

从SXX32F103移植到AT32F413

前言

这篇迁移指南旨在帮助您分析从现有的SXX32F103器件移植到AT32F413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SXX32F103系列移植到AT32F413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	AT32F413xx
------	------------

目录

1	AT32F413 与 SXX32F103 异同	6
1.1	相同点概述	6
1.2	差异点概述	6
2	快速替换 SXX32F103 芯片	8
2.1	快速验证兼容性	8
2.2	BSP 工程替换步骤.....	8
3	AT32F413 兼容性解析	9
3.1	功能增强.....	9
3.1.1	高频 PLL 设定	9
3.1.2	AT32F413 PLL 预分频器.....	10
3.1.3	FPU 的 ARM® 32 位的 Cortex®-M4F	10
3.1.4	安全库区保护	11
3.1.5	内存容量扩展	11
3.1.6	加挂 SPI 闪存(Bank3 功能)	11
3.1.7	32 位定时器	12
3.1.8	SPI1 复用为 I2S1.....	13
3.1.9	USBDEV 缓冲区	13
3.2	外设使用区别.....	13
3.2.1	内部温度传感器	13
3.2.2	GPIO 5V 容忍管脚兼容	14
3.2.3	GPIO 5V 容忍管脚特性	14
3.2.4	GPIO 非 5V 容忍管脚使用限制.....	14
3.2.5	Standby 模式下自动使能 PA11/PA12 弱下拉	14
3.2.6	GPIO 不识别 JTAG/SWD	15
3.2.7	BOOT0 自带下拉电阻.....	15
3.2.8	Standby 模式下自动使能 PA0 引脚下拉电阻	15

3.2.9	PVDEN 从关闭到开启会产生 PVD 大于 PVD 门限事件	15
3.2.10	USB 模块系统时钟频率须不小于 12MHz	15
3.2.11	USB_DP 存在内部上拉电阻	16
3.2.12	TMRx 单脉冲输出, CCx 值设置为 1 且 DIV=0 时波形输出异常	16
3.2.13	TMRx 在 CNTEN 未使能情况下, 强制输出无效	16
3.2.14	TMRx 计时不准	16
3.2.15	SPI 使用 DMA 传输的时候, 接收数据偏移一个 byte	16
3.2.16	开启 WWDG 并执行 Flash sector erase 操作会导致程序复位	16
3.2.17	SDIO 在发生 RX over run 之后接收数据异常	17
3.2.18	ADC 规则组多信道连续转换模式下, 切换注入组通道配置导致规则组转换错序	17
3.2.19	ADC 配置在混合的同步规则+交替触发模式, 开启 AUTO INJECT 功能时异常	17
3.2.20	USART 接收数据时, 当两包数据的停止位与起始位部分重叠时, 数据接收异常	17
3.2.21	USART 初始化完后, 立即清 TC flag, 然后通过判断 TC 发数据, 程序会卡死	17
3.2.22	USART 的 RDNE 标志清除异常	18
3.2.23	USART 同步模式, 特殊配置条件下, 时钟极性不能在串口 IDLE 态正常发出	18
3.2.24	Standby 唤醒后 RTC 中断未正常进入	18
3.2.25	进入 Standby mode 后, 无法再次下载代码	18
3.2.26	使能 IWDG 后立即进入 Standby mode 会导致 IWDG 复位	19
3.2.27	GPIO 在 50MHz 配置下可能存在过冲现象	19
3.2.28	CAN 从睡眠切换到初始化模式时必须清掉睡眠模式请求	19
3.2.29	Dual CAN 应用在过滤器组使用上的注意要点	20
3.2.30	USART 智能卡模式下接收数据异常	20
3.2.31	PWR 以 WFE 进入 low-power 模式的差别	20

4	版本历史	21
---	------------	----

表目录

表 1. 差异点概述.....	6
表 2. 六种应用情形	8
表 3. 支持的 SPI Flash 型号	12
表 4. 共享脚位描述	12
表 5. 文档版本历史	21

图目录

图 1. VSENSE 对温度理想曲线图.....	14
---------------------------	----

1 AT32F413 与 SXX32F103 异同

AT32F413系列微控制器基本兼容SXX32F103系列，同时强化许多功能的关系，有些许地方与SXX32F103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
- 寻址空间：内存与寄存器逻辑地址相同。扩增的外设占用SXX32保留空间
- 函数库文件：函数库相同。部分头文件依照强化功能优化
- 编译工具：完全相同，例如Keil, IAR

1.2 差异点概述

表 1. 差异点概述

	AT32F413	SXX32F103xC/xB/x8
系统		
内核	Cortex-M4 并支持 DSP 指令及浮点运算单元 FPU	Cortex-M3
系统时钟	主频 200MHz, APB1 100MHz, APB2 100MHz	主频 72MHz, APB1 36MHz, APB2 72MHz
启动	8 ms	2.5 ms
重置	3.6 ms	-
Standby 唤醒	3.6 ms	50 us
系统存储器		
SRAM 容量	全系列型号皆可支持扩充模式 64KB	SXX32F103xC (48KB) SXX32F103xB (20KB) SXX32F103x8 (20KB)
加挂 SPI 闪存	支持加挂 SPI Flash 为 Bank3, 最高达 16MB	无支援
系统存储器 (System Memory)	全系列 18KB, 比 SXX32F103 多支持以下功能: 1. USB DFU方式的ISP烧写 2. 对Bank3进行ISP烧写 3. 对闪存内容进行CRC校验	依型号分 6KB/2KB
选择字节 (Option Byte)	48 Byte, 扩增以下设定功能: 1. SRAM模式设定 2. 4 Byte自定义议字段(例如开发商ID) 3. 8 Byte Bank3 加密钥匙	16 Byte
闪存 16-bit 写入时间	50 μ s	52.5 μ s
闪存页擦除时间	50 ms (AT32F413xC) 40 ms (AT32F413xB/x8)	40 ms
闪存整片擦除时间	800 ms	40 ms
安全保护		
安全库区设定	支持, 详细描述请参阅 AT32F413 参考手册 5.3.5 节和 安全库区(sLib) 应用指南.pdf	无

	AT32F413	SXX32F103xC/xB/x8
外设		
备份寄存器	全系列支持 42 组半字节备份寄存器	中、小容量系列仅支持 10 组
SPI1 支援 I2S	SPI1 可支持 I ² S 功能	SPI1 仅为 SPI 功能
I2S 支援	48PIN 即支援 I ² S	48PIN 无 I ² S
32 位定时器	TMR2, TMR5 为 32 位定时器	皆为 16 位
USB 缓冲区	可扩充为 1280 Byte	512 Byte
ADC	2Msps (max ADCCLK=28MHz)	1Msps (max ADCCLK=14MHz)
温度传感器	正温度系数	负温度系数
电压范围	2.6V~3.6V	2.0V~3.6V
内核电压	1.2V 操作电流更低	1.8V
ESD 参数	HBM:5KV, CDM:1000V	HBM:2KV, CDM:500V
运行模式	28.4 mA @ 72MHz	51 mA @ 72MHz
睡眠功耗	23.9 mA @ 72MHz	29.5mA @ 72MHz
停机功耗	490 uA	25 uA
待机功耗	9.9 uA	2.1 uA

2 快速替换 SXX32F103 芯片

2.1 快速验证兼容性

- 步骤一：比对外设规格、Flash容量、SRAM容量等，解焊SXX32F103，换成AT32F413对应型号
- 步骤二：使用雅特力ICP, ISP或KEIL, IAR下载SXX32F103 HEX文件或BIN文件。
- 步骤三：如果有需要，下载SXX32F103 HEX文件或BIN文件以外的资料或进行系统校正。
- 步骤四：查看程序能否正常运行。
- 步骤五：其他问题快速排查请参考[3.2 外设使用区别](#)。
- 步骤六：如果经过上述步骤后程序仍无法正常运行，请参考本文件其他章节，或连络代理商及雅特力科技技术支持人员协助解决。
- *Note: 由于AT32F413采用灵活的内存扩展设计，如[3.1.5 内存容量扩展](#)中描述，内部闪存存储器存在非零等待区域，会导致有些SXX32F103程序在AT32F413上运行效能不佳。关于如何提高运行效能，请参考AT32提供的应用手册004_Performance_Optimization。*

2.2 BSP 工程替换步骤

- 六种应用情形概述如下

表 2. 六种应用情形

序号	使用什么 BSP/Pack 开发	是否使用 AT32F413 新功能	处理方法
1	AT32F413 BSP/Pack	是/否	1. 结合 3.2 外设使用区别 修改对应程序
2	SXX32F103 BSP/Pack	否	1. 结合 3.2 外设使用区别 修改对应程序
3	SXX32F103 寄存器操作	否	1. 结合 3.2 外设使用区别 修改对应程序
4	SXX32F103 BSP + AT32 Pack	否	1. 修改 FPU 设置 2. 结合 3.2 外设使用区别 修改对应程序
5	SXX32F103 寄存器操作	是	1. 需使用处理过的 BSP/Pack 2. 结合 3.2 外设使用区别 修改对应程序
6	SXX32F103 BSP/Pack	是	1. 需使用处理过的 BSP/Pack 2. 结合 3.2 外设使用区别 修改对应程序

- 详细请参阅《AT32F413 BSP和Pack应用指南》

3 AT32F413 兼容性解析

3.1 功能增强

3.1.1 高频 PLL 设定

- 描述:
 - AT32F413内置的PLL可输出200MHz时钟，设定略有不同
 - 该PLL支持两频段，以72MHz为分界，最高可达200MHz，须根据输出频率设定PLL_RANGE寄存器
- 使用范例:
 - SXX32F103 PLL设定程序范例：

```
RCC->CFGR |= (uint32_t)(RCC_CFGR_PLLSRC_HSE | RCC_CFGR_PLLMULL9);
```

AT32F413 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 */

#define RCC_CFG_PLLRANGE ((uint32_t)0x80000000) /*!< PLL Frequency range */
#define RCC_CFG_PLLRANGE_LE72MHZ ((uint32_t)0x00000000) /*!< When PLL frequency is less than or equal to 72MHz */
#define RCC_CFG_PLLRANGE_GT72MHZ ((uint32_t)0x80000000) /*!< When PLL frequency is greater than 72MHz */
```

以设置 72MHz 为例：

```
RCC->CFG |= (uint32_t)(RCC_CFG_PLLRC_HSE | RCC_CFG_PLLMULT9 |
RCC_CFG_PLLRANGE_LE72MHZ);
```

以设置 200MHz 为例：

```
RCC->CFG |= (uint32_t)(RCC_CFG_PLLRC_HSE | RCC_CFG_PLLMULT25 |
RCC_CFG_PLLRANGE_GT72MHZ);
```

- 需要特别注意的：

当AT32F413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
- 168MHz 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168MHz函

数:

```
static void SetSysClockTo168(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
```

3.1.2 AT32F413 PLL 预分频器

- 描述:
 - 由于主频提高至200MHz, 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
 - 主时钟输出(CLKOUT)扩增支持LSI, LSE, PLLCLK/4, USB48M, ADCCLK输出, 并加入输出预分频器。
 - 请参阅AT32F413参考手册3.3.2 RCC_CFG寄存器叙述

3.1.3 FPU 的 ARM® 32 位的 Cortex®-M4F

- 描述:
 - 带存储器保护单元(MPU)
 - 内建单周期乘法和硬件除法
 - 内建浮点运算(FPU)
 - 具有DSP指令集外设新增功能
 - 例程参考
AT32F4xx_StdPeriph_Lib_V1.x.x\Project\Examples\AT_START_F413\CortexM4\FPU
AT32F4xx_StdPeriph_Lib_V1.x.x\Libraries\CMSIS\DSP_Lib\Examples

3.1.4 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。
 - 为适应这一重要的需求, AT32F413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《安全库区(sLib)应用指南.pdf》

3.1.5 内存容量扩展

- 描述:
 - AT32F413支持的内存扩展功能不同, 可将内存可切换为16K,32K或64K字节
 - 内存扩展功能透过选择字节EOPB0[1:0] (0x1FFF_F810) 设定:
 - 0x0: 片上内存为 64K 字节
 - 0x1: 片上内存为 16K 字节
 - 0x2: 片上内存为 64K 字节
 - 0x3: 片上内存为 32K 字节
 - 若开启此模式, 则具备零等待特性的闪存地址空间将发生调整, 以AT32F413RCT7为例, 内部闪存存储器与内存的三种配置关系如下:
 - 内存配置: 32 K字节 (出厂默认) 零等待: 96 K字节, 非零等待: 160 K字节
 - 内存配置: 64 K字节 零等待: 64 K字节, 非零等待: 192 K字节
 - 内存配置: 16 K字节 零等待: 112 K字节, 非零等待: 144 K字节
 - 选择字节EOPB0更新后须重置系统方能生效
 - 请参阅AT32F413参考手册5.3.4节
- AT32F4xx_StdPeriph_Lib_V1.x.x\Project\AT_START_F413\Examples\SRAM\extend_SRAM

3.1.6 加挂 SPI 闪存(Bank3 功能)

- 描述:
 - AT32F413支持SPIM闪存接口, 支持下列及其兼容之型号, 容量可达16MB。如果无法判别SPI Flash 型号是否兼容, 可以利用AT32 ICP工具的Bank3闪存自动侦测功能
 - 请参阅AT32F413参考手册5.2.2节与5.4.14节
 - 支持下表SPI Flash型号外, 也支持兼容下表型号命令集的SPI Flash芯片

表 3. 支持的 SPI Flash 型号

Vendor	SPI Flash 型号
ESMT	EN25F20A/EN25QH128A
Winbond	W25Q128V
GD	GD25Q16C/GD25Q32C/GD25Q64C/GD25Q80C/GD25Q127C

- 加挂闪存为Bank3闪存，使用地址0x0840_0000~0x1FFF_0000
- 加挂闪存使用以下管脚，各型封装均支持，如下方共享脚位描述：

表 4. 共享脚位描述

SPIM	Pin	QFN32	LQFP48/ QFN48	LQFP64	共享脚位描述
SPIM_SCK	PB1	15	19	27	ADC12_IN9 / TMR3_CH4 / TMR8_CH3N / TMR1_CH3N
SPIM_NSS	PA8	18	29	41	TMR1_CH1 / CLKOUT / USART1_CK / I2C2_SCL
SPIM_IO0	PA11	21	32	44	USB_DM / TMR1_CH4 / USART1_CTS / CAN1_RX
	PB10	-	21	29	I2C2_SCL / USART3_TX / TMR2_CH3
SPIM_IO1	PA12	22	33	45	USB_DM / CAN1_TX / USART1_RTS / TMR1_ETR
	PB11	-	22	30	I2C2_SDA / USART3_RX / TMR2_CH4
SPIM_IO2	PB7	30	43	59	TMR4_CH2 / I2C1_SDA / USART1_RX
SPIM_IO3	PB6	29	42	58	TMR4_CH1 / I2C1_SCL / USART1_TX / CAN2_TX / I2S1_MCK

- 加挂闪存为Bank3闪存时，内部AHB时钟频率（ f_{HCLK} ）最高限制为120 MHz.
- 例程参考
AT32F4xx_StdPeriph_Lib_V1.x.x\Project\Examples\AT_START_F413\FLASH

3.1.7 32 位定时器

- 描述：
 - AT32F413 TMR2/TMR5 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 AT32F413 库文件 at32f4xx_tim.c, at32f4xx_tim.h)

3.1.8 SPI1 复用为 I2S1

- 描述：
 - AT32F413 SPI1可复用为I2S1
 - 适用于48PIN/64PIN/100PIN/144PIN封装
 - 增加I2S1_MCK引脚
 - 其他引脚，寄存器空间和中断号，DMA通道同SPI1

3.1.9 USBDEV 缓冲区

- 描述：
 - AT32F413 USB设备(USBDEV)缓冲区最大可扩充到1280 Byte
 - 于USB768B(RCC_MISC[24])设定（请参阅AT32F413参考手册3.3.11小节）
 - 512Byte模式：缓冲区地址范围0x4000 6000~0x4000 63FF
 - 768Byte模式：缓冲区地址范围0x4000 7800~0x4000 83FF，缓冲区最大到1280，
 - 当CAN1和CAN2都不使用时，USB分组缓冲区最大为1280 Byte，
 - 当CAN1或者CAN2使能其中一个时，USB分组缓冲区最大为1024 Byte
 - 当CAN1和CAN2同时使能时，USB分组缓冲区最大为768 Byte
- 768 Byte模式使用范例：
 - `RCC->MISC |= 0x00000001 << 24; /*打开USB768Byte模式*/`
 - `#define PMAAddr 0x40007800 /*修改缓冲区起始地址，使用768Byte模式，一定要修改缓冲区地址*/`

3.2 外设使用区别

3.2.1 内部温度传感器

- 描述：

AT32F413温度传感器为正温度系数，SXX32F103为负温度系数
- 解决方法：

按照数据手册中的值并利用下列公式得出温度：

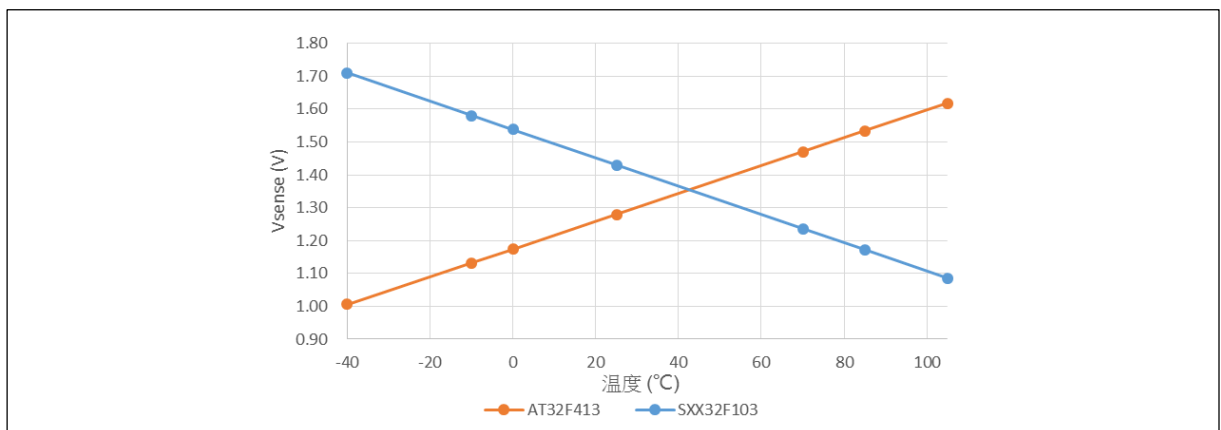
$$\text{温度}(\text{°C}) = \{(V25 - V\text{SENSE}) / \text{Avg_Slope}\} + 25$$

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).

V25和Avg_Slope必须根据数据手册中的典型值参与运算, AT32F413与SXX32F103不同。

图 1. V_{SENSE} 对温度理想曲线图

- 例程参考

AT32F4xx_StdPeriph_Lib_V1.x.x\Project\Examples\AT_START_F413\ADC\Temperature

3.2.2 GPIO 5V 容忍管脚兼容

- 描述：
各封装的 PA11, PA12 与 64/48 管脚封装的 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 $V_{DD} + 0.3V$ 。
- 解决方法：
使用时请留意此限制。

3.2.3 GPIO 5V 容忍管脚特性

- 描述：
5V 容忍管脚在浮空输入模式时，仍有约 10 μA 上拉电流能力以致管脚可能维持约 2.0V 电压让管脚输入逻辑仍判断为高电平。
- 解决方法：
外接 200 k Ω 或更小的下拉电阻避免 5V 容忍管脚电压维持在中间电平。
若推动 LED，建议设计线路让 GPIO 给低电平方式点亮 LED，否则会造成上电或覆位期间 LED 微点亮问题。

3.2.4 GPIO 非 5V 容忍管脚使用限制

- 描述：
对于非5V容忍管脚，灌入高于 $V_{DD} + 0.3V$ 时，会导致GPIO异常。
- 解决方法：
建议不要对非5V容忍管脚灌入高于 $V_{DD} + 0.3V$ 电压。

3.2.5 Standby 模式下自动使能 PA11/PA12 弱下拉

- 描述：
PA11 和 PA12 各有另一弱下拉电阻 330 k Ω 在 standby mode 时会由硬件自动使能。

- 解决方法:

使用时请留意此特性。PA11和PA12在standby mode时建议外部浮空或下拉以免系统产生微量额外耗电。

3.2.6 GPIO 不识别 JTAG/SWD

- 描述:

在使用AT32F413时,遇到disable JTAG/SWD接口后,无法下载程序并且找不到JTAG/SWD device。这里提供KEIL和IAR环境下的解决方法。

- 解决方法:

请参阅文档《AT32恢复下载办法1.0.x》

3.2.7 BOOT0 自带下拉电阻

- 描述:

使用BOOT0引脚时,由于BOOT0内部自带阻值约为90KΩ的下拉电阻(不可禁用),因此,在客户使用时,无需再额外外接下拉。

3.2.8 Standby 模式下自动使能 PA0 引脚下拉电阻

- 描述:

当芯片进入standby模式下时,PA0引脚的下拉电阻会由芯片内控制线路自动使能,以避免引脚浮空漏电。

- 解决方法:

请留意让PA0外部线路维持对此引脚0V输入电平,否则系统会有微量额外耗电。

3.2.9 PVDEN 从关闭到开启会产生 PVD 大于 PVD 门限事件

- 描述:

当VDD/VDDA=3.3V时,每次PVDEN位从OFF->ON,均会产生一个VDD大于门限中断

- 解决方法:

如果用户在开启PVD时不需要该中断,需要按照如下流程来开启PVD:

- 步骤一 PVD enable
- 步骤二 延时至少4us
- 步骤三 clear EXTI line 16 pending bit
- 步骤四 clear NVIC pending bit for EXTI line 16
- 步骤五 enable EXTI line 16 interrupt

3.2.10 USB 模块系统时钟频率须不小于 12MHz

- 描述:

系统时钟频率小于12MHz时,USB无法正常收发数据。

- 解决方法：
为满足USB正常收发数据的需求，系统时钟频率必须不小于12MHz。

3.2.11 USB_DP 存在内部上拉电阻

- 描述：
在USB_DP存在内部上拉电阻,支持全速设备。用户可省略外部1.5kΩ上拉电阻。

3.2.12 TMRx 单脉冲输出，CCx 值设置为 1 且 DIV=0 时波形输出异常

- 描述：
外部触发定时器，定时器使能了单脉冲模式，此时若定时器 CCx=1且DIV=0时，外部触发定时器使能后，当定时器停止计数后，CNT=1,导致波形输出异常。
- 解决方法：
设置定时器相应单脉冲输出通道的CCR值大于1 或设置该定时器的DIV值大于0。

3.2.13 TMRx 在 CNTEN 未使能情况下，强制输出无效

- 描述：
定时器CEN未使能情况下，即使信道输出配置完毕，配置OCxMODE强制输出高电平\低电平无效。
- 解决方法：
 - 方法1：开启定时器CNTEN。
 - 方法2：将引脚配置为普通IO输出。

3.2.14 TMRx 计时不准

- 描述：
当配置CNT>ARR时，定时器计时不准。
- 解决方法：
这里属于异常操作，使用者应该严格按照使用手册操作，即满足0<CNT<ARR配置。

3.2.15 SPI 使用 DMA 传输的时候，接收数据偏移一个 byte

- 描述：
SPI在使用DMA传输时，只开关DMA通道使能，不开关SPI的DMA请求使能，在某些情况下会出现接收数据偏移一个byte
- 解决方法：
开关DMA通道使能的同时开关SPI的DMA请求使能

3.2.16 开启 WWDG 并执行 Flash sector erase 操作会导致程序复位

- 描述：
Flash sector erase时间大于WWDG的最大时间，因此程序会发生看门狗复位
- 解决方法：

擦除前先关闭 WWDG，擦除完成之后在开启 WWDG

3.2.17 SDIO 在发生 RX over run 之后接收数据异常

- 描述：
如果使用的FIFO相关标志，例如RX FIFO half full来做接收数据判断，则接收数据会出现异常现象。
- 解决方法：
开 RX over run 中断，中断里重启 DPSM。

3.2.18 ADC 规则组多信道连续转换模式下，切换注入组通道配置导致规则组转换错序

- 描述：
ADC规则组多信道连续转换模式下，在规则组转换过程中更改注入组信道配置将导致规则组转换错序。
例如：规则转换 1、2、3、4 信道，在 2 信道时发生 2 次连续注入组通道切换，结果会有机率导致规则转换 1、2、2、3、4。
- 解决方法：
配置完注入组后，不再去切换注入组信道配置。

3.2.19 ADC 配置在混合的同步规则+交替触发模式，开启 AUTO INJECT 功能时异常

- 描述：
ADC在混合的同步规则+交替触发模式下，开启AUTO INJECT功能时，规则组与注入组皆不会正常转换。
- 解决方法：
在混合的同步规则+交替触发模式下，不要使用AUTO INJECT功能。

3.2.20 USART 接收数据时，当两包数据的停止位与起始位部分重叠时，数据接收异常

- 描述：
USART接收数据时，当两包数据的停止位与起始位部分重叠（第二包数据的起始位下降沿在第一包数据停止位的第4-10采样点来）时，会漏掉该第二包数据。
- 解决方法：
更改USART波特率，在我们串口波特率容忍度内，将波特率调整到最高，使得能在STOP位内尽快判断一帧数据起始位。

3.2.21 USART 初始化完后，立即清 TC flag，然后通过判断 TC 发数据，程序会卡死

- 描述：

在初始化USART后，立即清TC flag，然后判断TC置起后，再发送数据时，会因TC flag不能置起而卡死。

- 解决方法：

严格按照RM要求，在串口初始化结束后，不要去做清TC标志这个动作。

3.2.22 USART 的 RDNE 标志清除异常

- 描述：

普通异步通讯模式下，通过对RDNE Bit写0无法有效清零RDNE标志。

- 解决方法：

RDNE标志清除不要通过对RDNE Bit写0来清，只能采用对USART_DT的读操作方式。

3.2.23 USART 同步模式，特殊配置条件下，时钟极性不能在串口 IDLE 态正常发出

- 描述：

若先使能串口，再配置同步时钟使能CLKEN。时钟极性不能在串口IDLE态正常发出。

- 解决方法：

先配置同步时钟使能CLKEN，再使能串口。

3.2.24 Standby 唤醒后 RTC 中断未正常进入

- 描述：

从standby唤醒后，未正常进入RTC中断，同时RTC闹钟标志，溢出标志，秒标志均未被置起。

- 解决方法：

- 以PWR的SBF，WUF标志位作为从Standby mode唤醒依据，将RTC的中断处理函数RTC_IRQHandler中要处理的任务放到main函数内处理。
- 例程参考如下：在int main(void)中添加如下代码

```
RCC_APB1PeriphClockCmd(RCC_APB1PERIPH_PWR|RCC_APB1PERIPH_BKP,
ENABLE);
PWR_BackupAccessCtrl(ENABLE);
if((PWR->CSR&0x3) == 0x03)    //判断 SBF 和 WUF 是否置起
{
    ...//原 RTC 中断函数 RTC_IRQHandler 里面处理的任务
}
```

3.2.25 进入 Standby mode 后，无法再次下载代码

- 描述：

MCU进入Standby mode后，无法下载程序并且找不到JTAG/SWD device。

- 解决方法：

请参阅文档《AT32恢复下载办法1.0.x》

3.2.26 使能 IWDG 后立即进入 Standby mode 会导致 IWDG 复位

- 描述：
使能IWDG后立即进入Standby模式，会导致IWDG复位，造成系统复位。
- 解决方法：
在使能IWDG后需要等待至少100us以保证IWDG的正常工作，再进入Standby模式。
- 使用范例：
在IWDG使能之后，进入Standby模式之前添加如下代码，延时函数请参考BSP例程：

```
IWDG_Enable();  
Delay_us(100);  
PWR_EnterSTANDBYMode();
```

3.2.27 GPIO 在 50MHz 配置下可能存在过冲现象

- 描述：
对于AT32F413，当GPIO端口配置为输出模式，选择最大输出频率为50MHz时，可能存在过冲现象，有可能干扰电路并影响应用。
- 解决方法：
 - 情景一：对于常规使用，代码修改为GPIO_Speed_2MHz配置。
 - 情景二：对于强电流驱动，用户可根据AT32F413数据手册5.3.12的描述，依需求可配置成不同的速度参数MDEx。
- 情景一使用范例：
 - 针对使用 SXX32F103 BSP/Pack情形，修改方法如下黑体字部分

```
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5;  
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;  
/* 以下修改为针对使用 SXX32F103 BSP/Pack 情形 */  
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_2MHz;  
GPIO_Init(GPIOB, &GPIO_InitStructure);
```

- 情景二使用范例：
 - 略。

3.2.28 CAN 从睡眠切换到初始化模式时必须清掉睡眠模式请求

- 描述：
CAN处于睡眠模式时，切换到Init模式，在设置INRQ时还需要清SLP。
例如：如果当前 CAN 处于睡眠模式，通过软件写 CANx->MCTRL |= (uint32_t)CAN_MCTRL_INRQ;命令后，CAN 不会进入初始化模式，而还是会保持睡眠模式。
- 解决方法：

将 CANx->MCTRL |= (uint32_t)CAN_MCTRL_INRQ;命令更改为 CANx->MCTRL = (uint32_t)((CANx->MCTRL & (uint32_t)(~(uint32_t)CAN_MCTRL_SLP)) | CAN_MCTRL_INRQ);即可。

3.2.29 Dual CAN 应用在过滤器组使用上的注意要点

- 描述：
AT32F413的CAN1与CAN2相互独立，各自固定使用14个过滤器组。
区别与 ST 互联型产品，其针对双 CAN 设计 Bit 位 CAN2SB 来控制两个 CAN 之间的过滤器组的分配。
- 解决方法：
软件设计上，AT32F413 无 CAN2SB 这一 Bit 位。同时还需遵照两个 CAN 各自固定使用 14 个过滤器组这一基础来进行应用开发。

3.2.30 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 *****/
```

3.2.31 PWR以WFE进入low-power模式的差别

- AT32F413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();
```

4 版本历史

表 5. 文档版本历史

日期	版本	变更
2019.03.11	1.0.0	最初版本
2019.05.07	1.0.1	删除V1.0.0版“ TMR单脉冲模式下，有可能输出波形异常 ”重复描述
2019.05.07	1.0.2	更改“ GPIO 5V容忍管脚特性 ”描述
2019.05.07	1.0.3	新增“ Dual CAN应用在过滤器组使用上的注意要点 ”描述
2020.05.22	1.0.4	文档排版
2020.08.11	1.0.5	新增 USART智能卡模式下接收数据异常
2021.03.04	1.0.6	新增 PWR以WFE进入low-power模式的差别

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 保留所有权利