EL3 execution level that is dedicated to the secure monitor execution. The latter is in charge of switching between secure and nonsecure contexts and cluster management (low power modes, CPU plug/unplug, and so on). The secure monitor and secure operating system are executed in separate environments.

OpenSTLinux supports both Armv7-A and Armv8-A architectures.

- In the STM32MP1 series, the secure monitor functionality is facilitated by the OP-TEE secure operating system, which is a single software component that offers two distinct sets of features.
  - Secure monitor
  - Secure operating system
- On the STM32MP2 series, the trusted firmware Cortex®-A (TF-A) BL31 provides a secure monitor and OP-TEE provides a secure OS.

Another main difference between STM32MP13x devices and STM32MP21x devices is the addition of a Cortex®-M33 on STM32MP21x devices.

- STM32MP13x devices do not embed coprocessor.
- STM32MP21x devices embed a Cortex®-M33 processor based on the Armv8-M architecture. It owns secure (TrustZone®) and nonsecure execution contexts.
  - Cortex®-M33 nonsecure context can run STM32MPU cube firmware
  - Cortex®-M33 secure context (optional) can run trusted firmware Cortex®-M (TF-M) secure OS

8.2 Linux application migration

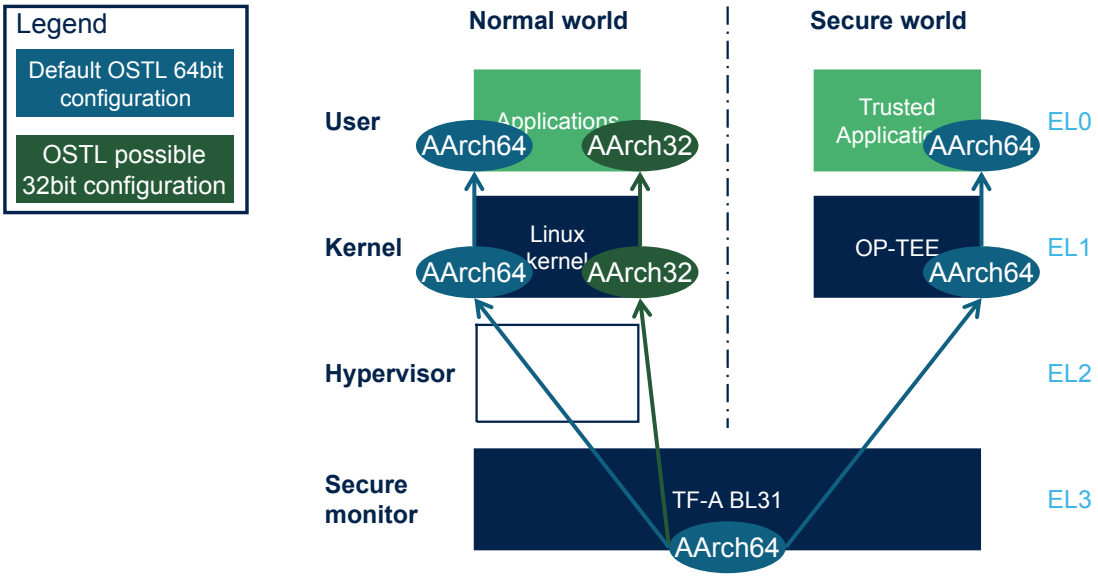
On the STM32MP1 series, Linux applications are compiled in ARMv7-A 32-bit, compliant with the Cortex®-A7 processor. The STM32MP2 series is equipped with a Cortex®-A35 processor, which natively supports the ARMv8-A AArch64 instruction set, designed for 64-bit architecture. To ensure compatibility and performance, all OSTL (OpenSTLinux) software components, including Linux applications, are compiled using the AArch64 instruction set. This approach aligns with the native capabilities of the Cortex®-A35 processor. Additionally, this enables support up to 4 Gbytes of DDR memory, fully utilizing the processor addressing range.

The Cortex®-A35 processor provides the flexibility to choose the instruction set at runtime. This choice can be made between the native AArch64 (64-bit) or AArch32 (32-bit) instruction sets.

Each execution level can independently select its preferred instruction set according to the following rules:

- EL3 is always Aarch64.
- Transition from Aarch64 ELx to Aarch32 ELy with x > y is possible.
- Transition from Aarch32 ELx to Aarch64 ELy with x > y is impossible.

Figure 5. Aarch32 and Aarch64 bit split detail



STMicroelectronics advises that the Linux kernel and its corresponding user-space applications should utilize the same instruction set. This recommendation is based on the fact that, although it is technically possible to mix 32-bit and 64-bit user-space applications, complexities are involved. This capability stems from the "CONFIG_COMPAT" option in the Linux kernel, which allows 32-bit applications to operate on a 64-bit kernel. Specifically, it complicates the build system by necessitating the generation of both 32-bit and 64-bit versions of libraries, toolchains, debug tools, and other components. Moreover, this duality can lead to an increase in the size of the root file system (rootfs).

STMicroelectronics proposes two options for porting an existing application running on the STM32MP1 series to the STM32MP2 series.

8.2.1 Port applications in 64-bit architecture

Use the OSTL delivery for straightforward compilation of applications in a 64-bit architecture. Two different methods are listed below:

- By relying on the OSTL build system: STM32MP2 machine automatically selects the Aarch64 compiler and associated compilation options.
- By using the OSTL STM32MP21 SDK: it provides the Aarch64 compiler and associated libraries and compilation options.

Refer to https://wiki.st.com/stm32mpu/wiki/Main_Page for more information on *How to integrate an external software package*.

In addition to the 64-bit compilation step, customers must take care of a few items to ensure the 64-bit compliance. Indeed, the size of many fundamental types have changed and even if well-written C code should not have many dependencies on the size of individual types, it is inevitable that customers come across some.

Table 10 sums up Armv8-A supported data models.

Table 10. Armv8-A supported data models

	ILP32	LP64	LLP64	ILP64
Char	8	8	8	8
Short	16	16	16	16
Int	32	32	32	64
Long	32	64	32	64
Long long	64	64	64	64
Size t	32	64	64	64
Pointer	32	64	64	64

Note:

- *ILP means* `Int`, `Long` *and* `Pointers`
- *LP means* `Long` *and* `Pointers`
- *LLP means* `Long Long` *and* `Pointers`

The 32-bit Armv7-A Linux implementation uses a data model equivalent to ILP32, while 64-bit Armv8-A Linux implementations use LP64. The differences are highlighted in Table 10.

A best practice is to enable all warnings and errors when recompiling and be notified of all the warnings and issues reported by the compiler. Particular attention must be given to type casts in the code, which are often the source of errors.

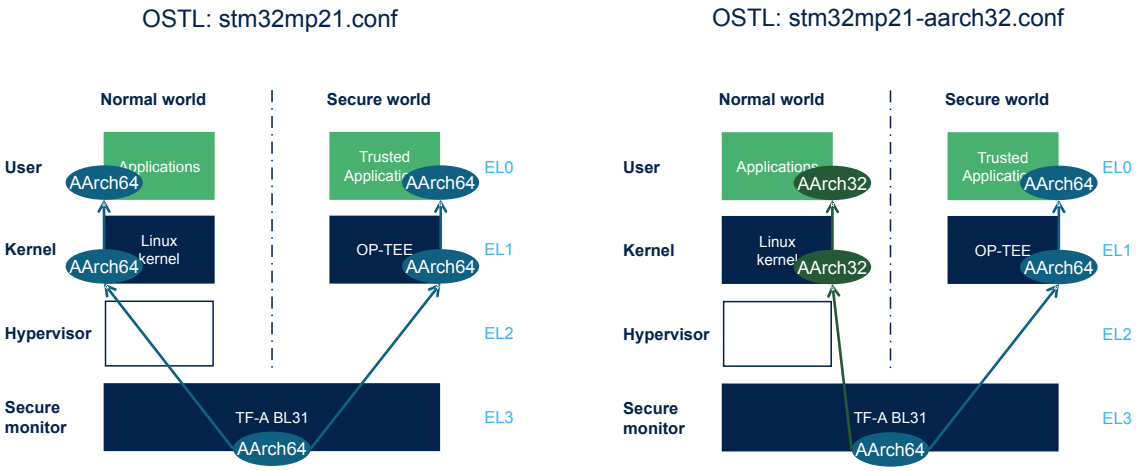
8.2.2 Configure Linux kernel and associated applications in 32-bit configuration

An alternative solution involves compiling the Linux kernel and associated applications in 32-bit mode. To facilitate this, OSTL provides a dedicated build machine named `stm32mp21-aarch32.conf`, which is designed to generate the Linux kernel and associated RootFS in Aarch32. Additionally, it compiles firmware components such as TF-A (BL2 and BL31), OP-TEE, and U-Boot in Aarch64.

Note: The `stm32mp21-aarch32` OSTL machine might not be available at the time this application note is published.

Figure 6 illustrates the differences between the default `stm32mp21` OSTL machine, which operates in Aarch64, and the `stm32mp21-aarch32` machine.

Figure 6. Aarch32 versus Aarch64 machine differences



The 32-bit machine configuration is not recommended for the following scenarios.

- Products with more than 2 Gbytes of memory: although the large physical address extension (LPAE) enables addressing more than a 4 Gbytes address space, the Cortex®-A35 in AArch32 mode is a 32-bit processor and is inherently limited to 4 Gbytes of virtual memory. To access the full range of DDR memory, the Linux kernel uses the highmem feature to map temporarily physical memory into the virtual address space. However, utilizing the highmem feature can lead to performance penalties on the overall system.
- Products with 756 Mbytes to 2 Gbytes of DDR: in configurations where the DDR memory ranges from 756 Mbytes to 2 Gbytes, the Cortex®-A35 in AArch32 mode can physically address the entire DDR. Nevertheless, its virtual memory remains capped at 4 Gbytes. In this scenario, the use of the Linux kernel highmem feature is still necessary. This may, once again, result in penalties to the overall system performance..

Note: No OSTL developer package is available for this OSTL aarch32 machine.

For further information, refer to https://wiki.st.com/stm32mpu/wiki/Main_Page.

Revision history

Table 11. **Document revision history**

Date	Revision	Changes
05-Sep-2025	1	Initial release.

Contents

List of tables

List of figures

IMPORTANT NOTICE – READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice.

In the event of any conflict between the provisions of this document and the provisions of any contractual arrangement in force between the purchasers and ST, the provisions of such contractual arrangement shall prevail.

The purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgment.

The purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of the purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

If the purchasers identify an ST product that meets their functional and performance requirements but that is not designated for the purchasers' market segment, the purchasers shall contact ST for more information.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.