AN5017
Application note

Migrating between STM32L476xx/486xx and STM32L4+ Series microcontrollers

Introduction

For designers of STM32 microcontroller applications, the ability to easily replace one microcontroller type with another from the same product family is an important asset. Migrating an application to a different microcontroller is often needed when product requirements grow, putting extra demands on memory size or increasing the number of I/Os. Cost reduction objectives may also be a reason to switch to smaller components and shrink the PCB area.

This application note analyzes the steps required to migrate an existing design between STM32L476xx/486xx and STM32L4+ Series microcontrollers. All the most important information is grouped here. Three aspects need to be considered for the migration: the hardware, peripheral(s) and firmware.

This document lists the full set of features available for STM32L476xx/486xx and STM32L4+ Series devices.

To fully benefit from this application note, the user should be familiar with the STM32 microcontroller documentation available on *www.st.com* with a particular focus on:

- STM32L4+ Series advanced ARM® based 32-bit MCUs (RM0432)
- STM32L4+ Series datasheets.

Table 1. Applicable products

Type	Part number
STM32L486xx	STM32L486JG, STM32L486QG, STM32L486RG, STM32L486VG, STM32L486ZG
STM32L476xx	STM32L476JE, STM32L476JG, STM32L476ME, STM32L476MG, STM32L476QE, STM32L476QG, STM32L476RC, STM32L476RE, STM32L476RG, STM32L476VC, STM32L476VE, STM32L476VG, STM32L476ZE, STM32L476ZG
STM32L4+ Series	STM32L4R5AG, STM32L4R5AI, STM32L4R5QG, STM32L4R5QI, STM32L4R5VG, STM32L4R5VI, STM32L4R5ZG, STM32L4R5GY, STM32L4R5ZI, STM32L4R7AI, STM32L4R7VI, STM32L4R7ZI, STM32L4R7AI, STM32L4R9AG, STM32L4R9AI, STM32L4R9VG, STM32L4R9VI, STM32L4R9ZG, STM32L4R9ZI STM32L4S5AI, STM32L4S5QI, STM32L4S5VI, STM32L4S5ZI, STM32L4S7AI, STM32L4S7VI, STM32L4S7ZI, STM32L4S9AI, STM32L4S9VI, STM32L4S9ZI

Contents

List of tables

List of figures

1 Hardware migration guide

1.1 PCB design compatibility

STM32L476xx/486xx devices do not share all the packages with STM32L4+ Series microcontrollers. *Table 2* illustrates the pinout/ballout compatibility for each common package.

Table 2. Package availability and PCB design compatibility

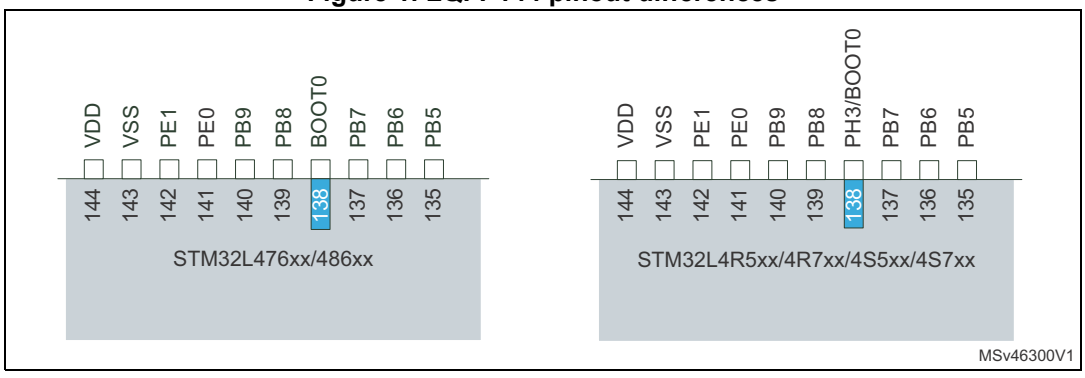
Package	STM32L4+ Series	STM32L476xx /486xx	Pinout/ballout difference	PCB design modification
LQFP144 (20x20)	x	x	Major / minor	Mandatory only for STM32L4R9xx/4S9xx
LQFP100 (14x14)	x	x	Major / minor	Mandatory only for STM32L4R9xx/4S9xx
LQFP64 (10x10)	-	x(1)	-	-
UFBGA169 (7x7)	x	-	-	-
UFBGA144	x	-	-	-
UFBGA132 (7x7)	x	x	Minor	Not mandatory(2)
WLCSP144	x	-	-	-
WLCSP72	-	x(1)	-	-

1. Only for STM32L476xx microcontrollers.
2. There is no change required from an application moving from STM32L476xx/486xx to STM32L4+ Series devices. For a migration from STM32L4+ Series to STM32L476xx/486xx devices, the PH3 GPIO is lost. Note that there is no alternate function attached to this IO for STM32L4+ Series devices.

1.1.1 LQFP144 package

Figure 1 illustrates the LQFP144 pinout differences between STM32L476xx/486xx and STM32L4R5xx/4R7xx/4S5xx/4S7xx microcontrollers.

Figure 1. LQFP144 pinout differences



1. For the highlighted (blue) terminals, the BOOT0 pin for STM32L476xx/486xx devices is replaced by a BOOT0 pin shared with a GPIO for STM32L4R5xx/4R7xx/4S5xx/4S7xx microcontrollers.

On the terminal 138 of the LQFP144 package the STM32L476xx/486xx have a BOOT0 pin while the STM32L4+ Series have a BOOT0 pin shared with a GPIO. This update is done in order to offer a maximum of flexibility if managing the BOOT0 function by option bytes for STM32L4+ Series devices.

By default, STM32L4+ Series devices are configured to use this pin as BOOT0 pin (which is the same configuration as in STM32L476xx/486xx microcontrollers), in order to keep direct compatibility with the STM32L476xx/486xx's PCB.

Figure 2 illustrates the LQFP144 pinout differences between STM32L476xx/486xx and STM32L4R9xx/4S9xx microcontrollers.

Figure 2. LQFP144 pinout differences between STM32L476xx/486xx and STM32L4R9xx/4S9xx

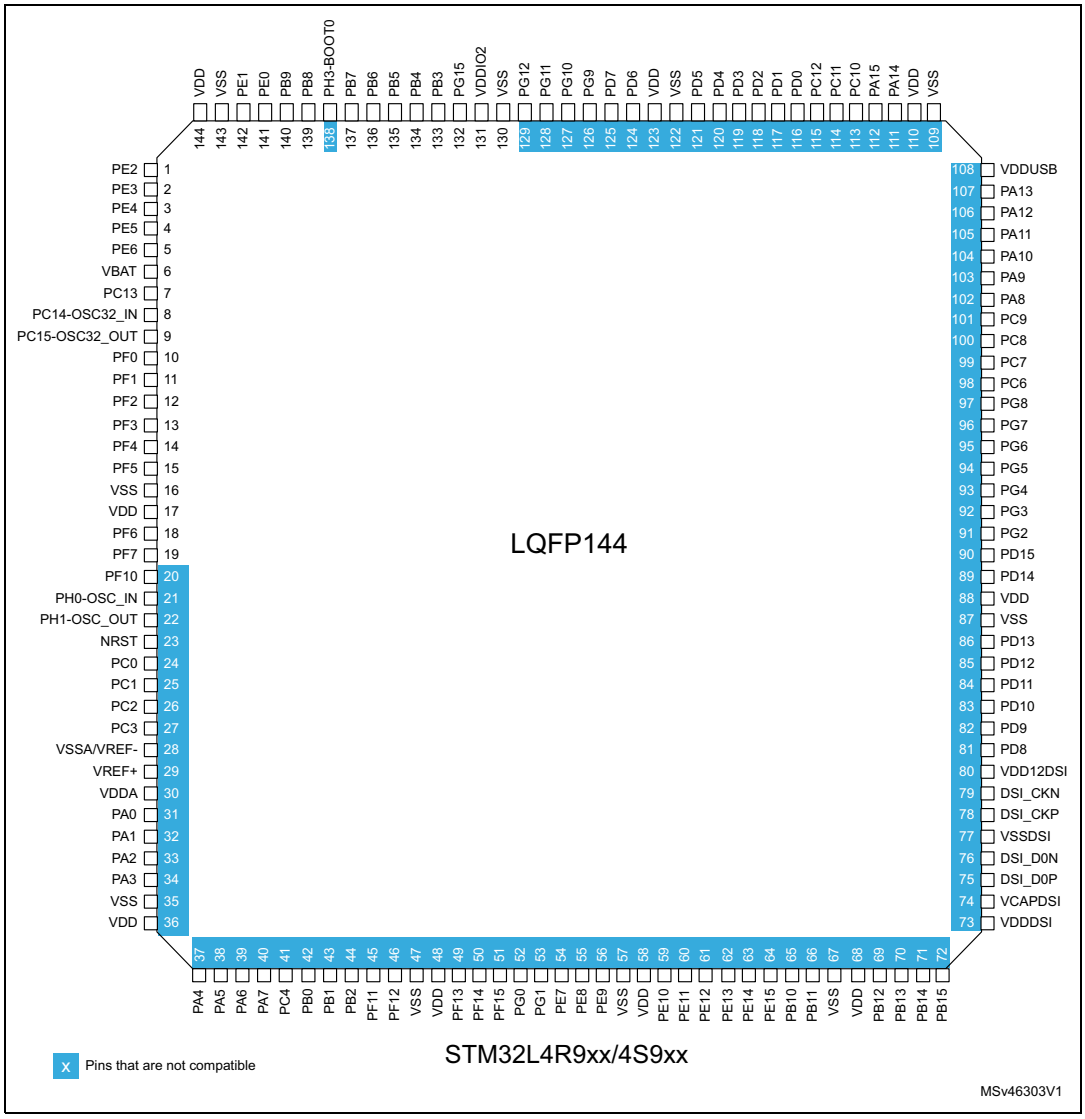


Table 3 details the LQFP144 pinout differences between STM32L476/486 and STM32L4R9xx/S9xx microcontrollers.

Table 3. STM32L476/486 and STM32L4R9xx/4S9xx pinout differences (LQFP144)

LQFP144	STM32L476/486	STM32L4R9xx/4S9xx	LQFP144	STM32L476/486	STM32L4R9xx/4S9xx
20	PF8	PF10	65	PE12	PB10
21	PF9	PH0-OSC-IN	66	PE13	PB11
22	PF10	PH0-OSC-OUT	67	PE14	VSS
23	PH0-OSC-IN	NRST	68	PE15	VDD
24	PH0	PC0	69	PB10	PB12
25	NRST	PC1	70	PB11	PB13
26	PC0	PC2	71	VSS	PB14
27	PC1	PC3	72	VDD	PB15
28	PC2	VSSA/VREF-	73	PB12	VDDDSI
29	PC3	VREF+	74	PB13	VCAPDSI
30	VSSA	VDDA	75	PB14	DSI_D0P
31	VREF-	PA0	76	PB15	DSI_D0N
32	VREF+	PA1	77	PD8	VSSDSI
33	VDDA	PA2	78	PD9	DSI_CKP
34	PA0	PA3	79	PD10	DSI_CKN
35	PA1	VSS	80	PD11	VDD12DSI
36	PA2	VDD	81	PD12	PD8
37	PA3	PA4	82	PD13	PD9
38	VSS	PA5	83	VSS	PD10
39	VDD	PA6	84	VDD	PD11
40	PA4	PA7	85	PD14	PD12
41	PA5	PC4	86	PD15	PD13
42	PA6	PB0	87	PG2	VSS
43	PA7	PB1	88	PG3	VDD
44	PC4	PB2	89	PG4	PD14
45	PC5	PF11	90	PG5	PD15
46	PB0	PF12	91	PG6	PG2
47	PB1	VSS	92	PG7	PG3
48	PB2	VDD	93	PG8	PG4
49	PF11	PF13	94	VSS	PG5
50	PF12	PF14	95	VDDIO2	PG6
51	VSS	PF15	96	PC6	PG7

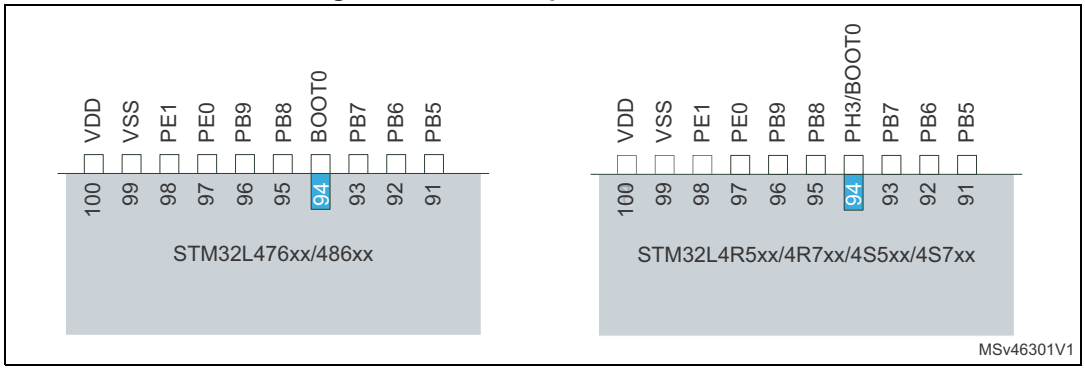
Table 3. STM32L476/486 and STM32L4R9xx/4S9xx pinout differences (LQFP144)

LQFP144	STM32L476/486	STM32L4R9xx/4S9xx	LQFP144	STM32L476/486	STM32L4R9xx/4S9xx
52	VDD	PG0	97	PC7	PG8
53	PF13	PG1	98	PC8	PC6
54	PF14	PE7	99	PC9	PC7
55	PF15	PE8	100	PA8	PC8
56	PG0	PE9	101	PA9	PC9
57	PG1	VSS	102	PA10	PA8
58	PE7	VDD	103	PA11	PA9
59	PE8	PE10	104	PA12	PA10
60	PE9	PE11	105	PA13	PA11
61	VSS	PE12	106	VDDUSB	PA12
62	VDD	PE13	107	VSS	PA13
63	PE10	PE14	108	VDD	VDDUSB
64	PE11	PE15	109	PA14	VSS
110	PA15	VDD	121	VDD	PD5
111	PC10	PA14	122	PD6	VSS
112	PC11	PA15	123	PD7	VDD
113	PC12	PC10	124	PG9	PD6
114	PD0	PC11	125	PG10	PD7
115	PD1	PC12	126	PG11	PG9
116	PD2	PD0	127	PG12	PG10
117	PD3	PD1	128	PG13	PG11
118	PD4	PD2	129	PG14	PG12
119	PD5	PD3	138	BOOT0	PH3-BOOT0
120	VSS	PD4	-	-	-

1.1.2 LQFP100 package

Figure 3 illustrates the LQFP100 pinout differences between STM32L476xx/486xx and STM32L4R5xx/4R7xx/4S5xx/4S7xx microcontrollers.

Figure 3. LQFP100 pinout differences



1. For the highlighted (blue) terminals, the BOOT0 pin for STM32L476xx/486xx devices is replaced by a BOOT0 pin shared with a GPIO for STM32L4R5xx/4R7xx/4S5xx/4S7xx microcontrollers.

On the terminal 94 of the LQFP100 package STM32L476xx/486xx have a BOOT0 pin while STM32L4+ Series have a BOOT0 pin shared with a GPIO. This update is done in order to offer a maximum of flexibility if managing the BOOT0 function by option bytes for STM32L4+ Series devices.

By default, STM32L4+ Series microcontrollers are configured to use this pin as BOOT0 pin (which is the same configuration as STM32L476xx/486xx microcontrollers), in order to keep direct compatibility with the STM32L476xx/486xx's PCB.

Figure 4 illustrates the LQFP100 pinout differences between STM32L476xx/486xx and STM32L4R9xx/4S9xx microcontrollers.

Figure 4. LQFP100 pinout differences between STM32L476xx/486xx and STM32L4R9xx/4S9xx

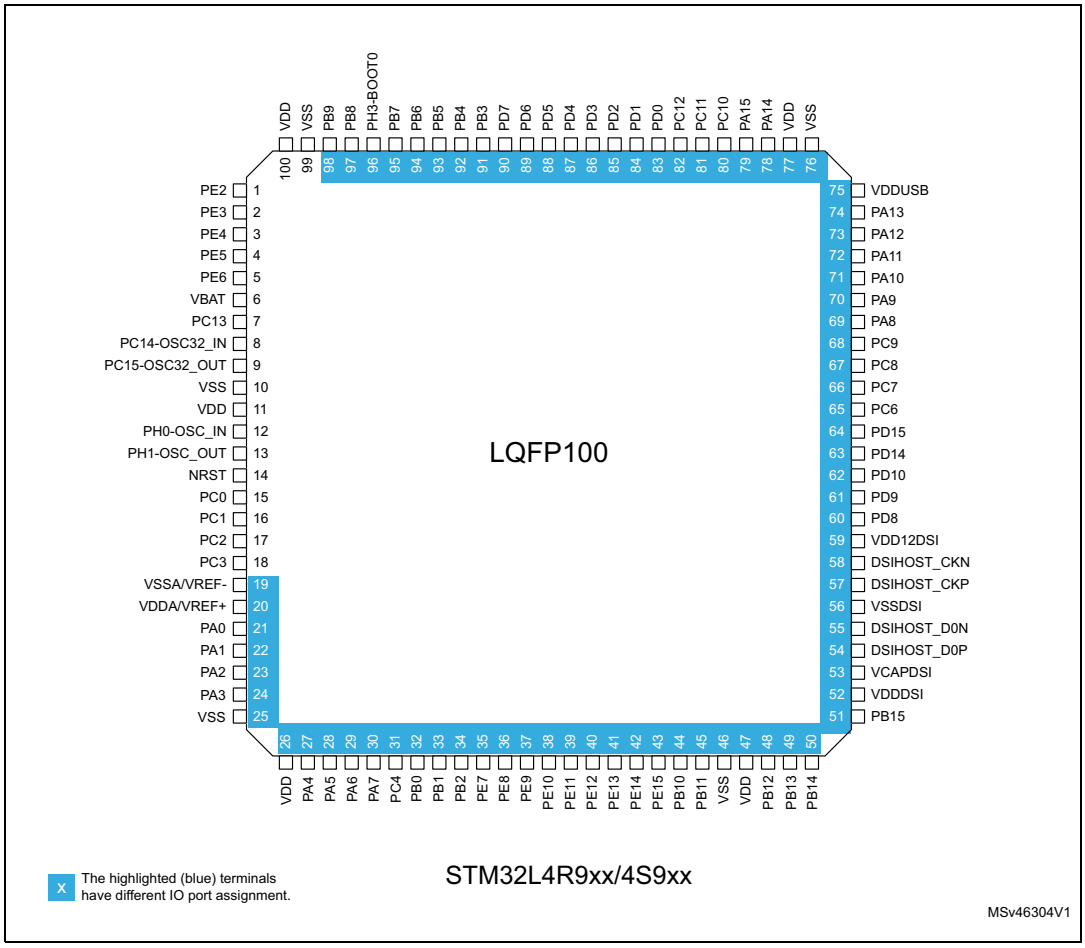


Table 4 presents the LQFP100's pinout differences between STM32L476/486 and STM32L4R9xx/4S9xx microcontrollers.

Table 4. STM32L476/486 and STM32L4R9xx/4S9xx pinout differences (LQFP100)

LQFP100	STM32L476/ 486	STM32L4R9xx/ S9xx	LQFP100	STM32L476/ 486	STM32L4R9xx/ S9xx
19	VSSA	VSSA/VREF-	59	PD12	VDD12DSI
20	VREF-	VDDA/VREF+	60	PD13	PD8
21	VREF+	PA0	61	PD14	PD9
22	VDDA	PA1	62	PD15	PD10
23	PA0	PA2	63	PC6	PD14
24	PA1	PA3	64	PC7	PD15
25	PA2	VSS	65	PC8	PC6

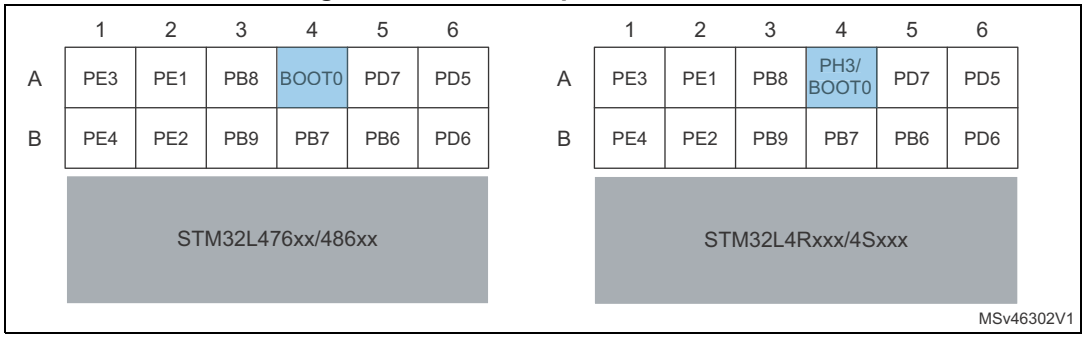
Table 4. STM32L476/486 and STM32L4R9xx/4S9xx pinout differences (LQFP100)

LQFP100	STM32L476/486	STM32L4R9xx/S9xx	LQFP100	STM32L476/486	STM32L4R9xx/S9xx
26	PA3	VDD	66	PC9	PC7
27	VSS	PA4	67	PA8	PC8
28	VDD	PA5	68	PA9	PC9
29	PA4	PA6	69	PA10	PA8
30	PA5	PA7	70	PA11	PA9
31	PA6	PC4	71	PA12	PA10
32	PA7	PB0	72	PA13	PA11
33	PC4	PB1	73	VDDUSB	PA12
34	PC5	PB2	74	VSS	PA13
35	PB0	PE7	75	VDD	VDDUSB
36	PB1	PE8	76	PA14	VSS
37	PB2	PE9	77	PA15	VDD
38	PE7	PE10	78	PC10	PA14
39	PE8	PE11	79	PC11	PA15
40	PE9	PE12	80	PC12	PC10
41	PE10	PE13	81	PD0	PC11
42	PE11	PE14	82	PD1	PC12
43	PE12	PE15	83	PD2	PD0
44	PE13	PB10	84	PD3	PD1
45	PE14	PB11	85	PD4	PD2
46	PE15	VSS	86	PD5	PD3
47	PB10	VDD	87	PD6	PD4
48	PB11	PB12	88	PD7	PD5
49	VSS	PB13	89	PB3	PD6
50	VDD	PB14	90	PB4	PD7
51	PB12	PB15	91	PB5	PB3
52	PB13	VDDDSI	92	PB6	PB4
53	PB14	VCAPDSI	93	PB7	PB5
54	PB15	DSIHOST_D0P	94	BOOT0	PB6
55	PD8	DSIHOST_D0N	95	PB8	PB7
56	PD9	VSSDSI	96	PB9	PH3-BOOT0
57	PD10	DSIHOST_CKP	97	PE0	PB8
58	PD11	DSIHOST_CKN	98	PE1	PB9

1.1.3 UFBGA132 package

Figure 5 illustrates the UFBGA132 pinout differences between STM32L476xx/486xx and STM32L4+ Series microcontrollers.

Figure 5. UFBGA132 pinout differences



1. For the highlighted (blue) terminals, the BOOT0 pin for STM32L476xx/486xx microcontrollers is replaced by a BOOT0 pin shared with a GPIO for STM32L4+ Series microcontrollers.

On the terminal 132 of the UFBGA132 package STM32L476xx/486xx have a BOOT0 pin while STM32L4+ Series have a BOOT0 pin shared with a GPIO. This update is done in order to offer a maximum of flexibility if managing the BOOT0 function by option bytes for STM32L4+ Series devices.

By default, STM32L4+ Series microcontrollers are configured to use this pin as BOOT0 pin (which is the same configuration as STM32L476xx/486xx microcontrollers), in order to keep direct compatibility with the STM32L476xx/486xx's PCB.

SMPS packages

Some devices of the STM32L4 Series offer a package option allowing the connection of an external SMPS.

This is done through two V_{DD12} pins that are replacing two existing pins in the baseline package.

The compatibility of the two V_{DD12} pins is kept between derivatives of the STM32L4 Series. The replaced pins are different across package types but are the same for all derivatives on similar packages. Please refer to the product datasheets for more details.

2 Peripheral migration guide

2.1 STM32 product cross-compatibility

STM32 microcontrollers embed a set of peripherals which can be classified in three groups:

- Peripherals that are by definition common to all products. Those peripherals are identical, so they have the same structure, registers and control bits. There is no need to perform any firmware change to keep the same functionality at the application level after migration. All the features and behavior remain the same.
- Peripherals that are shared by all products but have only minor differences (in general to support new features), so migration from one product to another is very easy and does not need any significant new development effort.
- Peripherals that have considerable changes from one product to another (new architecture or new features for example). For this group of peripherals, the migration requires a new development at application level.

Table 5 summarizes the available peripherals in STM32L476xx/486xx and STM32L4+ Series microcontrollers as well as their compatibility.

Table 5. Peripheral compatibility analysis between STM32L4+ Series and STM32L476xx/486xx microcontrollers(1)

Peripherals		STM32L476xx/ 486xx	STM32L4+ Series	Compatibility	
				Software	Comments
Flash memory(2)	Size (byte)	1 M	2 M	-	-
	Bank	Dual	Dual	-	-
SRAM (Kbytes)	SRAM1	96	192	-	The SRAM2 is contiguous to the SRAM1 (physical address is still in 0x1000 0000) in STM32L4+ Series
	SRAM2(3)	32	64	-	
	SRAM3	-	384	-	New on STM32L4+ Series
DMA		DMA request line is connected directly to peripherals	DMA request line is connected to peripherals through DMAMUX	-	-
FSMC(4) (external memory controller for static memory)		YES	YES	YES	FMC and OCTOSPI are instantiated in two different AHB slave ports in STM32L4+ Series microcontrollers
OCTOSPI		NO	2	-	New peripheral on STM32L4+ Series
OCTOSPIIOM		NO	YES	-	New peripheral on STM32L4R7xx/R9xx/ S7xx/S9xx

Table 5. Peripheral compatibility analysis between STM32L4+ Series and STM32L476xx/486xx microcontrollers(1) (continued)

Peripherals		STM32L476xx/ 486xx	STM32L4+ Series	Compatibility	
				Software	Comments
Timers	Advanced control	2 (16-bit)	2 (16-bit)	YES	-
	General purpose	5 (16-bit) 2 (32-bit)	5 (16-bit) 2 (32-bit)	YES YES	-
	Basic	2 (16-bit)	2 (16-bit)	YES	-
	Low power	2 (16-bit)	2 (16-bit)	YES	-
	Systick timer	1	1	YES	-
	Independent watchdog timer	1	1	YES	-
	Window watchdog timer	1	1	YES	-
Communication interfaces	SPI	3	3	YES	-
	I2C	3	4	YES	-
	USART UART LPUART	3 2 1	3 2 1	YES YES YES	Additional features in STM32L4+ Series(5)
	SAI	2	2	YES	New feature, PDM interface on STM32L4+ Series
	CAN	1	1	YES	-
	USB	OTG FS without clock recovery	OTF FS with clock recovery	YES	Additional clock source (HIS48) in STM32L4+ Series
	SDMMC(6)	YES	YES	-	Additional features in STM32L4+ Series
RTC		YES	YES	YES	APB clock control added (see RCC) in STM32L4+ Series
Tamper pins		YES up to 3	YES up to 3	YES	-
Random generator		YES	YES	YES	Additional clock source (HSI48) in STM32L4+ Series
GPIOs Wake up pins I/Os down to 1.08 V		YES up to 114 YES up to 5 YES up to 14	YES up to 140 YES up to 5 YES up to 14	YES YES YES	Additional I/O PH3 multiplexed with BOOT0 in STM32L4+ Series
Capacitive sensing		YES up to 24	YES up to 24	YES	-

Table 5. Peripheral compatibility analysis between STM32L4+ Series and STM32L476xx/486xx microcontrollers(1) (continued)

Peripherals		STM32L476xx/ 486xx	STM32L4+ Series	Compatibility	
				Software	Comments
DFSDM(7)		YES	YES	YES	Additional features in STM32L4+ Series
12-bit ADC	Instance	3	1		-
	Number of channels	24	24	MOSTLY	
12-bit DAC		2	2	YES	-
Internal voltage reference buffer		1	1	YES	VREFBUF is disabled for package lower than 100-pin
Analog comparator		2	2	YES	-
Operational amplifiers		2	2	YES	-
EXTI		YES	YES	YES	-
RCC		YES	YES	YES	New bits in STM32L4+ Series: To stop the APB clock of the RTC keeping ON the RTC kernel clock in sleep or run modes New HSI48 to manage the clock recovery for USB. It can be the clock source for the RNG and SDMMC HSI16 can be connected to SAI when there is no PLL ON (audio flow detection) MCO can be also output HSI48 PLL P dividers increased More bits to calibrate HSITRIM
PWR(8)		YES	YES	YES	Pull-up/pull-down control bit for stand-by mode for PH3 and additional IOs in STM32L4+ Series
SYSCFG(9)		YES	YES	YES	SRAM2 write protection area increased in STM32L4+ Series
DSI		NA	YES	-	New peripheral on STM32L4R9xxx/S9xxx
HASH		NA	YES	-	-

Table 5. Peripheral compatibility analysis between STM32L4+ Series and STM32L476xx/486xx microcontrollers(1) (continued)

Peripherals	STM32L476xx/ 486xx	STM32L4+ Series	Compatibility	
			Software	Comments
AES	NA	YES	-	-
QUADSPI	YES	NA	-	QUADSPI is not available in STM32L4+ Series
LCD	YES	NA	-	LCD is not available in STM32L4+ Series
SWPMI	YES	NA	-	SWPMI is not available in STM32L4+ Series
LTDC	NA	YES	-	New peripheral on STM32L4R7xx/R9xx/ S7xx/S9xx
DCMI	NA	YES	-	New peripheral on STM32L4+ Series
DMA2D	NA	YES	-	New peripheral on STM32L4+ Series
DMAMUX1	NA	YES	-	New peripheral on STM32L4+ Series
GFXMMU	NA	YES	-	New peripheral on SM32L4Rxxx/4Sxxx

1. The gray-background cells highlight the main improvements of STM32L4+ Series versus STM32L476xx/486xx devices.
2. Refer to *Section 2.4: Flash memory* for more details.
3. Refer to *Section 2.5: SRAM2 memory* or more details.
4. Refer to *Section 2.6: Flexible static memory controller (FSMC)* or more details
5. Refer to *Section 2.15: Universal synchronous asynchronous receiver transmitter (USART)* for more details.
6. Refer to *Section 2.16: Secure digital input/output MultiMediaCard interface (SDMMC)* for more details.
7. Refer to *Section 2.17: Digital filter for sigma delta modulators (DFSDM)* for more details.
8. Refer to *Section 2.11: Power control (PWR)* for more details.
9. Refer to *Section 2.12: System configuration controller (SYSCFG)* for more details.

Note: Most of the bugs known for STM32L476xx/486xx microcontrollers have been corrected for STM32L4+ Series microcontrollers. Please refer to the corresponding product errata sheets to find out the remaining bugs.

2.2 Boot modes

The boot mode selection changes slightly between STM32L4+ Series and STM32L476xx/486xx microcontrollers. The BOOT0 input pin is shared with a GPIO (PH3).

A new option bit nSWBOOT0 is added allowing to choose if the BOOT0 value is coming from the IO pin or from the option bit value. *Table 6* presents the STM32L4+ Series microcontrollers boot modes and highlights in gray-background cells the new modes.

Table 7 presents STM32L476xx/486xx microcontrollers boot modes.

Table 6. Boot modes for STM32L4+ Series(1)

nBOOT1 FLASH_OPTR[23]	nBOOT0 FLASH_OPTR[27]	BOOT0 pin PH3	nSWBOOT0 FLASH_OPTR[26]	Main Flash empty (2)	Boot Memory Space Alias
x	x	0	1	0	Main Flash memory is selected as boot area
x	x	0	1	1	System memory is selected as boot area
x	1	x	0	x	Main Flash memory is selected as boot area
0	x	1	1	x	Embedded SRAM1 is selected as boot area
0	0	x	0	x	Embedded SRAM1 is selected as boot area
1	x	1	1	x	System memory is selected as boot area
1	0	x	0	x	System memory is selected as boot area

1. x: do not care.
2. A Flash empty check mechanism is implemented to force the boot from system Flash if the first Flash memory location is not programmed (0xFFFF FFFF) and if the boot selection was configured to boot from the main Flash.

Table 7. Boot modes for STM32L476xx/486xx

Boot memory space	BOOT1	BOOT0 pin
Main Flash	do not care	0
System Flash	0	1
Embedded SRAM1	1	1

2.3 Memory mapping

Table 8 compares the peripherals register boundary addresses for STM32L4+ Series and STM32L476xx/486xx microcontrollers.

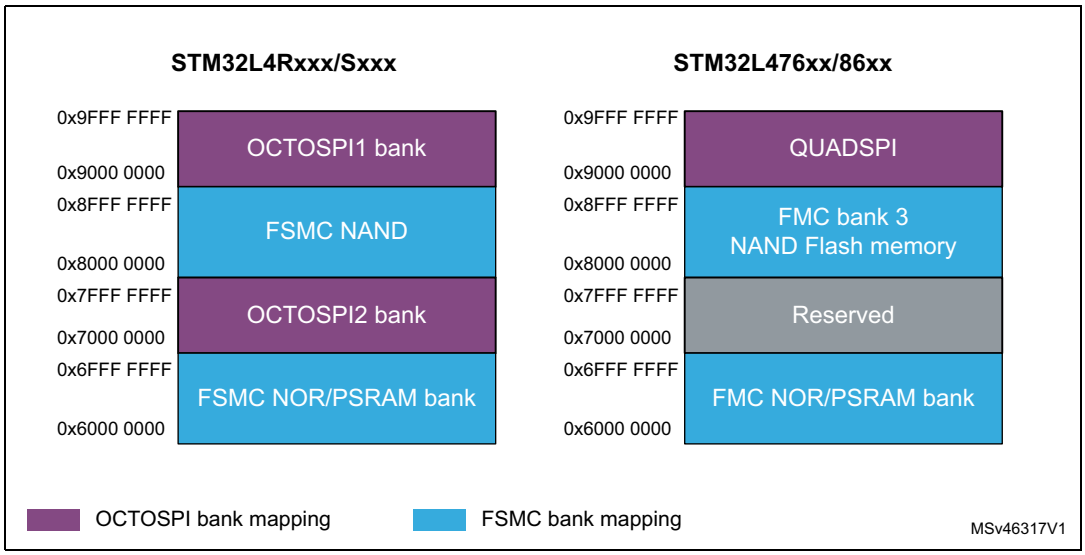
Table 8. Peripherals register boundary addresses comparison

Peripheral	Bus	STM32L4+ Series	STM32L476xx/486xx bus address
HASH	AHB2	0x5006 0400	NA
AES	AHB2	0x5006 0000	NA
DCMI	AHB2	0x5005 0000	NA
GPIOI	AHB2	0x4800 2400	NA
DMA2D	AHB1	0x4002 B000	NA
DMAMUX1	AHB1	0x4002 0800	NA
I2C4	APB1	0x4000 8400	NA
OCTOSPIIOM	AHB2	0x5006 1C00	NA
OCTOSPI1	AHB2	0xA000 1000	NA
OCTOSPI2	AHB2	0xA000 1400	NA
SDMMC1	APB2/AHB2(1)	0x5006 2400	0x4001 2800
GFXMMU	AHB1	0x4002 C000	NA
DSI	APB2	0x4001 6C00	NA
LTDC	APB2	0x4001 6800	NA

1. APB2 for STM32L476xx/486xx and AHB2 for STM32L4+ Series.

The *Figure 6* presents the external memory mapping differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers.

Figure 6. External memory mapping differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers



2.4 Flash memory

Table 9 details the differences between the Flash interface on STM32L4+ Series and STM32L476xx/486xx microcontrollers.

Table 9. Flash memory differences between STM32L4+ Series and STM32L476xx/486xx

Flash	STM32L4+ Series	STM32L476xx/486xx
Main/program memory	Up to 2 Mbytes Split in 2 banks 256 pages: Single Bank page size is: 8 Kbytes. Dual Bank page size is: 4 Kbytes	Up to 1 Mbytes Split in 2 banks 256 pages of 2 Kbytes per bank
Features	Read while write (RWW) Dual bank boot	
Read access	Single bank mode: read access of 128 bits Dual bank mode: read access of 64 bits	Read access of 64 bits
Wait states	Up to 5 (depending on the supply voltage and the frequency)	Up to 4 (depending on the supply voltage and the frequency)
Flash empty check	YES	NO
Protections	Write protection: – Dual Bank: 2 areas per bank – Single Bank: 4 areas per bank PCROP protection: – Dual Bank: 2 PCROP area per bank – Single Bank: 1 PCROP area per bank	Write protection: 2 areas per bank PCROP protection: 1 PCROP area per bank
One time programmable (OTP) memory	1 Kbyte	
Interface	0x4002 2000 – 0x4002 23FF	

Table 9. Flash memory differences between STM32L4+ Series and STM32L476xx/486xx (continued)

Flash	STM32L4+ Series	STM32L476xx/486xx
Option bytes	Bank 1: 0x1FF0 0000 – 0x1FF0 000F Bank 2 : 0x1FF0 1000 – 0x1FF0 100F	Bank 1: 0x1FFF 7800 – 0x1FFF 780F Bank 2: 0x1FFF F800 – 0x1FFF F80F
	nBOOT0	NA
	nSWBOOT0	NA
	SRAM2_RST	
	SRAM2_PE	
	nBOOT1	
	DBANK	NA
	DB1M (1 Mbyte/ 512 Kbytes Dual-Bank Flash with contiguous addresses)	DUALBANK (256 Kbytes/ 512 Kbytes Dual-Bank Flash)
	BFB2	
	WWDG_SW	
	IWDG_STDBY	
	IWDG_STOP	
	IWDG_SW	
	nRST_SHDW	
	nRST_STDBY	
	nRST_LEV	
	BOR_LEV	
	RDP	
	PCROP1_STRT[16:0]	PCROP1_STRT[15:0]
	PCROP_RDP	
	PCROP1_END[16:0]	PCROP1_END[15:0]
	WRP1A_STRT	
	WRP1A_END	
	WRP1B_STRT	
	WRP1B_END	
	PCROP2_STRT[16:0]	PCROP2_STRT[15:0]
	PCROP2_END[16:0]	PCROP2_END[15:0]
	WRP2A_STRT	
	WRP2A_END	
	WRP2B_STRT	
	WRP2B_END	

2.5 SRAM2 memory

There are very few differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers concerning the SRAM2 memory. See *Table 10* for the detailed differences.

Table 10. SRAM2 differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers

SRAM2	STM32L4+ Series	STM32L476xx/486xx
Size	64 Kbytes	32 Kbytes
Address	0x1000 0000 - 0x1000 FFFF	0x1000 0000 - 0x1000 7FFF
Parity check	YES	
Write protection	YES - 1 Kbyte granularity	
Read protection	RDP	
SRAM2 Erase	System reset or Software reset	
SRAM2 contiguous with SRAM1	YES	NO

Note: The following STM32L476xx/486xx limitation has been fixed for STM32L4+ Series microcontrollers: if a read occurs during an erase operation, the CPU is not stalled and the value returned is deterministic and equal to 0x0000 0000.

2.6 Flexible static memory controller (FSMC)

The *Table 11* presents the FSMC interface differences between of STM32L4+ Series and STM32L476xx/486xx microcontrollers.

Table 11. FSMC differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers

FSMC		STM32L4+ Series	STM32L476xx/486xx
External memory interfaces		– SRAM – NOR/NAND memories – PSRAM – NAND Flash memory with ECC hardware – FRAM Ferroelectric RAM	– SRAM – NOR/NAND memories – PSRAM – NAND Flash memory with ECC hardware
Features	Data bus width	8-bit or 16-bit	
	New timing	– NBL setup timing – Data hold timing – Clock divider ratio 1	NA
Registers	FSMC_BCRx	Bits[23:22]: NBLSET[1:0]	NA
	FSMC_BTRx	Bits[31:30]: DATAHLD[1:0]	NA

2.7 OctoSPI interface (OCTOSPI)

The OCTOSPI peripheral is available only on STM32L4+ Series microcontrollers. It shares the same features than the Quad-SPI peripheral in STM32L476xx/486xx and additionally it supports octal SPI memories.

The Octo-SPI is a specialized communication interface which targets single, dual, quad or octal SPI memories. It can be configured in three modes: Indirect mode, Status-polling mode and Memory-mapping mode.

2.8 NESTED vectored interrupt controller (NVIC)

Table 12 presents the interrupt vectors differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers.

Table 12. Interrupt vector differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers

Position	STM32L4+ Series	STM32L476xx/486xx
82	HASH and CRS	NA
83	I2C4_EV	
84	I2C4_ER	
85	DCMI	
90	DMA2D	
91	LCD-TFT	
92	LCD-TFT_ER	
93	GFXMMU	
94	DMAMUX1_OVR	

2.9 Extended interrupt and event controller (EXTI)

Table 24 presents the EXTI line differences between STM32L476xx/486xx and STM32L4+ Series devices.

Table 13. EXTI lines connections differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers

EXTI line	STM32L4+ Series	STM32L476xx/486xx
34	Reserved	SWPMI1 wakeup
39	Reserved	LCD wakeup
40	I2C4 wakeup	NA

2.10 Reset and clock control (RCC)

Table 14 highlights the main differences related to RCC (reset and clock controller) between STM32L4+ Series and STM32L476xx/486xx microcontrollers.

Table 14. RCC differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers

RCC		STM32L4+ Series	STM32L476xx/486xx
		Clock sources	
USB OTG FS		MSI clock PLL / Q PLLSAI1 / Q HSI48	MSI clock PLL / Q PLLSAI1/ Q
RNG/SDMMC			
DSI		PLLSAI2 / Q divider HSE	NA
LTDC		PLLSAI2 clock / R divider/ PLLSAI2RDIV	NA
USARTs	USART1	APB2 clock HSI16 LSE SYSCLK	
	USART2 and 3	APB1 clock HSI16 LSE SYSCLK	
	UART4 and 5	APB1 clock HSI16 LSE SYSCLK	
LPUART1		APB1 clock HSI16 LSE SYSCLK	
I2Cs		APB1 HSI16 SYSCLK	
SPIs		APB2 clock for SPI1 APB1 clock for SPI2 and SPI3	
SAI1, SAI2		PLLSAI1 clock /P divider PLLSAI2 clock /P divider PLL clock /P divider HSI16 for audio flow detection	PLLSAI1 clock /P divider PLLSAI2 clock /P divider PLL clock /P divider

Table 14. RCC differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers (continued)

RCC	STM32L4+ Series	STM32L476xx/486xx
	Clock sources	
OCTOSPI	MSI SYSCLK PLL clock / Q divider	NA
IWDG	LSI clock	
WWDG	APB1 clock	
ADC	SYSCLK PLLSAI1/ R divider	
RTC	HSE32 LSE LSI Register clock disabling (not the kernel one) in run or sleep modes RTCAPBEN in RCC_APB1ENR1	HSE LSE LSI
MCO	LSI LSE SYSCLK HSI16 HSE PLLCLK MSI HSI48	LSI LSE SYSCLK HSI16 HSE PLLCLK MSI
PLL	3 PLLs (PLL, PLLSAI1, PLLSAI2)	
PLLM divider	PLLM: 1 to 16	PLLM: 1 to 8
PLLN factor	PLLN: 8 to 127	PLLN: 8 to 86
PLLP divider	PLLP: 2 to 31	PLLP: 7 or 17
PLLSAI1M divider	PLLSAI1M: 1 to 16	NA
PLLSAI1N factor	PLLSAI1N: 8 to 127	PLLSAI1N: 8 to 86
PLLSAI1P divider	PLLSAI1P: 7 or 17 PLLSAI1PDOV: 2 to 31	NA
PLLSAI2M divider	PLLSAI2M: 1 to 16	NA
PLLSAI2N factor	PLLSAI2N: 8 to 127	PLLSAI2N: 8 to 86
PLLSAI2P divider	PLLSAI2P: 7 or 17 PLLSAI2PDIV: 2 to 31	PLLSAI2P: 7 or 17
HSI16	HSITRIM[6:0]	HSITRIM[4:0]
HSI48	RC with clock recovery used for USB/RNG/SDMMC	NA

2.11 Power control (PWR)

Table 15 shows the low-power mode differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers

Table 15. Low-power mode differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers

PVM	STM32L4+ Series	STM32L476xx/486xx
Low-power modes	STOP0	
	STOP1	
	STOP2 SRAM3 enabled RRSTP = “1” within PWR_CR1 register	STOP2
	STOP2 SRAM3 disabled RRSTP = “0” within PWR_CR1 register	
	Standby	
	Shutdown	

The *Table 16* shows the power control register differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers.

Table 16. PWR differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers

Power control register	STM32L4+ Series	STM32L476xx/486xx
PWR_PUCRH	PUy (y=0..15)	PUy (y=0..1)
PWR_PDCRH	PDy (y=0..15)	PDy (y=0..1)
PWR_PUCRI	PUy (y=0..15)	NA
PWR_PDCRI	PDy (y=0..15)	NA
PWR_CR1	Bit[4]: RRSTP	NA
PWR_CR5	New register Bit[8]: R1MODE	NA

Dynamic voltage scaling management

The difference between STM32L4+ Series and STM32L476xx/486xx in dynamic voltage scaling management is in the high-performance range. For STM32L4+ Series, the main regulator operates in two modes following the R1MODE bit in the PWR_CR5 register:

- Main regulator range 1 normal mode: provides a typical output voltage at 1.2 V. It is used when the system clock frequency is up to 80 MHz. The Flash access time for read access is minimum, write and erase operations are possible.
- Main regulator range 1 boost mode: provides a typical output voltage at 1.28 V. It is used when the system clock frequency is up to 120 MHz. The Flash access time for read access is minimum, write and erase operations are possible. To optimize the power consumption it is recommended to select the range 1 boost mode when the system clock frequency is greater than 80 MHz.

The *Table 17* shows the voltage regulator range differences between STM32L4Rxx/4Sxxx and STM32L476xx/486xx microcontrollers followed by the recommended sequence to switch between voltage regulator ranges.

Table 17. Main regulator configuration differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers

SYSCLK	STM32L4+ Series	STM32L476xx/486xx
80 MHz < SYSCLK <= 120 MHz	Range 1 boost mode	NA
26 MHz < SYSCLK <= 80 MHz	Range 1 normal mode (default)(1)	
SYSCLK <= 26 MHz	Range 2	

1. Normal mode for STM32L4+ Series.

The sequence to go from Range 2 to Range 1 is:

1. Program the VOS bits to “10” in the PWR_CR1 register.
2. Wait until the VOSF flag is cleared in the PWR_SR2 register.
3. Adjust the number of wait states according to the new frequency target in Range 1 (LATENCY bits in the FLASH_ACR).
4. Increase the system’s frequency by following the procedure below:

- If the system frequency is 26 MHz < SYSCLK <= 80 MHz, select the range 1 normal mode, just configure and switch to PLL for a new system frequency.
- If the system frequency is SYSCLK > 80 MHz, select the Range 1 boost mode:
  - The system clock must be divided by two using the AHB prescaler before switching to a higher system frequency.
  - Clear the R1MODE bit in the PWR_CR5 register.
  - Configure and switch to PLL for a new system frequency.
  - Wait for at least 1 us and then reconfigure the AHB prescaler to get the needed HCLK clock frequency.

The sequence to switch from Range 1 normal mode to Range 1 boost mode is:

1. The system clock must be divided by 2 using the AHB prescaler before switching to a higher system frequency.
2. Clear the R1MODE bit is in the PWR_CR5.
3. Adjust the number of wait states according to the new frequency target in Range 1 boost mode.
4. Configure and switch to new system frequency.
5. Wait for at least 1 us and then reconfigure the AHB prescaler to get the needed HCLK clock frequency.
6. Reconfigure the AHB prescaler to get the needed HCLK clock frequency.

2.12 System configuration controller (SYSCFG)

The *Table 18* shows the SYSCFG differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers.

Table 18. SYSCFG differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers

SYSCFG registers	STM32L4+ Series	STM32L476xx/486xx
SYSCFG_CFGR1	Bit[23] I2C4_FMP	NA
	Bit[9] ANASWVDD	NA
SYSCFG_EXTICR1	Bits EXTIy[3:0] (y=0..3)	Bits EXTIy[2:0] (y=0..3)
SYSCFG_EXTICR2	Bits EXTIy[3:0] (y=4..7)	Bits EXTIy[2:0] (y=4..7)
SYSCFG_EXTICR3	Bits EXTIy[3:0] (y=8..11)	Bits EXTIy[2:0] (y=8..11)
SYSCFG_EXTICR4	Bits EXTIy[3:0] (y=12..15)	Bits EXTIy[2:0] (y=11..15)

The *Table 19* describes when the ANASWVDD bit and the BOOSTEN bit should be set or reset depending on the voltage setting.

Table 19. BOOSTEN and ANASWVDD set/reset

VDD	VDDA	BOOSTEN	ANASWVDD
-	> 2.4V	0	0
> 2.4V	< 2.4V	0	1
< 2.4V	< 2.4V	1	0

The *Table 20*presents the memory mapping selection differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers

Table 20. Memory mapping differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers

-	STM32L4+ Series	STM32L476xx/486xx
Memory mapping selection at address 0x0000 0000	– Main Flash memory – System Flash memory – FMC bank1 (NOR/PSRAM 1 and 2) – SRAM1 – OCTOSPI1 memory – OCTOSPI2 memory	– Main Flash memory – System Flash memory – SRAM1 – QUADSPI memory

2.13 Interconnect matrix

The STM32L4+ Series's interconnect matrix is an enlarged set of the STM32L476xx/486xx's interconnect matrix since the number of peripherals is increased.

In particular, the STM32L4+ Series microcontrollers interconnect matrix includes one additional master for DMA2D, LTDC, SDMMC1, GFXMMU and the SRAM3, GFXMMU, FMC, OCTOSPI1, OCTOSPI2 slave is split into two separate slave ports.

Please refer to the *Table 5: Peripheral compatibility analysis between STM32L4+ Series and STM32L476xx/486xx microcontrollers* to see more details of the differences between the concerned products.

2.14 Direct memory access controller (DMA)

There are two DMA master interfaces for STM32L4+ Series microcontrollers, same as for STM32L476xx/486xx microcontrollers. The DMA channels connections corresponding to peripherals that are present only for STM32L4+ Series microcontrollers, are left free for STM32L476xx/486xx microcontrollers.

For STM32L4+ Series, each DMA request is connected in parallel to all the channels of the DMAMUX request line multiplexer. In STM32L476xx/486xx, each DMA channel is connected to dedicated hardware DMA requests.

The DMAMUX request multiplexer allows a DMA request line to be routed between the peripherals and the DMA controllers of the product. The routine function is ensured by a programmable multi-channel DMA request line multiplexer. Each channel select a unique DMA request line, unconditionally or synchronously with events from its DMAMUX synchronization inputs.

2.15 Universal synchronous asynchronous receiver transmitter (USART)

The difference between STM32L4+ Series and STM32L476xx/486xx in U(S)ART is mainly due to the additional features on STM32L4+ Series which are:

- Support of the ISO78716-3 smartcard protocol.
- Feature two internal FIFOs for transmit and receive data, and each FIFO can be enabled /disabled by software; they also come with a status flag.
- SPI slave transmission underrun error flag.

The *Table 19* presents the U(S)ART differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers.

Table 21. U(S)ART differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers

U(S)ART		STM32L4+ Series	STM32L476xx/486xx
UART/USART		5	
Interrupt		23 interrupt sources with flags	12 interrupt sources with flags
Features		Two internal FIFOs for transmit and receive data	NA
		SPI slave transmission underrun error flag	
		Lin mode SPI Master IrDA SIR ENDEC block Hardware flow control (CTS/RTS) Continuous communication using DMA Multiprocessor communication Single-wire half-duplex communication Support for Modbus communication Receiver timeout interrupt Auto baud rate detection	
U(S)ART registers	USARTx_CR1	Bit[29]: FIFOEN	NA
		Bit[30]: TXFEIE	
		Bit[31]: RXFFIE	
	USARTx_CR2	Bit[0]: SLVEN	
		Bit[3]: DIS_NSS	
	USARTx_CR3	Bit[23]: TXFTIE	
		Bit[24]: TCBGTIE	
		Bits[27:25]: RXFTCFG	
		Bit[28]: RXFTIE	
		Bits[31:29]: TXFTCFG	
	USARTx_ISR	Bit[13]: UDR	
		Bit[23]: TXFE	
		Bit[24]: RXFF	
		Bit[25]: TCBGT	
		Bit[26]: RXFT	
		Bit[27]: TXFT	
	USARTx_ICR	Bit[5]: TXFECF	
		Bit[7]: TCBGTC	
		Bit[13]: UDRCF	

2.16 Secure digital input/output MultiMediaCard interface (SDMMC)

The *Table 22* presents the differences between the SDMMC interface of the STM32L4+ Series and STM32L476xx/486xx microcontrollers

Table 22. SDDMC differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers

SDMMC	STM32L4+ Series	STM32L476xx/486xx
Bus	AHB2	APB2
Clock source	MSI clock PLL /Q PLLSAI1 /Q HSI48	MSI clock PLL /Q PLLSAI1/Q
Features	Full compliance with MultimediaCard Systm Specification Version 4.51. Card support for three different databus modes: 1-bit (default), 4-bit and 8-bit	Full compliance with MultimediaCard System Sepecification version 4.2. Card support for three different databus modes: 1-bit (default), 4-bit and 8-bit
	Full compliance with SD memory card specifications version 4.1	Full compliance with SD Memory Card specification Version 2.0
	Full compliance with SDIO card specification version 4.0. Card support for two different databus modes: 1-bit (default) and 4-bit	Full compliance with SD I/O card specification Version 2.0. Card support for two different databus modes: 1-bit (default) and 4-bit
	Data transfer up to 104 Mbyte/s for the 8-bit mode	Data transfer up to 50 MHz for the 8- bit mode
	SDDMC IDMA: is used to provide high-speed transfer between the SDMMC FIFO and the memory. The AHB master optimizes the bandwidth of the system bus. The SDMMC internal DMA (IDMA) provides one channel to be used either for transmit or receive	NA

2.17 Digital filter for sigma delta modulators (DFSDM)

The *Table 23* presents the DFSDM differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers.

Table 23. DFSDM differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers

DFSDM features		STM32L4+ Series	STM32L476xx/486xx
Features	Number of channels	8	
	Number of filters	4	
	Input from internal ADC	X	-
	Supported trigger sources	12(1)	11
	Pulses skipper	X	-
DFSDM register	DFSDMx_CR1	Bits[12:8]: JEXTSEL[4:0]	Bits[10:8]: JEXTSEL[2:0]
	DFSDM_CHyDLYR	New register Bits[5:0]: PLSSKP[5:0]	NA

1. The LPTIM1 is the new trigger source.

2.18 USB on-the-go full-speed (OTG_FS)

Table 24 presents the USB OTG differences between STM32L476xx/486xx and STM32L4+ Series microcontrollers.

Table 24. USB_OTG Implementation for STM32L476xx/486xx and STM32L4+ Series microcontrollers

USB features	OTG_FS for STM32L476xx/486xx	OTG_FS for STM32L4+ Series
Device bidirectional endpoints (including EP0)	6	
Host mode channels	12	
Size of dedicated SRAM	1.2 Kbytes	
USB 2.0 Link Power Management (LPM) support	X	
OTG revision supported	1.3 and 2.0	2.0
Attach detection protocol (ADP) support	X	
Battery charging detection (BCD) support	X	
PSRST/ERRATIM	-	X

2.19 Debug support (DBG)

Table 25 presents the DEBUG differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers.

Table 25. DEBUG differences between STM32L4+ Series and STM32L476xx/486xx microcontrollers

DEBUG register	STM32L4+ Series	STM32L476xx/486xx
DBGMCU_APB1FZR2	Bit[1]: DBG_I2C4_STOP	NA

3 Revision history

Table 26. Document revision history

Date	Revision	Changes
31-Aug-2017	1	Initial release.