

PY32F071_PY32F072 的应用 注意事项

前言

PY32F071/PY32F072 系列微控制器采用高性能的 32 位 ARM® Cortex®-M0+ 内核，宽电压工作范围的 MCU。嵌入高达 128 Kbytes flash 和 16 Kbytes SRAM 存储器，最高工作频率 72 MHz。包含多种不同封装类型多款产品。

本应用笔记将帮助用户了解 PY32F071/PY32072 各个模块应用的注意事项，并快速着手开发。

表 1. 适用产品

类型	产品系列
微型控制器系列	PY32F071、PY32F072

目录

1	ADC 上电校准.....	3
1.1	注意事项	3
1.2	操作流程	3
1.3	代码示例	3
2	ADC 硬件设计注意事项	3
3	ADC 使用 TIMER_CC/TRGO 触发注意事项.....	3
4	ADC 外部中断(EXTI)触发注意事项.....	3
5	TTCAN 系统时钟配置注意事项.....	4
6	COMP 硬件设计.....	4
7	RST 配置为 GPIO 注意事项.....	4
8	IWDG 应用注意事项.....	4
9	SPI 最快传输速度.....	4
10	TIMER 使用 CC 中断注意事项.....	4
11	USB DMA 使用注意事项	4
12	IO 倒灌电流使 MCU 工作	5
12.1	注意事项	5
12.2	操作流程	5
12.3	代码示例	5
13	LPTIM 连续模式注意事项	5
14	LPTIM 单次模式注意事项	5
15	版本历史.....	6

1 ADC 上电校准

1.1 注意事项

- 当 ADC 的工作条件发生改变时 (VCC 改变是 ADC offset 偏移的主要因素, 温度改变次之), 推荐进行再次校准操作
- 第一次使用 ADC 模块前, 必须增加软件校准流程。

1.2 操作流程

- 使能 ADC 时钟, ADCEN=1;
- 初始化 ADC;
- ADC 校准

1.3 代码示例

```
static void APP_AdcConfig()
{
  ADC_HandleTypeDef      AdcHandle;          /*ADC 句柄*/
  AdcHandle.Instance = ADC1;                /*选择 ADC1*/
  __HAL_RCC_ADC_CLK_ENABLE();              /*使能 ADC 时钟*/
  if(HAL_ADCEx_Calibration_Start(&AdcHandle) != HAL_OK)
  {
    APP_ErrorHandler();
  }
  /*启动 ADC 校准*/
}
```

2 ADC 硬件设计注意事项

ADC 通道电压不能高于 VCC(即使 ADC 通道未配置为 AD 功能),否则 ADC 采样不准

3 ADC 使用 TIMER_CC/TRGO 触发注意事项

使用 TIMER_CC/TIMER_TRGO 触发 ADC 转换, ADC 时钟不能 8 分频

4 ADC 外部中断(EXTI)触发注意事项

当时钟 AHBCLK/APBCLK \geq 4 时, 不能设置 EXTI_11/EXTI_15 触发模式

5 TTCAN 系统时钟配置注意事项

系统时钟不得低于 CAN 模块工作时钟的 0.75 倍，CAN 的工作时钟是 20Mhz，所以系统时钟不得低于 15Mhz

6 COMP 硬件设计

当比较器的 VINM 输入信号为内部的模拟电压源时（例如 VREFINT，TSVIN，VREF1P2），外部输入通道 VINP 需要加一个电容(1nF)到地

7 RST 配置为 GPIO 注意事项

Reset 配置为 GPIO 后，每次上电瞬间有一个短暂的高电平(约 3.6ms)

8 IWDG 应用注意事项

如果需要休眠模式禁止 IWDG，可以通过 OPTION 设置冻结功能，即 FLASH_OTPR->IWDG_STOP = 0

9 SPI 最快传输速度

SPI模式	收/发模式	SPI最快速度
从机全双工	收	PCLK/4
从机全双工	发	PCLK/4
主机全双工	收	PCLK/4
主机全双工	发	PCLK/2

10 TIMER 使用 CC 中断注意事项

使能 CC 中断时，对应的分频系数 PSC 不得高于 80

11 USB DMA 使用注意事项

USB 使用 DMA 通讯时，同时只能配置一个通道进行 DMA 通讯

12 IO 倒灌电流使 MCU 工作

12.1 注意事项

- VCC 未供电的情况下，IO 倒灌电流使 MCU 工作，可通过软件配置规避

12.2 操作流程

- 硬件：对应 IO 口需串 100Ω~1KΩ电阻
- 上电初始化前需设置对应 IO 输出为开漏模式
- 延迟 5ms
- 程序正常初始化

12.3 代码示例

```
int main(void)
{
    GPIO_InitTypeDef  GPIO_InitStructure;

    HAL_Init()                /*初始化所有外设, Flash 接口, SysTick*/

    /* GPIO 初始化 */
    __HAL_RCC_GPIOC_CLK_ENABLE();    /*使能 GPIOC 时钟*/
    GPIO_InitStructure.Pin = GPIO_PIN_13;    /*使用 PC13*/
    GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_OD; /*开漏输出*/
    GPIO_InitStructure.Pull = GPIO_NOPULL;    /*不上拉*/
    GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_HIGH; /*GPIO 速度*/
    HAL_GPIO_Init(GPIOC, &GPIO_InitStructure);    /*初始化 GPIO */

    HAL_Delay(5);            /*延迟 5ms */
}
```

13 LPTIM 连续模式注意事项

LPTIM 连续模式从 STOP 唤醒，再次进入 STOP 前需等待 1 个 LSI 时钟周期（约需 40us，包含程序执行时间）

14 LPTIM 单次模式注意事项

LPTIM 单次模式从 STOP 唤醒，再次进入 STOP 前需等待 3 个 LSI 时钟周期（约需 120us，包含程序执行时间）

15 版本历史

版本	日期	更新记录
V1.0	2023.6.15	初版



Puya Semiconductor Co., Ltd.

IMPORTANT NOTICE

Puya Semiconductor reserves the right to make changes without further notice to any products or specifications herein. Puya Semiconductor does not assume any responsibility for use of any its products for any particular purpose, nor does Puya Semiconductor assume any liability arising out of the application or use of any its products or circuits. Puya Semiconductor does not convey any license under its patent rights or other rights nor the rights of others.