

前言

本应用笔记旨在帮助您分析从现有的SXX32F0xx/GX32F3x0器件移植到AT32F415器件所需的步骤。本文档收集了最重要的信息，并列出了需要注意的重要事项。

要将应用程序从SXX32F0xx/GX32F3x0系列移植到AT32F415系列，用户需要分析硬件移植、外设移植和固件移植。

支持型号列表：

支持型号	AT32F415xx
------	------------

目录

1	AT32F415 与 SXX32F0xx/GX32F3x0 异同.....	5
1.1	相同点概述	5
1.2	差异点概述	5
2	硬件移植	7
3	外设移植	8
3.1	外设对比	8
3.2	存储器映射	9
3.3	功能区别	10
3.3.1	RCC 接口	10
3.3.2	DMA 接口	10
3.3.3	中断向量	12
3.3.4	EXTI 中断源选择	14
3.3.5	GPIO 接口	14
3.3.6	ADC 接口	15
3.3.7	SPI 接口	15
3.3.8	I2C 接口	15
3.3.9	USART 接口	16
3.3.10	RCC PLL 使能前需配置参考时钟配置表	16
3.4	功能增强	17
3.4.1	高频 PLL 设定	17
3.4.2	AT32F415 PLL 预分频器	18
3.4.3	安全库区保护	18
3.4.4	32 位定时器	18
3.4.5	USB 接口	18
3.5	外设使用区别	20
3.5.1	内部温度传感器	20

3.5.2	GPIO 5V 容忍管脚兼容	20
3.5.3	ERTC 日历读取区别	20
3.5.4	USART 智能卡模式下接收数据异常	21
3.5.5	SPI 无法释放 PA15 给其他外设	21
3.5.6	PWR 以 WFE 进入 low-power 模式的差别	21
4	版本历史	23

表目录

表 1.AT32F415 与 SXX32F0xx/GX32F3x0 差异概述	5
表 2.硬件引脚相容性	7
表 3.复用功能相容性	7
表 4.外设相容性.....	8
表 5.存储器映射关系差异	9
表 6.RCC 接口差异	10
表 7.DMA 接口差异	10
表 8.中断向量差异	12
表 9.GPIO 接口差异	14
表 10.ADC 接口差异.....	15
表 11.文档版本历史	23

1 AT32F415 与 SXX32F0xx/GX32F3x0 异同

AT32F415系列微控制器大部分兼容SXX32F0xx和GX32F3x0系列，有在对整体性价比考量下进行部分功能取舍，同时亦有强化许多新功能，故导致有些许地方存在不同，详述于本文档。

1.1 相同点概述

- 管脚定义：相同封装管脚定义相同。为扩增的外设作管脚复用定义延伸
- 编译工具：完全相同，例如Keil, IAR

1.2 差异点概述

表 1. AT32F415 与 SXX32F0xx/GX32F3x0 差异概述

	AT32F415	SXX32F0xx	GX32F3x0
系统			
内核	Cortex-M4	Cortex-M0	Cortex-M4
系统时钟	主频 150MHz, APB1 75MHz, APB2 75MHz	主频 48MHz, APB 48MHz	主频 108MHz, APB1 54MHz, APB2 54MHz
STOP 唤醒(调压器处于低功耗模式)	360us	最大 9us	-
Standby 唤醒	600us	60.4us	37.9ms
系统存储器			
SRAM 容量	全系列型号 32KB	全系列 16KB	依型号分 4/6/8/16KB
系统存储器 (System Memory)	全系列 18KB，相对比多支持以下功能： 1. USB DFU方式的ISP烧写 2. 对闪存内容进行CRC校验	12KB	3KB
闪存 16-bit 写入时间	42us	60us	86us
闪存页擦除时间	8ms	40ms	300ms
闪存整片擦除时间	10ms	40ms	1.6s
安全保护			
安全库区设定	支持，详细描述请参阅 <i>AT32F415 参考手册 4.3.5 节和安全库区(sLib) 应用指南.pdf</i>	无	无
外设			
备份寄存器	全系列支持 20 组 32 位备份寄存器	仅支持 5 组 32 位备份寄存器	仅支持 5 组 32 位备份寄存器
32 位通用定时器	TMR2, TMR5 为 32 位定时器	仅 TMR2	仅 TMR2
16 位通用定时器	TMR3, 4, 9, 10, 11	TMR3, 14, 15, 16, 17	TMR3, 14, 15, 16, 17
基本定时器	无	TMR6, 7	TMR6

	AT32F415	SXX32F0xx	GX32F3x0
HSI 微调	透过 ACC, 输入只有 USB	透过 CRS, 输入有外部引脚 TSC_SYNC、LSE、和 USB 可选	透过 CTC, 输入有外部引脚 CTC_SYNC、LXTAL、和 USB 可选
WKUP 引脚	1 根	6 根	5 根
TAMPER_RTC 引脚	1 根	2 根	2 根
USB	USB OTG, 无 NOE 引脚	USB Device, 有 NOE 引脚	USB Device
SPI 主机四线式	无	无	有
红外发射器	无	有	有
触控	无	有	有
ADC	2Msps (max ADCCLK = 28MHz)	1Msps (max ADCCLK = 14MHz)	2.86Msps (max ADCCLK = 40MHz)
电气特性			
电压范围	2.6V~3.6V	2.4V~3.6V	2.6V~3.6V
内核电压	1.2V 操作电流更低	1.8V	1.2V
ESD 参数	HBM:5KV, CDM:1000V	HBM:2KV, CDM:500V	HBM:6KV, CDM:2000V
运行模式	16.9mA@48MHz	24.1mA@48MHz	11.99mA@48MHz
睡眠功耗	13.3mA@48MHz	15.0mA@48MHz	5.36mA@48MHz
停机功耗	680uA	5.1uA	92.0uA
待机功耗	3.6uA	2.3uA	6.9uA

2 硬件移植

AT32F415与SXX32F0xx/GX32F3x0的各引脚基本上相兼容，只有极少数的引脚存在差异，转化起来极其方便，详细信息如下表。

表 2. 硬件引脚相容性

AT32F415				SXX32F0xx				GX32F3x0			
QFN32	QFP48	QFP64	引脚	QFN32	QFP48	QFP64	引脚	QFN32	QFP48	QFP64	引脚
5	5	5	PD0->OSCIN	-	5	5	PF0->OSCIN	5	5	5	PF0->OSCIN
6	6	6	PD1->OSCCOUT	-	6	6	PF1->OSCCOUT	6	6	6	PF1->OSCCOUT
-	-	18	PF4	-	-	18	VSS	-	-	18	PF4
-	-	19	PF5	-	-	19	VDD	-	-	19	PF5
-	35	47	PF6	-	35	47	VSS	-	35	47	PF6
-	36	48	PF7	-	36	48	VDDIO2	-	36	48	PF7

表 3. 复用功能相容性

	AT32F415	SXX32F0xx	GX32F3x0
引脚名	复用功能	复用功能	复用功能
PC2	-	SPI2_MISO, I2S2_MCK	-
PC3	-	SPI2_MOSI, I2S2_SD	-
PA0	-	-	I2C2_SCL
PA1	-	-	I2C2_SDA
PA4	-	-	SPI2_NSS
PA6	-	USAER3_CTS, I2S1_MCK	I2S1_MCK
PC4	-	USART3_TX	-
PC5	-	USART3_RX	-
PB0	USART3_RTS	USART3_CK	USART2_RX
PB1	USART3_CTS	USART3_RTS	SPI2_SCK
PB10	-	SPI2_SCK, I2S_CK	-
PB13	-	I2C2_SCL	-
PB14	-	I2C2_SDA, I2S2_MCK	-
PC6	I2S2_MCK	-	I2S1_MCK
PA8	-	-	USART2_TX
PA9	-	-	I2C1_SCL
PA10	-	-	I2C1_SDA
PA13	-	-	SPI2_MISO
PA14	-	USART2_TX	SPI2_MOSI, USART2_TX
PA15	-	USART2_RX	USART2_RX
PD2	-	USART3_RTS	-
PB4	-	I2S1_MCK	-
PB9	-	SPI2_NSS, I2S2_WS	I2S1_MCK

3 外设移植

3.1 外设对比

AT32F415在外设部分和SXX32F0xx/GX32F3x0相对比有些外设还是存在有一定的区别，且有些相对来说算是一个全新的设计。故针对这些外设需在应用层级的程序开发中进行修改或参考新外设驱动进行全新开发。

表 4. 外设相容性

外设	SXX32F0x/GX32F3x0	AT32F415	兼容性	
			引脚排列	固件驱动
SPI	Y	Y	相同	部分兼容
WWDG	Y	Y	NA	完全兼容
IWDG	Y	Y	NA	部分兼容
DBGMCU	Y	Y	NA	部分兼容
CRC	Y	Y	NA	部分兼容
EXTI	Y	Y	相同	部分兼容
CEC	Y	NA	NA	NA
DMA	Y	Y	NA	部分兼容
TMR	Y	Y	相同	部分兼容
PWR	Y	Y	NA	部分兼容
SDIO	NA	Y	NA	NA
USART	Y	Y	相同	部分兼容
I2C	Y	Y	相同	部分兼容
DAC	Y	NA	NA	NA
ADC	Y	Y	相同	部分兼容
RTC	Y	Y	相同	部分兼容
FLASH	Y	Y	NA	部分兼容
GPIO	Y	Y	新 GPIO	不兼容
CAN	NA	Y	NA	NA
USB OTG	NA	Y	NA	NA
触摸感应	Y	NA	NA	NA
COMP	Y	Y	相同	部分兼容
SYSCFG	Y	NA	NA	NA
AFIO	NA	Y	NA	NA

3.2 存储器映射

因性能上的考量和优化，AT32F415在架构上进行了更深层次的调整。相对于SXX32F0xx与GX32F3x0而言，外设地址和总线的排列分布有一定的区别，下面就详细的列出地址映射区别和总线所属关系。

表 5.存储器映射关系差异

外设	SXX32F0xx/GX32F3x0		AT32F415	
	总线	基址	总线	基址
TSC	AHB1	0x40024000	NA	NA
CRC		0x40023000	AHB	0x40023000
FLITF		0x40022000		0x40022000
RCC		0x40021000		0x40021000
DMA		0x40020000		0x40020000
GPIOF	AHB2	0x48001400	APB2	0x40011C00
GPIOE		0x48001000	NA	NA
GIOD		0x48000C00	APB2	0x40011400
GPIOC		0x48000800		0x40011000
GPIOB		0x48000400		0x40010C00
GPIOA		0x48000000		0x40010800
ACC	NA	NA		0x40015800
DBGMCU	APB	0x40015800	NA	NA
TMR17		0x40014800		
TMR16		0x40014400		
TMR15		0x40014000		
USART1		0x40013800	APB2	0x40013800
SPI1		0x40013000		0x40013000
TMR1		0x40012C00		0x40012C00
ADC		0x40012400		0x40012400
EXTI(SYSCFG)		0x40010400		0x40010400
SYSCFG+COMP		0x40010000	NA	NA
CEC		0x40007800	APB1	0x40007800
DAC		0x40007400		0x40007400
PWR		0x40007000		0x40007000
CRS		0x40006C00	NA	NA
I2C2		0x40005800	APB1	0x40005800
I2C1		0x40005400		0x40005400
USART2		0x40004400		0x40004400
SPI2		0x40003800		0x40003800
IWDG		0x40003000		0x40003000
WWDG		0x40002C00		0x40002C00
RTC		0x40002800		0x40002800
TMR14		0x40002000	NA	NA
TMR6		0x40001000		
TMR3		0x40000400	APB1	0x40000400
TMR2		0x40000000		0x40000000

外设	SXX32F0xx/GX32F3x0		AT32F415	
	总线	基址	总线	基址
USBFS SRAM		0x40006000	NA	NA
USBFS		0x40005C00		
TMR7		0x40001400		
TMR4	NA	NA	APB1	0x40000800
CAN	APB	0x40006400		0x40006400
UART5		0x40005000		0x40005000
UART4		0x40004C00		0x40004C00
USART3		0x40004800		0x40004800
DMA2	AHB1	0x40020400	AHB	0x40020400
USBOTG	NA	NA		0x50000000
TMR11			APB2	0x40015400
TMR10				0x40015000
TMR9				0x40014C00
TMR5				0x40000C00
AFIO				0x40010000

3.3 功能区别

3.3.1 RCC 接口

AT32F415与SXX32F0xx/GX32F3x0对比在RCC部分有以下区别

表 6.RCC 接口差异

RCC	SXX32F0xx	GX32F3x0	AT32F415
HSI	8MHz RC	8MHz RC	48MHz RC 除频 6
HSE	4-32MHz	4-32MHz	4-25MHz
LSI	40KHz RC	40KHz RC	40KHz RC
LSE	32.768KHz	32.768KHz	32.768KHz
HSI14	用于 ADC	NA	NA
HSI28	NA	用于 ADC	NA
HSI48	48MHz RC	48MHz RC	48MHz RC
RTC CLK	LSI、LSE、HSE/32	LSI、LSE、HSE/32	LSI、LSE、HSE/128
最大系统时钟频率	48MHz	108MHz	150MHz
MCO	HSI14、SYSCLK、HSI、HSE、PLL/2、PLL、LSE、LSI、HSI48	HSI28、LSI、LSE、HSI、HSE、PLL、PLL/2	ADCCLK、USB48、SYSCLK、LSI、LSE、HSI、HSE、PLL/2

3.3.2 DMA 接口

AT32F415与SXX32F0xx/GX32F3x0对比在DMA部分有以下区别

表 7.DMA 接口差异

外设	DMA 请求	SXX32F0xx	GX32F3x0	AT32F415
TMR17	TMR17_UP	DMA_Channel1/DMA_Channel2	DMA_Channel1/DMA_Channel2	NA

外设	DMA 请求	SXX32F0xx	GX32F3x0	AT32F415
	TMR17_CH1	DMA_Channel1/DMA_Channel2	DMA_Channel1/DMA_Channel2	
TMR16	TMR16_UP	DMA_Channel3/DMA_Channel4	DMA_Channel3/DMA_Channel4	NA
	TMR16_CH1	DMA_Channel3/DMA_Channel4	DMA_Channel3/DMA_Channel4	
TMR15	TMR15_UP	DMA_Channel5	DMA_Channel5	NA
	TMR15_CH1	DMA_Channel5	DMA_Channel5	
	TMR15_TRIG	DMA_Channel5	DMA_Channel5	
	TMR15_COM	DMA_Channel5	DMA_Channel5	
USART1	USART1_Rx	DMA_Channel3/DMA_Channel5	DMA_Channel3/DMA_Channel5	DMA1_Channel5
	USART1_Tx	DMA_Channel2/DMA_Channel4	DMA_Channel2/DMA_Channel4	DMA1_Channel4
SPI1	SPI1_Rx	DMA_Channel2	DMA_Channel2	DMA1_Channel2
	SPI1_Tx	DMA_Channel3	DMA_Channel3	DMA1_Channel3
TMR1	TMR1_UP	DMA_Channel5	DMA_Channel5	DMA1_Channel5
	TMR1_CH1	DMA_Channel2/DMA_Channel6	DMA_Channel2	DMA1_Channel2
	TMR1_CH2	DMA_Channel3/DMA_Channel6	DMA_Channel3	DMA1_Channel3
	TMR1_CH3	DMA_Channel5/DMA_Channel6	DMA_Channel5	DMA1_Channel6
	TMR1_CH4	DMA_Channel4	DMA_Channel4	DMA1_Channel4
	TMR1_TRIG	DMA_Channel4	DMA_Channel4	DMA1_Channel4
	TMR1_COM	DMA_Channel4	DMA_Channel4	DMA1_Channel4
ADC	ADC	DMA_Channel1	DMA_Channel1	DMA1_Channel1
		DMA_Channel2	DMA_Channel2	
DAC	DAC_Channel1	DMA_Channel3	DMA_Channel3	NA
	DAC_Channel2	DMA_Channel4		
I2C2	I2C2_Rx	DMA_Channel5	DMA_Channel5	DMA1_Channel5
	I2C2_Tx	DMA_Channel4	DMA_Channel4	DMA1_Channel4
I2C1	I2C1_Rx	DMA_Channel3/ DMA_Channel7	DMA_Channel3	DMA1_Channel7
	I2C1_Tx	DMA_Channel2/ DMA_Channel6	DMA_Channel2	DMA1_Channel6
SDIO	SDIO	NA	NA	DMA2_Channel4
USART2	USART2_Rx	DMA_Channel5	DMA_Channel5	DMA1_Channel6
	USART2_Tx	DMA_Channel4	DMA_Channel4	DMA1_Channel7
SPI2	SPI2_Rx	DMA_Channel4	DMA_Channel4	DMA1_Channel4
	SPI2_Tx	DMA_Channel5	DMA_Channel5	DMA1_Channel5
TMR6	TIM6_UP	DMA_Channel3	DMA_Channel3	NA
TMR3	TMR3_UP	DMA_Channel3	DMA_Channel3	DMA1_Channel3
	TMR3_CH1	DMA_Channel4	DMA_Channel4	DMA1_Channel6
	TMR3_TRIG	DMA_Channel4	DMA_Channel4	DMA1_Channel6
	TMR3_CH3	DMA_Channel2	DMA_Channel2	DMA1_Channel2
	TMR3_CH4	DMA_Channel3	DMA_Channel3	DMA1_Channel3
TMR2	TMR2_UP	DMA_Channel2	DMA_Channel2	DMA1_Channel2
	TMR2_CH1	DMA_Channel5	DMA_Channel5	DMA1_Channel5
	TMR2_CH2	DMA_Channel3/DMA_Channel7	DMA_Channel3	DMA1_Channel7
	TMR2_CH3	DMA_Channel1	DMA_Channel1	DMA1_Channel1
	TMR2_CH4	DMA_Channel4/DMA_Channel7	DMA_Channel4	DMA1_Channel7
TMR4	TMR4_UP	NA	NA	DMA1_Channel7
	TMR4_CH1			DMA1_Channel1

外设	DMA 请求	SXX32F0xx	GX32F3x0	AT32F415
	TMR4_CH2 TMR4_CH3			DMA1_Channel4 DMA1_Channel5
UART4	UART4_Rx UART4_Tx	DMA_Channel6 DMA_Channel7	NA	DMA2_Channel3 DMA2_Channel5
USART3	USART3_Rx USART3_Tx	DMA_Channel6/DMA_Channel3 DMA_Channel7/DMA_Channel2	NA	DMA1_Channel3 DMA1_Channel2
TMR5	TMR5_UP TMR5_CH1 TMR5_CH2 TMR5_CH3 TMR5_CH4 TMR5_TRIG	NA	NA	DMA2_Channel2 DMA2_Channel5 DMA2_Channel4 DMA2_Channel2 DMA2_Channel1 DMA2_Channel1

3.3.3 中断向量

AT32F415与SXX32F0xx/GX32F3x0对比在中断号及中断向量部分有以下区别

表 8.中断向量差异

位置	SXX32F0xx	GX32F3x0	AT32F415
0	WWDG	WWDG	WWDG
1	PVD	PVD	PVD
2	RTC	RTC	TAMPER
3	FLASH	FLASH	ERTC
4	RCC	RCC	EFLASH
5	EXTIO_1	EXTIO_1	RCC
6	EXTI2_3	EXTI2_3	EXTIO
7	EXTI4_15	EXTI4_15	EXTI1
8	TSC	TSC	EXTI2
9	DMA_CH1	DMA_CH1	EXTI3
10	DMA_CH2_CH3	DMA_CH2_CH3	EXTI4
11	DMA_CH4_CH5	DMA_CH4_CH5	DMA_CH1
12	ADC_COMP	ADC_COMP	DMA_CH2
13	TIM1_BRK_UP_TRG_COM	TIM1_BRK_UP_TRG_COM	DMA_CH3
14	TIM1_CC	TIM1_CC	DMA_CH4
15	TIM2	TIM2	DMA_CH5
16	TIM3	TIM3	DMA_CH6
17	TIM6_DAC	TIM6_DAC	DMA_CH7
18	TIM7	TIM7	ADC1
19	TIM14	TIM14	CAN1_TX
20	TIM15	TIM15	CAN1_RXS0
21	TIM16	TIM16	CAN1_RXS1
22	TIM17	TIM17	CAN_SCE
23	I2C1	I2C1_EV	EXTI5_9
24	I2C2	I2C2_EV	TMR1_BRK_TMR9
25	SPI1	SPI1	TMR1_OV_TMR10

位置	SXX32F0xx	GX32F3x0	AT32F415
26	SPI2	SPI2	TMR1_TRG_COM_TMR11
27	USART1	USART1	TMR1_CC
28	USART2	USART2	TMR2
29	USART3_4	NA	TMR3
30	CEC_CAN	CEC_CAN	TMR4
31	USBFS	Reserved	I2C1_EV
32	Reserved	I2C1_ER	I2C1_ER
33	Reserved	Reserved	I2C2_EV
34	Reserved	I2C2_ER	I2C2_ER
35	Reserved	Reserved	SPI1
36	Reserved	Reserved	SPI2
37	Reserved	Reserved	USART1
38	Reserved	Reserved	USART2
39	Reserved	Reserved	USART3
40	Reserved	Reserved	EXTI10_15
41	Reserved	Reserved	ERTCArm
42	Reserved	USBFS_Wakeup	Reserved
43	Reserved	Reserved	Reserved
44	Reserved	Reserved	Reserved
45	Reserved	Reserved	Reserved
46	Reserved	Reserved	Reserved
47	Reserved	Reserved	Reserved
48	Reserved	DMA_CH6_CH7	Reserved
49	Reserved	Reserved	SDIO
50	Reserved	Reserved	TMR5
51	Reserved	Reserved	Reserved
52	Reserved	Reserved	UART4
53	Reserved	Reserved	UART5
54	Reserved	Reserved	Reserved
55	Reserved	Reserved	Reserved
56	Reserved	Reserved	DMA2_CH1
57	Reserved	Reserved	DMA2_CH2
58	Reserved	Reserved	DMA2_CH3
59	Reserved	Reserved	DMA2_CH4_5
60	Reserved	Reserved	Reserved
61	Reserved	Reserved	Reserved
62	Reserved	Reserved	Reserved
63	Reserved	Reserved	Reserved
64	Reserved	Reserved	Reserved
65	Reserved	Reserved	Reserved
66	Reserved	Reserved	Reserved
67	Reserved	Reserved	USBOTG
68	Reserved	Reserved	Reserved

位置	SXX32F0xx	GX32F3x0	AT32F415
69	Reserved	Reserved	Reserved
70	Reserved	Reserved	COMP1
71	Reserved	Reserved	COMP2
72	Reserved	Reserved	ACC
73	Reserved	Reserved	Reserved
74	Reserved	Reserved	Reserved
75	Reserved	Reserved	Reserved
76	Reserved	Reserved	Reserved
77	Reserved	Reserved	Reserved
78	Reserved	Reserved	Reserved
79	Reserved	Reserved	Reserved
80	Reserved	Reserved	Reserved
81	Reserved	Reserved	Reserved
82	Reserved	Reserved	Reserved
83	Reserved	USBFS_Global	Reserved

3.3.4 EXTI 中断源选择

在外部中断配置方式上，AT32F415和SxxF0xx/GxxF3x0存在一定的差异。AT32F415使用AFIO_EXTICRx寄存器进行外部中断的配置，SxxF0xx/GxxF3x0通过使用SYSCFG_EXTICRx寄存器进行配置。此处只是EXTICRx寄存器的映射地址发生了改变，EXTIx配置的意义一样。

3.3.5 GPIO 接口

表 9.GPIO 接口差异

GPIO	SXX32F0xx	GX32F3x0	AT32F415
输入模式	悬空 PU PD	悬空 PU PD	悬空 PU PD
输出模式	PP PP+PU PP+PD OD OD+PU OD+PD	PP PP+PU PP+PD OD OD+PU OD+PD	PP OD
功能复用	PP PP+PU PP+PD OD OD+PU OD+PD	PP PP+PU PP+PD OD OD+PU OD+PD	PP OD
速度	2MHz 10MHz 48MHz	2MHz 10MHz 50MHz	无速度定义

● GPIO功能复用

AT32F415

1. I/O用作复用功能的配置取决于所使用的外设模式。例如，USART Tx引脚应配置为复用功能推挽，而USART Rx引脚应配置为输入悬空或输入上拉。
2. 为在不用封装上优化外设I/O功能数量，可将某些复用功能重新映射到其他引脚上。采用AFIO_MAPx寄存器进行各外设引脚的复用配置。

SXX32F0xx/GX32F3x0

1. 任何外设再使用到功能复用时，都必须将I/O配置为复用功能，之后才能正确使用I/O功能。
2. 引脚复用及映射采用GPIOx_AFR1和GPIOx_AFR2这一组寄存器进行配置。

3.3.6 ADC 接口

AT32F415与SXX32F0xx/GX32F3x0在ADC上对比有如下区别

表 10.ADC 接口差异

ADC	AT32F415		GX32F3x0		SXX32F0xx
通道数	16 通道+2 内部通道(无 V _{BAT} /2)		16 通道+3 内部通道		16 通道+3 内部通道
转换模式	单一/连续/间断/扫描		单一/连续/间断/扫描		单一/连续/间断/扫描
分辨率	12 位		12 位		12 位
外部触发	规则组	注入组	规则组	注入组	TIM1_TRGO TIM1_CC4 TIM2_TRGO TIM3_TRGO TIM15_TRGO
	TMR1 CC1	TMR1 TRGO	TMR1 CC1	TMR1 TRGO	
	TMR1 CC2	TMR1 CC4	TMR1 CC2	TMR1 CC4	
	TMR1 CC3	TMR2 TRGO	TMR1 CC3	TMR2 TRGO	
	TMR2 CC2	TMR2 CC1	TMR2 CC2	TMR2 CC1	
	TMR3 TRGO	TMR3 CC4	TMR3 TRGO	TMR3 CC4	
	TMR4 CC	TMR4 TRGO	TMR14 CC	TMR14 TRGO	
	EXTI line11	EXTI line15	EXTI line11	EXTI line15	
	SWSTR	JSWSTR	SWSTR	JSWSTR	
	TMR1_TRGO	TMR1 CC1			
电源要求	2.6V 到 3.6V		2.6V 到 3.6V		2.4V 到 3.6V

3.3.7 SPI 接口

- AT32F415内置的SPI外设与GX32F3x0相兼容。
- 对比SXX32F0xx在SPI接口上存在部分的不同。

经性价比等全方面的考虑和分析，剔除了SXX32F0xx上的部分SPI的以下功能特性。

1. NSS的脉冲模式和TI模式配置。
2. 数据帧长度的可编程控制。
3. Tx/Rx FIFO缓冲区。

3.3.8 I2C 接口

- AT32F415内置的I2C外设与GX32F3x0相兼容。
- 相比于SXX32F0xx的I2C有较大的区别。

二者在结构、特性和编程上都不同，因此基于SXX32F0xx的I2C部分的代码需要重新编写后才可以在AT32F415上运行。

Demo请参考AT32F4xx_StdPeriph_Lib_Vx.x.x\Project\AT_START_F415\Examples\I2C

详细的编程发放和接口请参阅AT32F415说明手册第13章节。

3.3.9 USART 接口

- AT32F415内置的USART外设相比于SXX32F0xx/GX32F3x0的USART有较大的区别。

二者在结构、特性和编程上都不同，因此USART部分的代码需要重新编写后才可以AT32F415上运行。

Demo请参考AT32F4xx_StdPeriph_Lib_Vx.x.x\Project\AT_START_F415\Examples\USART

详细的编程方法和接口请参阅AT32F415说明手册第14章节。

3.3.10 RCC PLL 使能前需配置参考时钟配置表

- 描述：

因AT32F415 RCC IP的更新，在系统时钟配置流程中，RCC PLL配置和使能之前需要根据实际所使用的PLL时钟源来对参考时钟配置表PLL_FREF参数进行配置(寄存器RCC_PLL[26:24])。

位 26: 24	PLL_FREF: PLL 参考时钟配置表 (PLL Configuration Table) 由软件配置使用何种 PLL 参考时钟配置表 000: PLL 使用 4M 参考时钟配置表; 001: PLL 使用 6M 参考时钟配置表; 010: PLL 使用 8M 参考时钟配置表; 011: PLL 使用 12M 参考时钟配置表; 100: PLL 使用 16M 参考时钟配置表; 101: PLL 使用 25M 参考时钟配置表; 110: 保留; 111: 保留。
----------	--

- 解决方法：

示例使用8MHz晶振配置144MHz系统时钟的修改方法如下：

```
static void SetSysClockTo144M(void)
{
    ...

    #if defined (AT32F415xx)
        RCC->CFG |= (uint32_t)(RCC_CFG_PLLRC_HSE | RCC_CFG_PLLMULT18);
    #else
        RCC->CFG |= (uint32_t)(RCC_CFG_PLLRC_HSE | RCC_CFG_PLLMULT18 |
        RCC_CFG_PLLRANGE_GT72MHZ);
    #endif

    ...
}
```

修改为

```
static void SetSysClockTo144M(void)
{
    ...

    RCC->PLL |= (0x2 << 24); // PLL使用8M参考时钟配置表
    #if defined (AT32F415xx)
```



```

RCC->CFG |= (uint32_t)(RCC_CFG_PLLRC_HSE | RCC_CFG_PLLMULT18);
#else
RCC->CFG |= (uint32_t)(RCC_CFG_PLLRC_HSE | RCC_CFG_PLLMULT18 |
RCC_CFG_PLLRANGE_GT72MHZ);
#endif
...
}

```

3.4 功能增强

3.4.1 高频 PLL 设定

- 描述:

- AT32F415内置的PLL可输出最高可达150MHz时钟，设定略有不同

- 使用范例:

- SXX32F0xx PLL设定程序范例:

```
RCC->CFGR |= (uint32_t)(RCC_CFGR_PLLSRC_HSE | RCC_CFGR_PLLMULL5);
```

- AT32F415 PLL设定程序范例:

```

#define RCC_CFG_PLLMULT1 ((uint32_t)0x20000000) /*!< PLL input clock * 17 */
#define RCC_CFG_PLLMULT18 ((uint32_t)0x20040000) /*!< PLL input clock * 18 */
#define RCC_CFG_PLLMULT19 ((uint32_t)0x20080000) /*!< PLL input clock * 19 */
#define RCC_CFG_PLLMULT20 ((uint32_t)0x200C0000) /*!< PLL input clock * 20 */
...
#define RCC_CFG_PLLMULT61 ((uint32_t)0x60300000) /*!< PLL input clock * 61 */
#define RCC_CFG_PLLMULT62 ((uint32_t)0x60340000) /*!< PLL input clock * 62 */
#define RCC_CFG_PLLMULT63 ((uint32_t)0x60380000) /*!< PLL input clock * 63 */
#define RCC_CFG_PLLMULT64 ((uint32_t)0x603C0000) /*!< PLL input clock * 64 */

```

- 需要特别注意的:

- 当AT32F415内置的PLL为108MHz以上时钟时，PLL设定略有不同，需要操作自动滑顺频率切换功能

- 144MHz PLL使用范例:

打开system_SXX32f10x.c找到当前的系统时钟频率配置函数（需经过上述PLL配置），如144MHz函数:

```
static void SetSysClockTo144(void)
```

增加如下斜黑体部分:

```

/* Wait till PLL is ready */
while((RCC->CR & RCC_CR_PLLRDY) == 0)
{
}

*((unsigned int *)0x40021054) |= (0x30);// 开启自动滑顺频率切换功能

```

```

/* Select PLL as system clock source */
RCC->CFGR &= (uint32_t)((uint32_t)~(RCC_CFGR_SW));
RCC->CFGR |= (uint32_t)RCC_CFGR_SW_PLL;
/* Wait till PLL is used as system clock source */
while ((RCC->CFGR & (uint32_t)RCC_CFGR_SWS) != (uint32_t)0x08)
{
}

*((unsigned int *)0x40021054) &=~ (0x30); //关闭自动滑顺频率切换功能

```

3.4.2 AT32F415 PLL 预分频器

- 描述：
 - 由于主频提高至150MHz，相关预分频器做出扩增
 - USB预分频器扩增支援/2, /2.5, /3, /3.5, /4输出
 - ADC预分频器扩增支持/12, /16输出
 - 请参阅AT32F415参考手册3.3.2 RCC_CFG寄存器叙述

3.4.3 安全库区保护

- 描述
 - 目前越来越多的微控器(MCU)应用需要使用到复杂的算法及中间件解决方案(middleware solution)，因此，如何保护软件方案商开发出来的核心算法等知识产权代码(IP-Code)，便成为微控制器应用中一项很重要的课题。

为因应这一重要的需求，AT32F415系列提供了安全库区(sLib)的功能，以防止重要的IP-Code被终端用户的程序做修改或读取，进而达到保护的目的。

- 使用范例
 - 请参考《AT32F415安全库区(sLib)应用指南.pdf》
 - Demo请参考AT32F4xx_StdPeriph_Lib_Vx.x.x\Utilities\AT32F415_SLIB_Demo

3.4.4 32 位定时器

- 描述：
 - AT32F415 TMR2/TMR5 都可配置成32位定时器
- 使用范例：
 - TMR2->CTRL1|=0X0400; //开启TMR2的32位模式
 - TMR5->CTRL1|=0X0400; //开启TMR5的32位模式

(相关变量须改为32位，请参考AT32F415库文件 at32f4xx_tim.c, at32f4xx_tim.h)

3.4.5 USB 接口

- 描述
 - AT32F415支持USBOTG，SxxF0xx/GxxF3x0只支持USBDEV。AT32F415内置一个集成了收发器的USB OTG全速(12 Mb/s)设备、主机和OTG模式控制模块。USB OTG FS模块兼容USB2.0和OTG1.3协议。端点可由软件配置，并支持挂起/恢复机制。故AT32F415上的USB接口相较于SxxF0xx/GxxF3x0是一个全新的外设，且功能及性能更加强大。

- 使用范例
 - Host Demo请参考AT32F4xx_StdPeriph_Lib_Vx.x.x\Project\AT_START_F415\Examples\OTG_Host
 - Device Demo请参考AT32F4xx_StdPeriph_Lib_Vx.x.x\Project\AT_START_F415\Examples\OTG_Device

3.5 外设使用区别

3.5.1 内部温度传感器

- 描述：
AT32F415温度传感器部分与GX32F3x0兼容，与SXX32F0xx有较大区别。
- 解决方法：
 - AT32F415和GX32F3x0按照数据手册中的值并利用下列公式得出温度：
温度(°C) = $\{(V25 - VSENSE) / Avg_Slope\} + 25$
这里：
V25 = VSENSE在25°C时的数值
Avg_Slope = 温度与VSENSE曲线的平均斜率 (单位为mV/°C).
 - SXX32F0xx的计算公式如下：
温度(°C) = $\{(110 - 30) / (TS_CAL2 - TS_CAL1) * (TS_DATA - TS_CAL1)\} + 30$
TS_CAL1 = 30°C时的校准值。
TS_CAL2 = 110°C时的校准值。
- 例程参考
AT32F4xx_StdPeriph_Lib_Vx.x.x\Project\Examples\AT_START_F415\ADC\Temperature

3.5.2 GPIO 5V 容忍管脚兼容

- 描述：
PA10、PA11、PA12与PD0、PD1不属于5V电压输入容忍管脚，故这些管脚输入电平不可超过VDD + 0.3V。
- 解决方法：
使用时请留意此限制。

3.5.3 ERTC 日历读取区别

- 描述：
AT32F415 ERTC在读取时间(ERTC_TIME)和日期(ERTC_DATE)寄存器时和SXX32F0xx、GX32F3x0有一个区别，AT32F415需先读取一下ERTC_CTRL寄存器以更新时间和日期寄存器值。
- 解决方法：
 - 方法1：使用库函数开发
使用V1.2.3及其以上的BSP，无需修改，使用步骤和SXX32F0xx一样
 - 方法2：使用寄存器开发
 - ◆ SXX32F0xx读取步骤：
time = ERTC->TIME;
date = ERTC->DATE;
 - ◆ AT32F415读取步骤：
volatile uint32_t reg = ERTC->CTRL;

```
time = ERTC->TIME;
```

```
date = ERTC->DATE;
```

3.5.4 USART 智能卡模式下接收数据异常

- 描述：
初始化USART的智能卡模式时，若先使能USART再使能智能卡模式，USART 智能卡模式使能后的第一帧数据长度时段内无法接收数据。
- 解决方法：
使能USART之前先使能智能卡模式。

```
/******先使能智能卡模式*****/  
/* Enable the NACK Transmission */  
USART_SmartCardNACKCmd(USARTx, ENABLE);  
/* Enable the Smartcard Interface */  
USART_SmartCardCmd(USARTx, ENABLE);  
  
/******配置完成后再使能 USART *****/
```

3.5.5 SPI 无法释放 PA15 给其他外设

- 描述：
当SPI Remap到PA15/PB3/PB4/PB5这一组后，SPI就会强制占用PA15/PB3/PB4/PB5。哪怕实际此时SPI使用软件NSS，PA15（NSS Pin）也只能作为普通GPIO翻转，不能再释放给其他外设使用，比如无法再用于Timer的PWM输出。
- 解决方法：
使用时请留意此限制。

3.5.6 PWR 以 WFE 进入 low-power 模式的差别

- AT32F415系列采用Cortex-M4内核，WFE操作有所区别，需要按如下使用范例操作。
- 使用范例：

```
/* Request Wait For Event */  
__SEV();  
__WFE();  
__WFE();
```

3.5.7 CAN接收数据域期间出现位填充错误时导致下一帧数据错位

- 描述：
CAN在接收数据域期间若出现“位填充错误”时，将导致收到的下一帧数据出现错位，但后续帧又自动恢复正常的现象。
- 解决方法：
开启CAN的上次错误中断号对应的错误中断，在CAN错误中断的中断函数内检测到出现位填充错

误时，复位CAN，并重新调用CAN初始化函数。以CAN1为例，其典型示例代码如下：

```
__IO uint32_t err_index = 0;
void CAN1_SCE_IRQHandler(void)
{
    if (CAN_GetINTStatus (CANx,CAN_INT_LEC) == SET )
    {
        err_index = CAN1->ESTS;
        CAN_ClearINTPendingBit (CANx, CAN_INT_LEC);
        if(err_index & 0x00000010) ///<判定是否出现位填充错误
        {
            CAN_Reset(CAN1);
            /*调用 CAN 初始化函数*/;
        }
    }
}
```

4 版本历史

表 11.文档版本历史

日期	版本	变更
2019.09.24	1.0.0	最初版本
2020.06.22	1.0.1	新增 RCC PLL 使能前需配置参考时钟配置表
2020.08..03	1.0.2	新增 ERTC 日历读取区别 新增 USART 智能卡模式下接收数据异常
2020.10..26	1.0.3	新增 SPI 无法释放 PA15 给其他外设 修改表格 硬件引脚相容性 新增表格 复用功能相容性
2021.03.04	1.0.4	新增 PWR 以 WFE 进入 low-powe 模式的差别
2021.05.06	1.0.5	增加 CAN 接收数据域期间出现位填充错误时导致下一帧数据错位

重要通知 - 请仔细阅读

买方自行负责对本文所述雅特力产品和服务的选择和使用，雅特力概不承担与选择或使用本文所述雅特力产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为雅特力授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在雅特力的销售条款中另有说明，否则，雅特力对雅特力产品的使用和/或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途(及其依据任何司法管辖区的法律的对应情况)，或侵犯任何专利、版权或其他知识产权的默示保证。

雅特力产品并非设计或专门用于下列用途的产品：(A) 对安全性有特别要求的应用，如：生命支持、主动植入设备或对产品功能安全有要求的系统；(B) 航空应用；(C) 汽车应用或汽车环境；(D) 航天应用或航天环境，且/或(E) 武器。因雅特力产品不是为前述应用设计的，而采购商擅自将其用于前述应用，即使采购商向雅特力发出了书面通知，风险由购买者单独承担，并且独力负责在此类相关使用中满足所有法律和法规要求。

经销的雅特力产品如有不同于本文档中提出的声明和/或技术特点的规定，将立即导致雅特力针对本文所述雅特力产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大雅特力的任何责任。

© 2020 雅特力科技 (重庆) 有限公司 保留所有权