AN0052
Application note
Rev 1.00 2021/11/3

Migrating from MM32F0130 to MM32F0140

Introduction

This application note describes and analyzes the differences between MM32F0130 and MM32F0140, and provides steps required to migrate from the exsiting MM32F0130 device to MM32F0140 device. Hardware difference, peripheral migration and firmware migration are introduced respectively.

Table 1 Applicable series and models

Series	Models
MM32F0130	MM32F0131C3N
	MM32F0131C4N
	MM32F0131C4P
	MM32F0131C6P
	MM32F0131C7P
	MM32F0132C4N
	MM32F0132C4P
	MM32F0132C6P
	MM32F0132C7P
	MM32F0133C4N
	MM32F0133C4P
	MM32F0133C6P
	MM32F0133C7P
MM32F0140	MM32F0141B1T
	MM32F0141B4P

	MM32F0141B4Q
	MM32F0141B6P
	MM32F0144C1T
	MM32F0144C4P
	MM32F0144C4Q
	MM32F0144C6P

1 Differences and similarities between MM32F0130 and MM32F0140

1.1 Comparisons of the two series

MM32F0140 MCU is largely compatible with MM32F0130 MCU in functions of the same package products. On the whole, these two series have their unqiue functions.

Item	MM32F0130	MM32F0140	Descriptions
Core	Cortex-M0	Cortex-M0	Have the same core
FLASH	64KB	32KB / 64KB	Flash start addresses and operation controllers are the same; but differences can be seen in read-protect setting, since MM32F0140 adds 32KB Flash;
SRAM	16KB	8KB	RAM start addresses are the same, and the Size of MM32F0140 is only 8K.
Maximum CPU frequency	72MHz	72MHz	MM32F0130 and MM32F0140 supports PLL multiplier, which could be up to 72MHz, but differences can be seen in their egister position address for configuring PLL.
Operating voltage	2.0V to 5.5V	2.0V to 5.5V	Same
GPIO	Not support Tolerant	Not support Tolerant	Same
Boot	Only support UART1 boot and it can be multiplexed to	Only support UART1 boot and it can be multiplexed to more pins.	The multiplex pins are the same in the same packages.

		more pins.		
Timer	Advanced	1	1	Same
	General-purpose (4 channels)	2, one is 32-bit and the other is 16-bit.	2, one is 32-bit and the other is 16-bit.	Same
	General-purpose (2 channels)	2	2	Same
	Basic	1	1	Same
	Systick	1	1	Same
ADC		1 x 12bit	1 x 12bit	Both support the arbitrary channel configuration. MM32F0140 adds interjected sampling.
COMP		2	1	MM32F0130 supports two groups of COMP，while MM32F0140 only supports one group of COMP and adds output mode selection.
RTC		1	0	MM32F0130 supports RTC function and has no VBAT Pin while MM32F0140 does not support RTC function.
USB Device		1	0	MM32F0140 does not support USB Device
CAN		1	1	MM32F0140 supports new FlexCAN while MM32F0130 supports classic CAN.
UART		2	3	MM32F0140 supports three UARTs and IrDA is added to the

			UART function.
SPI	2	2	MM32F0130 and MM32F0140 support two SPIs. MM32F0140 supports I2S, while MM32F0130's SPI does not support I2S.
I2C	1	1	MM32F0140 supports I2C slave multi-address function.

1.2 Similarities and differences among functional pins

MM32F0140 series MCU is compatible with GPIO pin and power supply pin (take LQFP48 of the same package as an example) of MM32F0130 series. Boot0 pin of MM32F0140 has an additional PD5, giving MM32F0140 one more GPIO than MM32F0130. MM32F0140's GPIO can multiplex more peripheral PIN combinations than MM32F0130. The former will be more dynamic than the latter. For more details, please refer to the multiplex function tables of MM32F0140 data sheet. Introductions, together with function descriptions will be given in the following sections.

	MM32F0130					MM32F0140			
Package	QFP64	QFP48	QFN32	QFP32	Pinout	QFP48	QFN32	QFP32	Pinout
Number of GPIO	56	39	27	25		40	28	26	MM32F0140 has one more GPIO and PD5 than MM32F0130.
Boot0 Pin	Pin 56	Pin 44	Pin31	Pin31	Boot0	Pin 44	Pin31	Pin31	PD5/Boot0

MM32F0140 MCU is compatible with the power supply pin of MM32F0130.

Take the power supply pin of LQFP48 package chip as an example, and the results are shown below:

LQFP48	MM32F0130	MM32F0140
PIN8	VSSA	VSSA
PIN23	VSS	VSS
PIN47	VSS	VSS
PIN9	VDDA	VDDA
PIN24	VDD	VDD
PIN48	VDD	VDD

MM32F0140 MCU is compatible with the power supply pin of MM32F0130.

LQFP48	MM32F0130	MM32F0140
WKUP	PA0	PA0
BOOT0	PIN44@LQFP48	PIN44@LQFP48, PD5
NRST	PIN7@LQFP48	PIN7@LQFP48

MM32F0140 MCU is compatible with WKUP, Boot0 and NRST pins of MM32F0130.

1.3 Differences in UART pins

The UART pins multiplexed by MM32F0140 and MM32F0130 of the same package chip are not compatible. MM32f0140 supports UART3.

Pin functions	MM32F0130	MM32F0140
UART1_TX	PD0	
	PA1	
	PA9	PA9
	PA10	PA10
	PA13	PA13
		PB3
	PB6	PB6
	PB9	
UART1_RX	PD1	
	PA0	
	PA9	PA9
	PA10	PA10
	PA14	PA14
		PB4
	PB7	PB7
	PB8	
UART1_RTS	PA12	PA12
UART1_CTS	PA11	PA11
UART2_TX	PA2	PA2
	PA14	PA14
		PB7
		PD4
UART2_RX	PA3	PA3
		PA6
	PA15	PA15
		PB8
UART2_RTS	PA1	PA1
UART2_CTS	PA0	PA0
UART3_TX	-	PD0
	-	PB10
	-	PA11
UART3_RX	-	PD1
	-	PB11

Pin functions	MM32F0130	MM32F0140
	-	PA12
UART3_RTS	-	PB14
UART3_CTS	-	PB13

1.4 Differences in SPI pins

The SPI pins multiplexed by MM32F0140 and MM32F0130 of the same package chip are not wholly compatible.

Pin functions	MM32F0130	MM32F0140
SPI1_NSS	PA4	PA4
	PA5	
	PD2	
		PA14
	PA15	PA15
SPI1_SCK	PD1	
	PD3	
	PA4	
	PA5	PA5
	PB3	PB3
SPI1_MOSI	PA7	PA7
	PB5	PB5
	PD0	
SPI1_MISO	PA6	PA6
	PB4	PB4
	PD1	
	PD3	
		PD6
SPI2_NSS		PD4
		PA0
		PA2
	PB12	PB12
	PB13	PB13
	PB14	PB14
	PB15	PB15
	PB9	PB9
SPI2_SCK	PB10	PB10
	PB12	PB12
	PB13	PB13
	PB14	PB14
	PB15	PB15
		PA10
	PA15	
SPI2_MOSI	PB12	PB12
	PB13	PB13
	PB14	PB14
	PB15	PB15

		PA11
	PA15	
SPI2_MISO	PB12	PB12
	PB13	PB13
	PB14	PB14
	PB15	PB15
		PA12
		PA13
	PA15	

In addition, MM32F0140 also supports I2S. For the specific multiplex pin, please refer to I2S multiplex function of MM32F0140 data sheet.

1.5 Differences in I2C pins

The I2C pin multiplexed by MM32F0140 is not wholly compatible with the one multiplexed by MM32F0130.

Pin functions	MM32F0130	MM32F0140
I2C1_SCL	PD1	PD1
	PD2	
		PA5
	PB10	PB10
	PB13	PB13
	PA9	PA9
	PA11	PA11
	PB6	PB6
	PB8	PB8
I2C1_SDA	PD0	PD0
	PD3	
		PA4
	PB11	PB11
	PB14	PB14
	PA10	PA10
	PA12	PA12
	PB7	PB7
	PB9	PB9

1.6 Differences in CAN pins

MM32F0130 supports classic CAN function while MM32F0140 achieves CAN2.08 function with the use of new FlexCAN. The following is the comparison of the relevant multiplex pins.

Pin fucntions	MM32F0130	MM32F0140
CAN1_TX		PA10
	PA12	PA12
	PB9	PB9
CAN1_RX		PA9
	PA11	PA11
	PB8	PB8

1.7 Differences in MCO pins

The MCO pin multiplexed by MM32F0140 covers and is compatible with the one multiplexed by MM32F0130.

Pin functions	MM32F0130	MM32F0140
MCO		PB1
	PA8	PA8
	PA9	PA9
		PA13
		PB5

1.8 Differences in SWD pins

The SWD pin multiplexed by MM32F0140 is wholly compatible with the one multiplexed by MM32F0130.

Pin functions	MM32F0130	MM32F0140
SWDIO	PA13	PA13
SWDCLK	PA14	PA14

1.9 Differences in OSC pins

The OSC pin multiplexed by MM32F0140 is wholly compatible with the one multiplexed by MM32F0130.

Pin functions	MM32F0130	MM32F0140
OSC_IN	PD0	PD0
OSC_OUT	PD1	PD1

1.10 Differences in ADC pins

The ADC pin multiplexed by MM32F0140 covers and is compatible with the one multiplexed by MM32F0130. The Vref and Temperature Sensor channels are mutually compatible.

Pin functions	MM32F0130	MM32F0140

ADC1_VIN[0]	PA0	PA0
ADC1_VIN[1]	PA1	PA1
ADC1_VIN[2]	PA2	PA2
ADC1_VIN[3]	PA3	PA3
ADC1_VIN[4]	PA4	PA4
ADC1_VIN[5]	PA5	PA5
ADC1_VIN[6]	PA6	PA6
ADC1_VIN[7]	PA7	PA7
ADC1_VIN[8]	PB0	PB0
ADC1_VIN[9]	PB1	PB1
ADC1_VIN[10]		PB3
ADC1_VIN[11]		PB4
ADC1_VIN[12]		PB7
ADC1_VIN[13]		PB6

1.11 Differences in COMP pin

MM32F0130 has one more group of COMP than MM32F0140. Significant differences can be seen in the COMP pin multiplexed by MM32F0140 and the one multiplexed by MM32F0130.

Pin functions	MM32F0130	MM32F0140
COMP1_OUT	PA0	PA0
COMP1_OUT	PA6	PA6
COMP1_OUT	PA11	PA11
COMP1_INP[0]	PA0	PA1
COMP1_INP[1]	PA1	PA2
COMP1_INP[2]	PA2	PA3
COMP1_INP[3]	PA3	PA4
COMP1_INM[0]	PA4	PA5
COMP1_INM[1]	PA5	PA6
COMP1_INM[2]	PA0	PA7
COMP2_OUT	PA2	
COMP2_OUT	PA7	
COMP2_OUT	PA12	
COMP2_INP[0]	PA0	
COMP2_INP[1]	PA1	
COMP2_INP[2]	PA2	
COMP2_INP[3]	PA3	
COMP2_INM[0]	PA4	
COMP2_INM[1]	PA5	
COMP2_INM[2]	PA2	

2 Peripheral migration

2.1 Peripheral differences and compatibility

The MM32F0140 core and most of the peripherals are compatible with MM32F0130, on the basis of which the classic CAN is replaced with FlexCAN function. Moreover, MM32F0140 does not support USB Device function and RTC function.

These two series have the same peripherals, register base address, register function, register offset address, and register bits. No code changes are required during migration. The same function can be maintained at the application level, while the main features and behaviors of the peripherals remain unchanged.

For peripherals with minor enhancements, the register base address, register function, register offset address, and register bits are the same for the same part of the peripheral. This part of the code does not need to be changed during migration, and the same function is maintained at the application level. Only when a new function is needed, the new control and status bits are operated by calling a new function.

For some peripherals of the same type, their functions have changed significantly, with the use of new IP, new architecture, and new features. During migration, a top-to-bottom replacement starts from the application level. In the current sample, the underlying HAL tries to achieve the same operation via the same function name. However, the parameters will vary with the registers.

2.2 Reset and clock controller (RCC)

The main difference between MM32F0140 and MM32F0130 in RCC (reset and clock controller) is that MM32F0140 modifies the control part of PLL and the related PLL controller is in the PLLCFGR register, while the related PLL controller of MM32F0130 is in CFCG register.

Besides, the number of clock control bits of relevant IPs are increased or decreased accordingly.

Newly added registers	Register descriptions	Offset	MM32F0140 vs MM32F0130
BDCR	Backup Domain Control Register	0x20	Remove
ICSCR	Internal clock source calibration register	0x4C	Add
PLLCFGR	PLL Configuration Register	0x50	Add
PLLDLY	PLL delay Register	0x88	Add

RCC	MM32F0130	MM32F0140
HSI	Factory-calibrated 48 MHz RC oscillator	Factory-calibrated 8 MHz RC oscillator with improved accuracy.
HSE	2 - 24 MHz	4 -24 MHz（Crystal）
PLL	PLL（relevant registers of PLL are in CFGR）	PLL（relevant registers of PLL are in PLLCFGR）
LSI	40 KHz RC	Same
System clock source	HIS, HSI/6, HSE, LSI or PLL	HSI（DIV 1,2,4,8 … 128）, HSE or PLL, LSI
System clock frequency	Use HSI/6 after reset and the RC can be set up to 72MHz by PLL, if HSI/4 is used as PLL.	Use the built-in RC 8MHz after reset and the RC can be set up to 72MHz via PLL
APB1/APB freqquency	Up to 72 MHz	Same
MCO clock source	SYSCLK、HSI/4、HSE、LSI、PLL/2	SYSCLK、HSI、HSE、LSI、PLL/2
MCO pin:	PA8, PA9	Same

2.3 Interrupt vectors

MM32F0140 and MM32F0130 are mutually compatible in interrupt vector and their vector addresses are the same. The the main difference is that MM32F0140 adds the corresponding interrupt vectors of UART3 and FlexCAN, while reduces the corresponding interrupts of classic CAN and USB Device.

Interrupts	MM32F0130	MM32F0140
UART3	NA	UART3_IRQHandler
CAN	Legacy CAN	NA
FLEX_CAN	NA	FLEX_CAN_IRQHandler
USB	Support	NA

2.4 GPIO

MM32F0140 and MM32F0130 of the same package chip, are mutually compatible in GPIO and their registers and register bit functions are the same.

2.5 EXTI

MM32F0140 and MM32F0130 are mutually compatible in EXTI and their registers and register bit functions are the same. The main difference is that MM32F0140 reduces the following connections:

EXTI line 17 is connected to RTC clock event

EXTI line 18 is connected to USB bus suspend interrupt

EXTI line 20 is connected to comparator 2 for output

MM32F0130	MM32F0140
EXTI line 16 is connected to PVD output EXTI line 17 is connected to RTC clock event EXTI line 18 is connected to USB bus suspend interrupt EXTI line 19 is connected to comparator 1 for output EXTI line 20 is connected to comparator 2 for output EXTI line 24 is connected to IWDG interrupt	EXTI line 16 is connected to PVD output EXTI line 19 is connected to comparator 1 for output EXTI line 24 is connected to IWDG

2.6 DMA

MM32F0140 and MM32F0130 are mutually compatible in DMA and their registers and register bit functions are the same. Changes can be seen in the mapping function of the corresponding peripherals.

2.7 Flash

MM32F0140 and MM32F0130 are mutually compatible in Flash and their registers and register bit functions are the same. The main difference is the way in which read-protect is set.

The way for MM32F0140 to set read-protect is different from that of MM32F0130. Most operations performed in Flash are the same, but the setting position and value of read-protect are different.

The way for MM32F0130 to set read-protect:

Write RDP2, RDP3 half-words to the corresponding addresses in sequence according to the half-word programming operations in the option byte area:

Write target value 0x7F80 to 0x1FFE0000；

Write target value 0xFF00 to 0x1FFE0002；

Read-protect is set based on the fact that the software resets the target system or the software is powered down, then powered up.

The way for MM32F0140 to set read-protect:

Write RDP half-word to the corresponding address according to the half-word programming operations in the option byte area.

Set FLASH AR address value as 0x1FFFF800 and perform block erase;

Write target value 0x7F80 to 0x1FFFF800；

The software resets the target system or the software is powered down, then powered up to set read-protect.

For the specific implementation way, please refer to the following codes.

LibSamples\FLASH\FLASH_SetReadProtect\HARDWARE\FLASH\flash.c

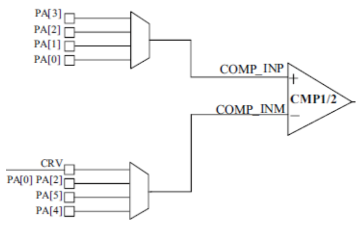
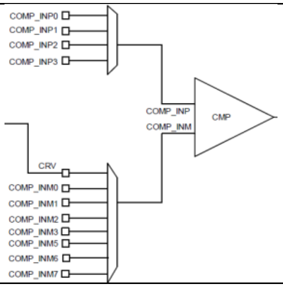
2.8 ADC

MM32F0140 and MM32F0130 are mutually compatible in ADC and their basic control registers and register bit functions are the same. The main difference is that MM32F0140 adds the interjected sampling function and related registers.

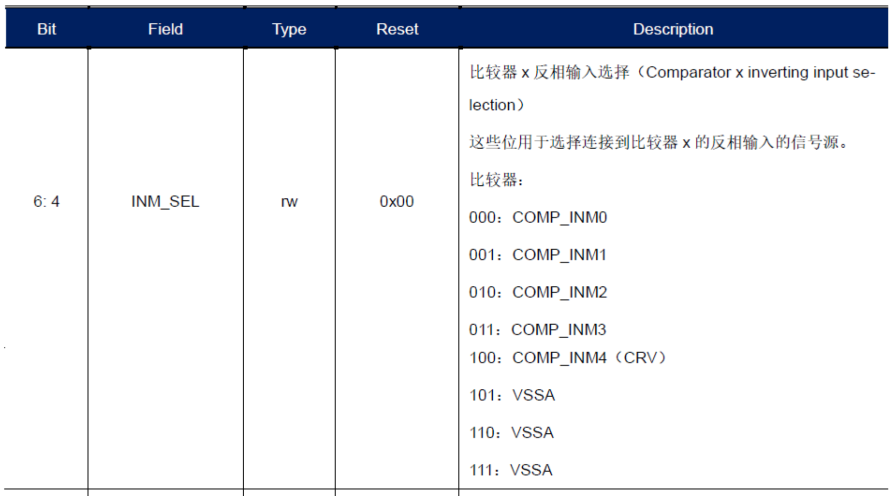
Newly added registers	Register descriptions	Offset	Descriptions
JOFR0	Injectionchanneldatacompensationregister0	0x7C	The newly added injected register
JOFR1	Injectionchanneldatacompensationregister1	0x80	The newly added injected register
JOFR2	Injectionchanneldatacompensationregister2	0x84	The newly added injected register
JOFR3	Injectionchanneldatacompensationregister3	0x88	The newly added injected register
JSQR	Injectionsequenceregister	0x8C	The newly added injected register
JDATA	Injectdataregister	0x90	The newly added injected register
JDR0	Injectionchanneldataregister0	0xB0	The newly added injected register
JDR1	Injectionchanneldataregister1	0xB4	The newly added injected register
JDR2	Injectionchanneldataregister2	0xB8	The newly added injected register
JDR3	Injectionchanneldataregister3	0xBC	The newly added injected register

2.9 COMP

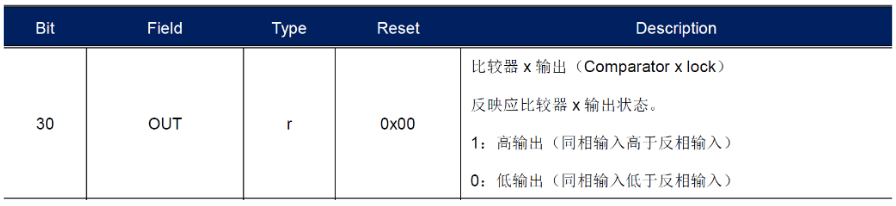
MM32F0130 has one more group of COMP than MM32F0140. MM32F0140 and MM32F0130 are compatible in most part of COMP and their registers and register bit functions are the same. The main difference is that MM32F0140 modifies the internal input options of COMP's INM (COMP_INM5,6,7), and also adds comparator output mode selection. The multiplexed pins of the MM32F0140's COMP input are adjusted. For more details, please refer to the AF function table of the MM32F0140's GPIO.

MM32F0130	MM32F0140	Descriptions
 PA[3] PA[2] PA[1] PA[0] COMP_INP + CMP1/2 COMP_INM − CRV PA[0] PA[2] PA[5] PA[4]	 COMP_INP0 COMP_INP1 COMP_INP2 COMP_INP3 COMP_INP COMP_INM CMP CRV COMP_INM0 COMP_INM1 COMP_INM2 COMP_INM3 COMP_INM5 COMP_INM6 COMP_INM7	Compared to MM32F0130, MM32F0140 modifies the pin multiplex of COMP and adds input channel.

MM32F0140’s COMP pin input selection:



Bit	Field	Type	Reset	Description
6: 4	INM_SEL	rw	0x00	比较器 x 反相输入选择（Comparator x inverting input se-lection） 这些位用于选择连接到比较器 x 的反相输入的信号源。 比较器: 000：COMP_INM0 001：COMP_INM1 010：COMP_INM2 011：COMP_INM3 100：COMP_INM4（CRV） 101：VSSA 110：VSSA 111：VSSA



Bit	Field	Type	Reset	Description
30	OUT	r	0x00	比较器 x 输出（Comparator x lock） 反映应比较器 x 输出状态。 1：高输出（同相输入高于反相输入） 0：低输出（同相输入低于反相输入）

2.10 PWR

MM32F0140 and MM32F0130 are compatible in most fucntions of PWR and their registers and register bit functions are the same. The main difference is that MM32F0140 adds the Deep Stop configuration in the low power mode and the configuration of wake-up time delay in the Standby wake-up mode.

In terms of the overall performance, enhancements have been made on power-up/down by MM32F0140.

PWR	Register	Descriptions
CR2	Control register 2	Newly added
SR	Status register	Newly added
SCR	clear status register	Newly added
CFGR	Configuration register	Newly added

2.11 MCU_DBG

MM32F0140 and MM32F0130 are compatible in most functions of DBG and the functions achieved by registers and register bits are the same. The main difference is that the ID value of MM32F0140 is different from that of MM32F0130 and such value can be used for the detection and judgement of different series; besides, MM32F0140 adds the configuration of entering STOP mode when debugging, and whether TIM16, TIM17 and TIM18 continue to work when they are selected at debugging.

	MM32F0130	MM32F0140
DBGMCU_CR_STOP	NA	Support
DBGMCU_CR_TIM16_17_14	NA	Support

2.12 HWDIV

MM32F0140 and MM32F0130 are wholly compatible in DIV.

2.13 CRC

MM32F0140 and MM32F0130 are mutually compatible in CRC and their registers and register bit functions are the same. The main difference is as follow: MM32F0140 supports internal wait. The operation has been finished when data is read. For MM32F0130, one or two _NOP() operations need to be added when necessary. Besides, MM32F0104 adds one MIR register and the register CRC_MIR is used to keep the intermediate results of data.

Add new function bits like POLY32_MGN:

With POLY32_MGN bit of CRC_CTRL to set the algorithm as CRC32/MPEG-2 or CRC32;

The input can be set as 8-bit/16-bit/32-bit via CRC_BitSel bit of register CRC_CTRL;

The input can be set as big-endian or little-endian via BIG_EI bit of register CRC_CTRL;

The output can be set as big-endian or little-endian via BIG_EO bit of CRC_CTRL register;

Read register CRC_MIR to keep the intermediate result of the data.

CRC	MM32F0130	MM32F0140
MIR register	NA	Yes
BITSEL of CR register	NA	Yes
BIG_EI of CR register	NA	Yes
BIG_EO of CR register	NA	Yes

2.14 I2C

MM32F0140 and MM32F0130 are compatible in most functions of I2C and their registers and register bit functions are the same. The main difference is that

MM32F0140 adds the slave multi-address function.

	MM32F0130	MM32F0140
I2C_CR_SLV_TX_DIS	NA	Yes
I2C_CR_PAD_SEL	NA	MM32F0140 supports I2C PIN remap

MM32F0130 only supports one I2C，while MM32F0140 supports two I2Cs.

2.15 SPI

MM32F0140 and MM32F0130 are compatible in most functions of SPI and their registers and register bit functions are the same. The main difference is that MM32F0140 adds I2S and PIN remap configuration function.

The following are newly added registers:

	MM32F0130	MM32F0140
CFGR(I2S)	NA	Support

MM32F0140 adds control bit and status bit, including the newly added control bit and status bit for I2S:

Bit Field	Bit Description	Descriptions
GCR_PAD_SEL	SPI Bus mapping transformation	MM32F0140 starts to support SPI PIN remap
SPI_SR_BUSY	I2S or SPI Data transferring flag	MM32F0140 starts to support
SPI_SR_CHSIDE	I2S Left or Right channel transmission flag	MM32F0140 starts to support
ISR_FRE_INTF	I2S frame transmission error flag bit	MM32F0140 starts to support
IER_FRE_IEN	I2S frame transmission interrupt enable bit	MM32F0140 starts to support
ICR_FRE_ICLR	I2S frame transmission interrupt clear bit	MM32F0140 starts to support

2.16 UART

With enhancements, UART function of MM32F0140 covers and is compatible with that of MM32F0130. The main difference can be seen below: MM32F0140 supports three UARTs while MM32F031 only supports two UARTs; In the UART module, MM32F0140 adds IrDA function, thus, relevant registers and control bits are added accordingly.

	Register	Descriptions
IRDA	Infrared function control register	Newly added

2.17 IWDG

The IWDG of the MM32F0140 series is compatible with the IWDG of the MM32F0130 series. The registers and register bits are the same. The main difference is that the MM32F0140 modifies the function meaning of the register that triggers the threshold value of the IWDG interrupt.

MM32F0130 supports interrupt and reset; if interrupt is selected, reset cannot be supported; and if reset is selected, interrupt function is no longer supported.

The MM32F0140 supports simultaneous enabling of interrupts and resets. If the preset interrupt trigger value (the set value in IGEN) is reached, the interrupt is triggered; the value of the register CNT continues to decrease. If the watch-dog is still not fed in time, the reset will be triggered after the value is reduced to 0.

	Register	Description
IGEN	IWDG Interrupt Generator Register	Modify meaning
CNT	IWDG Counter Register	Modify meaning
PS	IWDG Prescaler Counter Register	Modify meaning

2.18 WWDG

MM32F0140 is wholly compatible with MM32F0130 in WWDG and their registers and register bits are the same.

2.19 TIM1

MM32F0140 is compatible with MM32F0130 in TIM1.

2.20 TIM2, TIM3, 14, 16, 17

MM32F0140 and MM32F0130 are wholly compatible in TIM2, TIM3, 14, 16 and 17 and their registers and register bit functions are the same.

2.21 FlexCAN

Compared with MM32F0130, MM32F0140 replaces the classic CAN function with the new FlexCAN. For specific functions, please refer to the FlexCAN section in the user manual.

2.22 USB

Compared to MM32F0130, MM32F0140 has no USB.

2.23 RTC

Compared to MM32F0130, MM32F0140 has no RTC.

3 Firmware migration with the use of library

This section introduces how to migrate applications from MM32F0130 library to applications from MM32F0140 library.

The libraries of MM32F0130 and MM32F0140 are the same in structure and compatible with CMSIS. For all mutually compatible peripherals, they all use the same driver naming rule and API, so as to ensure the maximum compatibility.

When migrating from MM32F0130 to MM32F0140, application update can be done with only a small amount of work.

3.1 Migration steps

To update the application code of MM32F0130 and run on the MM32F0140 library, the following steps need to be followed:

A. Replace library function

   Download MM32F0140's library functions and sample packages from the official website. For example, if the project implementation structure from customers needs to be the same as the MM32F0130's library function samples on the official website, it is very convenient to replace the Device directory in the original MM32F0130 library functions with the files in the Device directory. For any changes in the relevant folders, they need to be re-included in the project files.

B. Modify and replace project files

   KEIL, IAR or Eclipse, etc. are used for development in implementing application. The project files relative to the application have to be modified accordingly after replacing the library.

   First, download KEIL's Pack package or IAR's Pack of MM32F0140, install these packages accordingly to select the MM32F0140's devices (Device Name), and check the relevant loading files of FLASH, RAM, as well as FLASH downloading algorithm.

   Next, re-include the relevant library files and header files such as started.s file, included register header files reg_xxx.h and hal_xxx.h among the project engineering files.

C. Modify the code (functions, structures, variables) of peripherals that have been changed. For example, the application codes of RCC, PWR, GPIO, FLASH and ADC driver that have been used in the applications, need to be checked and replaced one by one.

MM32F0140 library and sample packages provide rich examples in which, the specific use of different peripherals is demonstrated. For more information, please refer to the following introduction.

3.2 System clock configuration and migration

Same structure can be found in the configuration files of system clock source for MM32F0130 and MM32F0140. In order to distinguish these two files, the file names are different and users only need to use

\Device\MM32F0140\Source\system_mm32f0140.c file

Replace:

Device\MM32F0130\Source\system_MM32F0130.c file

However, these two products differ from each other in the frequency selection range, PLL configuration, and delay wait period configuration required for system clock switch. With the use of CMSIS layer, these differences can be hidden from the application code through the call of SystemInit() function in system_MM32F0140.c; then, users only need to replace system_MM32F0130.c file with system_MM32F0140.c file, and select the macro definition of the correct system clock source based on the needs emerged from implementing projects. How to achieve SystemInit() function is provided in both files. The microcontroller’s system clock source can be configured at startup and before transferring to the main() program by calling this function.

3.3 ADC configuration and migration

ADC of MM32F0140 covers that of MM32F0130. On this basis, MM32F0140 adds injected sampling.

The differences in library functions are mainly reflected as follows:

MM32F0130 ADC related functions	MM32F0140 ADC related functions
	u16 ADC_GetInjectedConversionValue(ADC_TypeDef* adc, ADC_INJ_SEQ_Channel_TypeDef off_addr)
	u16 ADC_GetInjectedCurrentConvertedValue(ADC_TypeDef* adc)
	void ADC_AutoInjectedConvCmd(ADC_TypeDef* adc, FunctionalState state)
	void ADC_ExternalTrigInjectedConvCmd(ADC_TypeDef* adc, FunctionalState state)
	void ADC_ExternalTrigInjectedConvertConfig(ADC_TypeDef* adc, EXTER_INJ_TRIG_TypeDef ADC_ExtInjTrigSource)
	void ADC_InjectedConvCmd(ADC_TypeDef* adc, FunctionalState state)
	void ADC_InjectedSequencerChannelConfig(ADC_TypeDef* adc, ADC_INJ_SEQ_Channel_TypeDef off_addr, ADCCHANNEL_TypeDef channel)
	void ADC_InjectedSequencerConfig(ADC_TypeDef* adc, u32 event, u32 sample_time)
	void ADC_InjectedSequencerLengthConfig(ADC_TypeDef* adc, ADC_INJ_SEQ_LEN_TypeDef Length)

MM32F0130 ADC related functions	MM32F0140 ADC related functions
	void ADC_SetInjectedOffset(ADC_TypeDef* adc, ADC_INJ_SEQ_Channel_TypeDef off_addr, u16 value)
	void ADC_SoftwareStartInjectedConvCmd(ADC_TypeDef* adc, FunctionalState state)

ADC_JSQR sample is added to the sample for injected sampling.

3.4 COMP configuration and migration

The COMP function of MM32F0140 covers that of MM32F0130. On this basis, MM32F0140 adds the internal input options of INM (COMP_INM5,6,7), and comparator output mode selection. Available parameters are added, but library functions and header files remain unchanged. Only available parameters are newly added. MM32F0130 has one more group of COMP function than MM32F0140, and the latter can only select COMP1. Besides, the multiplexed pins have different GPIOs. When input is configured, corresponding changes should be made according to the actual configuration.

3.5 CRC configuration and migration

CRC of MM32F0140 covers that of MM32F0130. On this basis, available configuration parameters and intermediate values are added.

MM32F0130 CRC related functions	MM32F0140 CRC related functions

u32 CRC_CalcCRC(u32 data)
{
    CRC->DR = data;
    __NOP();
    __NOP();
    return (CRC->DR);
}
	u32 CRC_CalcCRC(u32 data)
	{
	    CRC->DR = data;
	
	    return (CRC->DR);
	}

	u32 CRC_RevData(u32 value)

3.6 Debug configuration and migration

Debug of MM32F0140 covers that of MM32F031. On this basis, MM32F0140 adds the configuration of entering STOP mode when debugging, and whether TIM16, TIM17 and TIM18 continue to work when they are selected at debugging.

Thus, library functions and header files remain unchanged. Only available parameters are newly added.

MM32F0130 DEBUG related parameters	MM32F0140 DEBUG related parameters
	DBGMCU_CR_STOP_FOR_LDO
	DBGMCU_CR_TIM16_STOP
	DBGMCU_CR_TIM17_STOP

3.7 FLASH operation and migration

MM32F0140 and MM32F0130 are mutually compatible in Flash and their registers and register bit functions are the same. The main difference is the way in which read-protect is set.

For specific read-protect setting method of MM32F0140, please refer to the following code:

LibSamples\FLASH\FLASH_SetReadProtect\HARDWARE\FLASH\flash.c

3.8 I2C migration

MM32F0140 and MM32F0130 are mutually compatible in I2C and their registers and register bit functions are the same. The main difference is that MM32F0140 adds the slave multi-address function.

The newly added functions are seen below:

MM32F0130 related functions	MM32F0140 related functions
NA	void I2C_SlaveReceivedAddressMask(I2C_TypeDef* i2c, u16 mask) u32 I2C_GetSlaveReceivedAddr(I2C_TypeDef* i2c)

3.9 SPI migration

MM32F0140 and MM32F0130 are basically compatible in SPI. The main difference is that MM32F0140 adds I2S function and register functions related to setting I2S.

The newly added functions are seen below:

MM32F0130 related functions	MM32F0140 related functions
NA	void I2S_Cmd(SPI_TypeDef* spi, FunctionalState state) void I2S_Init(SPI_TypeDef* spi, I2S_InitTypeDef* I2S_InitStruct)

SPI_I2S_DMA_PollingPlaytone is added to the sample of MM32F0140 to demonstrate I2S function.

4 Conclusion

The above introduction briefly describes the matters and methods for migrating from MM32F0130 to MM32F0140. For more detailed descriptions, please refer to the User Manulas and related sample projects of MM32F0130 and MM32F0140.

5 Record histroy

Table 2 Record history

Date	Version	Content
2021/11/03	1.00	AN0052 First public release