AN4617
Application note

Migrating between STM32F0 and STM32L0 microcontrollers

Introduction

For designers of STM32 microcontroller applications, it is important to be able to easily replace one microcontroller type with another one of the same product family.

Migration of an application to a different microcontroller is frequent, because of new product requirements: this usually requires increase of memory density, and of the I/Os number. I/Os. On the other hand, cost reduction objectives may force the switch to smaller components, or shrink of the PCB area. Another important factor is the effort to decrease overall power consumption, especially if application is sensitive to that topic.

This application note is written to help users to analyze the steps needed to migrate from a STM32F0 microcontroller to one of the STM32L0 series and vice versa. It summarizes the most relevant informations and lists the vital aspects to be addressed.

To migrate an application between the STM32F0 and the STM32L0 series, user must carefully analyze the hardware migration, the peripheral migration and then the firmware migration.

To benefit fully from the information contained in this application note, the user should be familiar with the STM32 microcontroller family, for this we recommend to look at the following documents that are available on www.st.com:

- the STM32F0 series reference manuals (RM0091 and RM0360) and STM32F0 datasheets;
- the STM32L0 series reference manuals (RM0367, RM0376 and RM0377) and STM32L0 datasheets.

For an overview of the whole STM32 series and a comparison of the different features of each STM32 product series, please refer to AN3364 "*Migration and compatibility guidelines for STM32 microcontroller applications*".

Table 1 lists the products concerned by this application note.

Table 1. Applicable products

Type	Product series
Microcontrollers	STM32F0
	STM32L0

Contents

List of tables

1 Hardware migration

Basically the STM32F0 and STM32L0 series are pin-to-pin compatible and directly replaceable.

All peripherals share most of the pins, but there are some minor differences between packages. When performing the migration, user can gain or lose some GPIOs, and some power supply pins can become incompatible, so a careful check of pinouts has to be carried out.

Differences can appear at handling BOOT0 pin, as this pin can be provided optionally according to internal option bits. Standard OSC pin can be replaced by OSC32 for some small packages. Some products feature the NPOR pin, too.

The possible impacted pins are summarized in *Table 2*.

Table 2. STM32F0 series and STM32L0 series pinout differences

Package LQFP				STM32F0 series	STM32L0 series
32 pin	48 pin	64 pin	100 pin		
-	1	1	6	VBAT/VDD	VDD/VLCD
2	-	-	-	PF0-OSC_IN	PC14-OSC32_IN
3	-	-	-	PF1-OSC_OUT	PC15-OSC32_OUT
-	5	5	-	PF0-OSC_IN	PH0-OSC_IN
	6	6		PF1-OSC_OUT	PH1-OSC_OUT
-	-	-	10	PF9	PH9
-	-	-	11	PF10	PH10
17	-	-	-	VDDIO2	VDD
-	-	-	18	PF2	VSSA
-	-	-	19	VSSA	VREF-
-	-	18	-	PF4/VSS	VSS
-	-	19	-	PF5/VDD	VDD
-	-	-	20	VDDA	VREF+
-	-	-	21	PF3	VDDA
-	20	28	-	PB2/NPOR	PB2
-	35	47	-	PF6/VSS	VSS
-	-	-	73	PF6	VDD
-	36	48	75	PF7 / VDDIO2(1)	VDD / VDD_USB(1)
31	44	60	-	BOOT0/PF11-BOOT0/PB8	BOOT0

1. When power supply is applied at its pin, it powers USB, PA8-15, PC6-12 and PD2 for the F0 family, USB, and USB associated pins PA11 and PA12 for the L0 family.

1.1 Boot mode compatibility

There is no difference in boot mode configuration possibilities between F0 and L0 series but the selection of boot mode may be slightly different.

Both series get the nBOOT1 value from an option bit located in the User option bytes. Some more recent L0 and F0 products can do the same with the BOOT0 value optionally, too, while BOOT0 pin can be used as GPIO in dependency on setting BOOT_SEL option bit.

Table 3 summarizes the different configurations available for selecting the Boot mode.

Table 3. Boot modes

F0/L0 boot mode configuration				Boot mode (memory aliased in boot memory)
nBOOT1 bit	BOOT0 external pin	nBOOT0 bit(1)	BOOT_SEL bit(1)	
x	0	x	1	Main Flash memory
1	1	x	1	System memory
0	1	x	1	Embedded SRAM
x	x	1	0	Main Flash memory
1	x	0	0	System memory
0	x	0	0	Embedded SRAM

1. The BOOT_SEL and nBOOT0 option bits are not present in all L0 and F0 products.

Embedded bootloader for both STM32F0 and STM32L0 devices uses UART and USB. Some devices feature I2C as well, while SPI can be an additional option in the L0 series.

2 Peripheral migration

As shown in *Table 4*, there are three categories of peripherals. The common peripherals are supported with the dedicated firmware library without any modification, except if the peripheral instance is no longer present. You can change the instance and, of course, all the related features (clock configuration, pin configuration, interrupt/DMA request).

The modified peripherals (such as ADC, RCC and RTC) are partially different from those of the F0 series, and dedicated drivers are updated to take advantage of the enhancements and the new features in the L0 series.

Most of the modified peripherals in the L0 series are enhanced to have smaller silicon print with features designed to offer advanced high-end capabilities in economical end products, and to fix some limitations present in the F0 series.

2.1 STM32 product cross-compatibility

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

1. The first category is for the 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.
2. The second category is that of peripherals shared by all products, but have minor differences (in general to support new features). The migration from one product to another is very easy and does not need any significant new development effort.
3. The third category is for peripherals considerably different from one product to another (new architecture, new features...). For this category of peripherals, the migration will require a dedicated development at the application level.

Table 4 gives a general overview of this classification for the two STM32 series under consideration.

Table 4. STM32 peripheral compatibility analysis F0 versus L0 series

Peripheral	F0 series	L0 series	Compatibility		
			Possible differences from user point of view	Pinout	FW driver compatibility
SPI	Yes++	Yes	FIFO availability Data size selection Data packing	Identical	Partial
WWDG	Yes	Yes+	None	NA	Full
IWDG	Yes	Yes+	None	NA	Full
DBGMCU	Yes	Yes+	Sleep mode control	Identical for the SWD	Partial
CRC	Yes	Yes	None	NA	Full
NVIC	Yes	Yes+	EXTI vectors Wake up capability Internal event sources	Identical for the same feature	Partial
DMA	Yes	Yes+	Remapping control	NA	Partial
TIM	Yes	Yes+	Number of timers Functionality Low power timer	Identical for the same feature	Partial
PWR	Yes	Yes+	Dynamic voltage scaling BOR threshold control Low power modes Fast wake up Internal reset sources	Identical for the same feature	Partial
RCC	Yes	Yes+	Internal clock sources CSS on LSE Internal clock measurement Calibration capability	Identical for the same feature except OSC32 for small packages	Partial
USART	Yes	Yes+	Low power UART	Identical for the same feature	Partial
I2C	Yes	Yes+	None	Identical	Full
DAC	Yes	Yes+	Supply range HW trigger	Identical for the same feature	Partial
ADC	Yes	Yes++	Supply range HW trigger HW oversampling Low frequency mode Interrupt sources	Identical for the same feature	Partial
COMP	Yes	Yes+	Ultra low power Speed, hysteresis Outputs/Inputs	Identical for the same feature	Partial
RTC	Yes	Yes+	Number of alarms Triggers from tamper events	Identical for the same feature	Partial
FLASH	Yes	Yes+	Sector & firewall protection	NA	Partial
RAM	Yes	Yes	Firewall protection (L0 only) Parity check (F0 only)	NA	Partial

Table 4. STM32 peripheral compatibility analysis F0 versus L0 series (continued)

Peripheral	F0 series	L0 series	Compatibility		
			Possible differences from user point of view	Pinout	FW driver compatibility
USB FS Device	Yes	Yes	SRAM partially shared with CAN	Identical	Full
Touch Sensing	Yes	Yes+	Dedicated IO pins	Identical for the same feature	Full
GPIO	Yes	Yes+	GPIO mapping and configuration locking	Identical for the same feature	Full
CAN	Yes	No	NA	NA	NA
HDMI-CEC	Yes	No	NA	NA	NA
LCD	No	Yes	NA	NA	NA
AES	No	Yes	NA	NA	NA
RNG	No	Yes	NA	NA	NA
IRTIM	Yes	No	NA	NA	NA
Data EEPROM	No	Yes	NA	NA	NA

Note: Yes++ = New feature or new architecture
Yes+ = Same feature, but specification change or design enhancement
Yes = Feature available
No = Feature not available
NA= Not applicable

2.2 System architecture

The STM32F0 MCU family has been designed to target an entry-level market, with low-power capabilities and easy handling. To achieve this target and at the same time offer advanced high-end features, it uses a 32-bit Cortex®-M0 core. Its small silicon area, coupled with minimal code footprint, results in low-cost applications with 32 bits performance.

STM32L0 MCU family development goes in the similar way, one step further concerning optimization of power consumption. Design is based on advanced Cortex®-M0+ core while targeting application operating on battery or supplied by energy harvesting. Autonomous peripherals capable to operate at low power mode reduce the core load and power consumption.

The basic difference between Cortex®-M0 and Cortex®-M0+ cores is three stages pipe line in M0, two stages for M0+, both with maximum prefetch capacity up to two 16-bit instructions. In addition, M0+ core features dedicated single cycle I/O port and memory protection unit used by some L0 devices. A detailed comparison between these two cores can be found on the ARM® website, while the enhancements of the Cortex M0+ embedded in STM32L0 are described in Cortex®-M0+ Programming manual (PM0223).

2.3 Memory mapping

The peripheral address mapping has been changed in the F0 series compared to the L0 series. The main change concerns the GPIOs, they are on AHB bus in the F0 series and on Cortex®-M0+ specific IOPORT in the L0 series.

Table 5 provides the peripheral address mapping correspondence between F0 and L0 series (differences between the two series are highlighted with bold type).

Table 5. IP bus mapping cross reference

Peripheral	STM32F0 series		STM32L0 series	
	Bus	Base address	Bus	Base address
AES	AHB1	**NA**	AHB	**0x40026000**
RNG		**NA**		**0x40025000**
TSC		0x40024000		0x40024000
CRC		0x40023000		0x40023000
FLITF		0x40022000		0x40022000
RCC		0x40021000		0x40021000
DMA / DMA1		0x40020000		0x40020000
DMA2		0x40020400		NA
GPIOH	AHB2	**NA**	IOPORT	**0x50001C00**
GPIOF		**0x48001400**		**NA**
GPIOE		**0x48001000**		**0x50001000**
GPIOD		**0x48000C00**		**0x50000C00**
GPIOC		**0x48000800**		**0x50000800**
GPIOB		**0x48000400**		**0x50000400**
GPIOA		**0x48000000**		**0x50000000**

Table 5. IP bus mapping cross reference (continued)

Peripheral	STM32F0 series		STM32L0 series	
	Bus	Base address	Bus	Base address
DBGMCU	APB	0x40015800	APB2	0x40015800
TIM17		**0x40014800**		**NA**
TIM16		**0x40014400**		**NA**
TIM15		**0x40014000**		**NA**
USART1		0x40013800		0x40013800
SPI1 / I2S1		0x40013000		0x40013000
TIM1		**0x40012C00**		**NA**
ADC / ADC1		0x40012400		0x40012400
USART8		**0x40011C00**		**NA**
USART7		**0x40011800**		**NA**
USART6		**0x40011400**		**NA**
Firewall		**NA**		**0x40011C00**
TIM22		**NA**		**0x40011400**
TIM21		**NA**		**0x40010800**
EXTI		0x40010400		0x40010400
SYSCFG + COMP		0x40010000		0x40010000

Table 5. IP bus mapping cross reference (continued)

Peripheral	STM32F0 series		STM32L0 series	
	Bus	Base address	Bus	Base address
LPTIM1	APB	**NA**	APB1	**0x40007C00**
CEC		**0x40007800**		**NA**
I2C3		**NA**		**0x40007800**
DAC / DAC1		0x40007400		0x40007400
PWR		0x40007000		0x40007000
CRS		0x40006C00		0x40006C00
CAN		**0x40006400**		**NA**
USB/CAN SRAM		0x40006000		0x40006000
USB / USB FS		0x40005C00		0x40005C00
I2C2		0x40005800		0x40005800
I2C1		0x40005400		0x40005400
USART5		0x40005000		0x40005000
USART4		0x40004C00		0x40004C00
USART3 / LPUART1		**0x40004800**		**0x40004800**
USART2		0x40004400		0x40004400
SPI2		0x40003800		0x40003800
IWDG		0x40003000		0x40003000
WWDG		0x40002C00		0x40002C00
RTC + BKP_REG		0x40002800		0x40002800
LCD		**NA**		**0x40002400**
TIM14		**0x40002000**		**NA**
TIM7		**0x40001400**		**0x40000C00**
TIM6		0x40001000		0x40001000
TIM3		0x40000400		0x40000400
TIM2		0x40000000		0x40000000

Note: NA = Not applicable - Feature not available

Note: F0 devices have only one APB bus, APB1 and APB2 used in L0 devices indicate on which APB register the clock configuration bits of those peripherals are defined.

Note: Recent products can feature additional peripherals (such as expanded GPIO ports, GP timers and other digital or analog ones). User has to refer to the latest available documentation.

2.4 Reset and clock controller (RCC) interface

The main differences related to the RCC (Reset and clock controller) between the STM32F0 and the STM32L0 series are presented in *Table 6*.

Table 6. RCC differences between STM32F0 and STM32L0 series

RCC	STM32F0 series	STM32L0 series
MSI	NA	Multispeed internal RC clock (default system clock at startup)
HSI14	High speed internal oscillator dedicated to ADC	NA
HSI	8 MHz RC factory-trimmed	16 MHz RC factory-trimmed
HSI48	48 MHz RC clock (USB, PLL)	48 MHz RC clock (USB, RNG)
LSI	40 kHz RC	38 kHz RC
HSE	1 - 32 MHz	0 - 32 MHz 1 - 32 MHz with PLL or CSS
LSE	32.768 kHz (RTC)	32.768 kHz (RTC, LPTIMER, LCD, UARTs)
PLL	Main PLL	Main PLL
System clock source	HSI, HSE or PLL (opt. HSI48)	HSI, MSI, HSE or PLL
System clock frequency	Up to 48 MHz	Up to 32 / 16 / 4.2 MHz (depending on power range)
APB1,2/APB frequency	Up to 48 MHz	Up to 32 MHz (depending on power range)
RTC clock source	LSI, LSE or divided HSE	LSI, LSE or divided HSE
MCO clock source MCO pin: (PA8)	SYSCLK, HSI, HSE, HSI14, PLLCLK/2, LSE, LSI, (HSI48, PLL)	SYSCLK, HSI16, HSI48, MSI, HSE, PLL, LSI, LSE
Internal oscillator measurement / calibration	LSE / LSI / HSI / HSI14 / HSI48 / HSE clocks can be measured indirectly through MCO by the timer TIM14 input capture channel 1 with respect to HSI / HSE clock feeding the timer	LSE / LSI / MSI / HSE_RTC clocks can be measured by the timer TIM21 input capture channel 1 with respect to HSI / MSI / HSE clock feeding the timer
Firewall internal reset source	NA	Yes

Note: NA = Not applicable - Feature not available

When migrating between STM32F0 and STM32L0 series user should respect the possible system frequency limitations and check and update the setting if necessary:

a) System clock frequency while respecting limits related to applied voltage range

b) Flash parameters with respect to applied system frequency and voltage range

c) PLL configuration in case PLL is used as the system clock source (L0 has no implemented pre-divider on HSE, and programmable one on HSI PLL source input paths).

Both F0 and L0 series running at maximum performance (system clock at 48 MHz / 32 MHz) require Flash configured with 1 wait state.

User can use the clock configuration tool (STM32xxxx_Clock_Configuration.xls), to generate a customized system_stm32xxxx.c file containing a system clock configuration routine, depending on the application requirements.

Concerning the peripheral access configuration user has to check mapping of the peripherals' registers and adapt their control of reset and clock gating at proper AHBx and APBx registers at RCC (xxxxRSTR & xxxxENR) refer to *Table 5*. L0 series feature additional control of clock gating at sleep mode (xxxxSMENR).

When configuring peripheral clock sources to a dedicated clock source independent from the system clock, the source has to be checked during migration because of differences summarized in *Table 7*.

Table 7. Possible configuration of IPs' clock sources

Peripheral	STM32F0 series	STM32L0 series
ADC	HSI14, PCLK divided by 2 or by 4	HSI16 or HSI16/4 with prescaler, PCLK divided by 1, 2 or 4
USB	HSI48, PLL	HSI48, PLL VCO
USART	system clock, HSI8, LSE, APB	system clock, HSI16, LSE, APB
I2C	system clock, HSI8	system clock, HSI16, APB
RTC	LSE, LSI (40 kHz), HSE (divided by 32)	LSE, LSI (38 kHz), HSE_RTC (programmable prescaler)

2.5 DMA interface

Both STM32F0 and STM32L0 series feature up to 7-channel DMA controller. Some devices (e.g. STM32F09x) feature 5 additional independently configurable channels on DMA2. Each channel is dedicated to managing memory access requests from one or more peripherals.

The difference lies in optional remapping of DMA requests. While for F0 it is performed by specific bits at SYSCFG_CFGR1 configuration register, in L0 we have specific channel selection registers (CSELR).

Table 8 shows the basic differences between the DMA1 requests of peripherals among the STM32F0 and STM32L0 series.

Table 8. DMA request differences between STM32F0 and STM32L0 series

Peripheral	STM32F0 series	STM32L0 series
USART3 USART4 USART5 LPUART1	Ch2, Ch3, Ch6, Ch7 Ch6, Ch7 Ch1, Ch2, Ch3, Ch4, Ch5, Ch6, Ch7 Not available	Not available Ch2, Ch3, Ch6, Ch7 Ch2, Ch3, Ch6, Ch7 Ch2, Ch3, Ch6, Ch7
TIM1 TIM3 TIM6 TIM15 TIM16 TIM17	Ch2, Ch3, Ch4, Ch5, Ch7 Ch2, Ch3, Ch4, Ch6 Ch3 Ch5 Ch3, Ch4, Ch6 Ch1, Ch2, Ch7	Not available Ch2, Ch3, Ch5, Ch6 Ch2 Not available Not available Not available
DAC	Ch3, Ch4	Ch2, Ch4
AES	Not available	Ch1, Ch2, Ch3, Ch5

Note: Recent products can feature additional DMA channels corresponding to the added peripherals. User has to refer to the latest available documentation.

2.6 Interrupt vectors

Table 9 summarizes the differences of the interrupt vector table between the STM32F0 and STM32L0 series.

Table 9. Interrupt vector differences between STM32F0 and STM32L0 series

Position [Priority]	STM32F0 series	STM32L0 series
- [0]	Reserved	Memory management
- [1]	Reserved	Pre-fetch fault, memory access fault
- [2]	Reserved	Undefined instruction or illegal state
- [4]	Reserved	Debug Monitor
1 [8]	PVD and V_{DDIO2} supply comparator EXTI lines 16 and 31	PVD comparator through EXTI Line
3 [10]	Flash	Flash and EEPROM
10 [17]	DMA1_CH2_3, DMA2_CH1_2	DMA1_CH2_3
11 [18]	DMA1_CH4_5_6_7, DMA2_CH3_4_5	DMA1_CH4_5_6_7
13 [20]	TIM1 break, update, trigger and commutation	LPTIM1 through EXTI29
14 [21]	TIM1 capture compare	Reserved
19 [26]	TIM14	Reserved
20 [27]	TIM15	TIM21
21 [28]	TIM16	I2C3

Table 9. Interrupt vector differences between STM32F0 and STM32L0 series (continued)

Position [Priority]	STM32F0 series	STM32L0 series
22 [29]	TIM17	TIM22
29 [36]	USART3 and USART4	LPUART through EXTI28, AES and RNG
30 [37]	CEC, CAN, EXTI27	LCD

Note: Recent products can feature additional interrupt vectors corresponding to the added peripherals. User has to refer to the latest available documentation.

2.7 GPIO interface

The STM32F0 GPIO peripheral is mapped on AHB bus while the STM32L0 features a new I/O interface which allows single cycle accesses and so faster operations on I/O ports.

GPIO control is the same but alternate function mapping differs completely. User has to analyze related datasheet to check the mapping at the dedicated AFR registers. STM32F0 features configuration locking registers for port A and B only while STM32L0 has embedded locking registers for all the available ports.

2.8 EXTI source selection

Both STM32F0 and STM32L0 series handle external (configurable) and internal (direct) interrupt/event lines. Peripherals with wakeup from Stop capability are connected to dedicated EXTI configurable linen in both series. In this case the EXTI configuration is required to allow the wakeup from Stop mode. *Table 10* summarizes connection differences.

Table 10. EXTI lines connection differences

EXTI line	STM32F0 series	STM32L0 series
19	RTC tamper or timestamp	RTC tamper, timestamp or CSS_LSE
24	Reserved	I2C3
27	CEC	Reserved
28	USART3	LPUART1
29	Reserved	LPTIM1
31	VDDIO2 comparator	Reserved

2.9 Flash interface

Table 11 summarizes the difference between the Flash interface of STM32F0 and STM32L0 series, both differ in prefetch management and memory access control.

The interface is not compatible and control routines have to be rewritten. In addition, the STM32L0 features EEPROM data memory, proprietary code read out protection and a quite different option byte organization.

Timing and supply current for all programming phases can differ significantly between the two series, endurance and retention parameters may vary too. User has to carefully check values specified in datasheets to identify the differences.

Table 11. Flash interface differences between STM32F0 and STM32L0 series

Feature		STM32F0 series	STM32L0 series
Main/Program memory	Start Address	0x0800 0000	0x0800 0000
	End Address	Up to 0x0803FFF for 256 KBytes devices	Up to 0x0802FFF for 192 KBytes devices
	Granularity	64 pages of 1 KByte (64KB), 4 pages per sector (16x), or 64 pages of 2 KBytes (128KB), 2 pages per sector (32x), or 128 pages of 2 KBytes (256KB), 2 pages per sector (31x) + 66 pages in the 32nd sector	512 pages of 32 words (64KB) at 16 sectors of 4KB (dual bank system is used, 2x 64K for 128 KBytes and 2x 96K for 192 KBytes devices with separated charge pumps, RWW and swap capability)
EEPROM memory	Granularity	Only available through SW emulation	Up to 1536 pages of single words Up to 6K of data memory for 192 KBytes devices
	Start Address	Not available	0x0808 0000
	End Address	Not available	Up to 0x0808 17FF for 192 KBytes device
System memory	Start Address	0x1FFF EC00 on STM32F03/5x 0x1FFF C400 on STM32F04x 0x1FFF C800 on STM32F07x 0x1FFF D800 on STM32F09x	0x1FF0 0000
	End Address	0x1FFF F7FF	0x1FF0 1FFF
Option Bytes	Start Address	0x1FFF F800	0x1FFF F800
	End Address	0x1FFF F80F	0x1FF8 001F
Flash interface	Start address	0x4002 2000	0x4002 2000
	Programming and erasing procedures and granularities	Flash – half word (16 bits) programming – page (1 or 2 KByte) or mass erase Option bytes – half word (16 bits) format, – half word programming and erase	Flash – word or half page (16 words) programming – parallel half page programming - 32 words - is available in dual bank systems – page (32 words) or mass erase data EEPROM – byte, half word and word programming – word erase Option bytes – word format – word programming and erase

Table 11. Flash interface differences between STM32F0 and STM32L0 series (continued)

Feature		STM32F0 series	STM32L0 series
Read Protection	Unprotection	Level 0 no protection RDP = 0xAA	Level 0 no protection RDP = 0xAA
	Protection	Level 1 memory protection RDP != (Level 2 & Level 0) Level 2: Lvl 1 + Debug disabled, RDP = 0xCC	Level 1 memory protection RDP != (Level 2 & Level 0) Level 2: Lvl 1 + Debug disabled, RDP = 0xCC
Proprietary Code Read-Out Protection		Not available	Sector granularity (4KB)
Write protection		Protection by 4-Kbyte block	Protection by 4-Kbyte block

2.10 Embedded SRAM

Basic difference between volatile memories is parity check supported by STM32F0 family while L0 features by firewall protection.

The latest STM32F0 products feature memory of up to 32 KBytes with HW parity while STM32L0 ones go up to 20 KBytes.

2.11 System of timers

Both STM32F0 and STM32L0 feature a set of general purpose timers with some small differences, summarized in *Table 12*. In addition STM32F0 embeds an advanced control timer. On the other side STM32L0 features a single 16-bit low power timer.

Table 12. Timers' system differences between STM32F0 and STM32L0 series

Timer feature	STM32F0 series	STM32L0 series
Number of timers	1x 16-bit advanced control up to 5x 16-bit general purpose 2x 16-bit basic	4x 16-bit general purpose 2x 16-bit basic 1x 16-bit low power
Advanced control	TIM1 – complementary outputs – programmable dead-time – break input	Not available
General purpose	TIM15 / TIM16 / TIM17 – complementary PWM – dead-time insertion, break input – PWM edge aligned only	TIM21 / TIM22 – PWM edge & center aligned – ext. trigger – encoder interface
Synchronization	TIM15 only	TIM21 / TIM22

2.12 ADC interface

Table 13 presents the differences between the ADC interface of STM32F0 and STM32L0 series; these differences are the following:

- power supply range
- faster conversion
- new features.

Table 13. ADC differences between STM32F0 and STM32L0 series

ADC	STM32F0 series	STM32L0 series
Maximum sampling frequency	1 MSPS @ HSI14 (1000 ns @ 12-bit resolution)	1,14 MSPS @ HSI16 (875 ns @ 12-bit resolution)
Supply range	2.4 to 3.6 V	1.8 to 3.6 V (for some product even from 1.65 V)
Internal monitoring	TRGx, VSENSE (IN16), VREFINT (IN17), VBAT (IN18)	TRGx, VSENSE (IN18), VREFINT (IN17), LCD_VLCD1 (IN16)
External triggers	TIM1, TIM2, TIM3 and TIM15	TIM2, TIM3, TIM6, TIM21, TIM22 or EXTI11
Oversampling	Not available	Ratio from 2 to 256x
Voltage regulator	Not available	Yes
Asynchronous clock prescaler	Not available	16 levels from 1 to 256
Low frequency mode	Not available	Yes
EOCAL interrupt	Not available	Yes

2.13 DAC interface

The STM32L0 series embeds an enhanced DAC versus some members of the F0 series. The basic differences are summarized in *Table 14*.

Table 14. DAC differences between STM32F0 and STM32L0 series

DAC feature	STM32F0 series	STM32L0 series
Number of channels	1x (2x F07x and F09x only)	1x (2x L07 only)
Noise & triangular wave generation	F07x and F09x only	Yes
HW trigger	TIM2, TIM3, TIM6, TIM7, TIM15 and EXTI	TIM2, TIM3, TIM6, TIM7, TIM21 and EXTI
Supply range	2.4 to 3.6 V	1.8 to 3.6 V
Simultaneous conversion	Dual mode (F07x and F09x only)	L07 only

2.14 COMP interface

Both the STM32F0 and STM32L0 series embed an enhanced ultra low power comparator.

User has to consider differences at input and output mapping. Additional differences are summarized in *Table 15*.

Table 15. COMP differences between STM32F0 and STM32L0 series

COMP feature	STM32F0 series	STM32L0 series
Number	Up to 2x rail to rail	1x rail to rail + 1x ultra-low power
Outputs	PWM emergency shut down (BKIN) Cycle by cycle current control loop (OCREF_CLR events) Input capture of timers	Input capture & ETR of timers
Programmable hysteresis	Yes	No
Programmable speed / power mode	4 levels	2 levels

2.15 PWR interface

The PWR controller of STM32F0 series has several differences vs. L0 series, summarized in the *Table 16*. Note that the STM32F0x8 subfamily doesn't feature voltage regulator to supply internal 1.8 V to the digital power domain, a stable voltage has to be delivered by the application.

Table 16. PWR differences between STM32F0 and STM32L0 series

PWR	STM32F0x1/2 series	STM32F0x8 series	STM32L0 series
Power supplies	– V_{DD} = 2.0 to 3.6 V: external power supply for I/Os and the internal regulator. Provided externally through VDD pins. – V_{SSA}, V_{DDA} = 2.0 to 3.6 V: external analog power supplies for ADC, DAC, Reset blocks, RCs and PLL. – V_{BAT} = 1.65 to 3.6 V: power supply for RTC, external clock 32 kHz oscillator and backup registers (through power switch) when V_{DD} is not present. – VDDIO2 = 1.65 to 3.6 V independent IO supply rail powering USB as well	– V_{DD} = 1.8 V +/- 8%: external power supply for I/Os. Provided externally through VDD pins. – V_{SSA}, V_{DDA} = 1.65 to 3.6 V: external analog power supplies for ADC, DAC, Reset blocks, RCs and PLL. – V_{BAT} = 1.65 to 3.6 V: power supply for RTC, external clock 32 kHz oscillator and backup registers (through power switch) when V_{DD} is not present. – VDDIO2 = 1.65 to 3.6 V independent I/O supply rail powering USB as well	– V_{DD} = 1.8 to 3.6 V (with BOR - else from 1.65 V): external power supply for I/Os and the internal regulator. Provided externally through VDD pins. – V_{SSA}, V_{DDA} = 1.8 to 3.6 V (with BOR - else from 1.65 V): external analog power supplies for ADC, DAC, reset blocks, RC oscillators and PLL. – External VREF+ available on TFBGA64 and 100 pin packages only (otherwise bonded to VDDA) – VLCD 2.5 - 3.6 V external power supply (else internal embedded step-up converter) – VDD_USB 3.0 - 3.6 V
Voltage regulator	Feeds 1.8 V domain: – Run mode: Core, memories, digital parts – Stop mode: Preserving internal RAM & registers content – Standby mode: switched off	Not available (regulator bypassed)	The regulator output voltage (VCORE) can be programmed by software to three different ranges within 1.2 - 1.8 V (dynamic voltage scaling) – Run mode: Core, memories, digital parts – LPR mode: VCORE domain, preserving the contents of the registers and internal SRAM – Stop mode: Preserving internal RAM & registers content – Standby mode: switched off
Battery backup domain	– Backup registers – RTC – LSE – RCC Backup Domain Control Register	– Backup registers – RTC – LSE – RCC Backup Domain Control Register	Not available - RTC contains 5 backup data registers (20 bytes)
Power supply supervisor	Integrated POR / PDR circuitry Programmable voltage detector (PVD) of VDD	Controlled externally through the dedicated NPOR pin.	Integrated POR/ PDR - BOR circuitry Programmable voltage detector (PVD) of VDD & VDDA

Table 16. PWR differences between STM32F0 and STM32L0 series (continued)

PWR	STM32F0x1/2 series	STM32F0x8 series	STM32L0 series
Low-power modes	Sleep Stop Standby (1.8 V domain powered-off)	Sleep Stop	Low power run Sleep Low power sleep Stop Standby
Wake-up sources	Sleep – Any peripheral interrupt/wakeup event Stop – Any EXTI line event/interrupt Standby – WKUP0 or WKUP1 pin rising edge – RTC alarm – External reset in NRST pin – IWDG reset	Sleep – Any peripheral interrupt/wakeup event Stop – Any EXTI line event/interrupt	Low power run – switch MR on Sleep – Any wakeup or interrupt Low power sleep – Any wakeup or interrupt Stop – Any EXTI line event/interrupt Standby – WKUP pin rising edge – RTC alarm – RTC wake up – External reset in NRST pin – IWDG reset

2.16 Real-time clock (RTC) interface

The STM32L0 series embeds an improved RTC peripheral versus the F0 series, however the architecture, features and programming interface are similar.

The L0 RTC provides additionally:

- A sub-second programmable alarm
- 16-bit wakeup auto reload timer with interrupt capability (STM32F0x7 devices features by wakeup timer only)
- RTC output remapping
- Low power timer triggering by tamper event

For advanced information about the RTC programming, please refer to Application Note AN3371 “*Using the STM32 HW real-time clock (RTC)*”.

2.17 SPI interface

The STM32F0 series embeds a new SPI peripheral versus the L0 series. The architecture, features and programming interface are modified to introduce new capabilities.

As a consequence, the F0 SPI programming procedures and registers are a bit different to those of the L0 series. The code written for the F0 series using the SPI needs little rework to run on L0 series and oppositely.

The basic differences are summarized in *Table 17*:

Table 17. SPI differences between STM32F0 and STM32L0 series

SPI feature	STM32F0	STM32L0
Max speed	18 MBits/s	16 MBits/s
Data size	Programmable 4-16 bits	Fixed 8 or 16-bit
Data buffers	32-bit Tx & Rx FIFOs (up to 4 data frames)	16-bit Tx & Rx buffers (single data frame)
Data packing	Data packing (8 & 16 & 32-bit access, programmable FIFOs data threshold)	Not available
Instances	1-2x SPI (inc. 1-2x I2S)	2x SPI (inc. 1x I2S)

2.18 I2C interface

Both STM32F0 and STM32L0 series embed compatible I2C peripherals. Although the L0 features an enhanced version correcting some minor bugs of the older one used on F0, the programming procedures and registers are the same. Note that STM32L07x devices feature up to 3 instances of the I2C.

2.19 USART interface

The STM32L0 series embeds an enhanced USART compared with the F0 series, but the devices are compatible from the user point of view, as the programming procedures and registers remain the same.

Maximum speed is 6 MBits/s for STM32F0, 4 MBits/s only for STM32L0 devices. Number of instances is product dependent.

The main difference is implementing low power UART baud rate generator on STM32F0. Not all the applied instances support all the available features. Refer to reference manuals to check the range and supported communication.

The STM32L0 series features low power asynchronous UART supporting clocking from LSE (up to 9.6 KBd) and communication at stop mode. The features supported by this peripheral are limited vs. standard USARTs.

2.20 TSC interface

Both STM32F0 and STM32L0 series embed the same TSC interface, the only difference is the pinout concerning specific pins. While STM32F0 uses ports PA, PB, PD and PE, the STM32L0 uses PA, PB and PC. Configuration control and registers are the same.

2.21 Debug module

User can expect differences in mapping of bits at peripheral freezing registers and debug sleep mode supported in STM32L0 devices.

3 Firmware migration using the library

Both STM32F0 and STM32L0 members are supported by new HAL (hardware abstraction layer) firmware interface. It collects universal drivers working mostly upon predefined structures of configuration data. This should ensure portable APIs across all the STM32 series in general for what concerns functional operations and control of given peripherals.

However, considering the IPs specificities and implementation differences between these two subfamilies, some differences may appear mainly in the peripherals initialization, definition of the configuration structures and range of valid enumerators to be applied into them. Some functions could have different number of parameters.

There should be however no differences when handling any standard flow except support of newly added features. User can find more detailed information in dedicated manuals describing HAL firmware and migration between supported devices.

The basic idea here is to provide a comparison of header files of dedicated IP drivers referencing all the used data structures, enumerators, exported functions and parameters.

These differences can help to detect parts of software to be modified when migrating between the FW interfaces.

Table 18 summarizes basic differences concerning functions exported by drivers.

Table 18. HAL drivers' function difference summary

Driver	STM32F0	STM32L0
stm32xxxx_hal	-	HAL_EnableDBGSleepMode HAL_DisableDBGSleepMode HAL_DBG_LowPowerConfig HAL_VREFINT_Cmd HAL_VREFINT_OutputSelect HAL_ADC_EnableBuffer_Cmd HAL_ADC_EnableBufferSensor_Cmd HAL_COMP_EnableBuffer_Cmd HAL_RC48_EnableBuffer_Cmd
stm32xxxx_hal_adc_ex	-	HAL_ADCEx_Calibration_GetValue HAL_ADCEx_Calibration_SetValue
stm32xxxx_hal_dac_ex	HAL_DACEx_DualGetValue HAL_DACEx_DualSetValue HAL_DACEx_ConvCpltCallbackCh2 HAL_DACEx_ConvHalfCpltCallbackCh2 HAL_DACEx_ErrorCallbackCh2 HAL_DACEx_DMAUnderrunCallbackCh2 DAC_DMAConvCpltCh2 DAC_DMAErrorCh2 DAC_DMAHalfConvCpltCh2	-
stm32xxxx_hal_pwr_ex	-	HAL_PWREx_EnableFastWakeUp HAL_PWREx_DisableFastWakeUp HAL_PWREx_EnableUltraLowPower HAL_PWREx_DisableUltraLowPower HAL_PWREx_EnableLowPowerRunMode HAL_PWREx_DisableLowPowerRunMode

Table 18. HAL drivers' function difference summary (continued)

Driver	STM32F0	STM32L0
stm32xxxx_hal_rcc	-	HAL_RCC_GetPCLK2Freq
stm32xxxx_hal_rtc_ex	HAL_RTCEx_Tamper3EventCallback HAL_RTCEx_PollForTamper3Event	HAL_RTCEx_AlarmBEventCallback
stm32xxxx_hal_spi	HAL_SPI_InitExtended	-
stm32xxxx_hal_tim_ex	HAL_TIMEx_OCN_Start HAL_TIMEx_OCN_Stop HAL_TIMEx_OCN_Start_IT HAL_TIMEx_OCN_Stop_IT HAL_TIMEx_OCN_Start_DMA HAL_TIMEx_OCN_Stop_DMA HAL_TIMEx_PWMN_Start HAL_TIMEx_PWMN_Stop HAL_TIMEx_PWMN_Start_IT HAL_TIMEx_PWMN_Stop_IT HAL_TIMEx_PWMN_Start_DMA HAL_TIMEx_PWMN_Stop_DMA HAL_TIMEx_OnePulseN_Start HAL_TIMEx_OnePulseN_Stop HAL_TIMEx_OnePulseN_Start_IT HAL_TIMEx_OnePulseN_Stop_IT	-

4 Revision history

Table 19. Document revision history

Date	Revision	Changes
10-Feb-2015	1	Initial release.