



### PY32T020 的应用 注意事项

#### 前言

PY32T020 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 32Kbytes Flash 和 4Kbytes SRAM 存储器，最高工作频率 48MHz。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 PY32T020 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 | PY32T020 |

## 目录

|                                                             |                       |   |
|-------------------------------------------------------------|-----------------------|---|
| 1                                                           | LSE 配置注意事项 .....      | 3 |
| 2                                                           | SPI 使用注意事项.....       | 3 |
| 3                                                           | PWR 使用注意事项 .....      | 3 |
| 4                                                           | GPIO 配置注意事项 .....     | 3 |
| 5                                                           | IO 倒灌电流使 MCU 工作 ..... | 4 |
| 6                                                           | STOP 模式操作注意事项 .....   | 4 |
| 7                                                           | TK 使用注意事项.....        | 4 |
| 8                                                           | LED 恒流输出配置注意事项.....   | 5 |
| 9                                                           | I2C 从机使用注意事项.....     | 5 |
| 10                                                          | NRST 复位注意事项.....      | 5 |
| 11                                                          | ADC 上电校准.....         | 5 |
| 12                                                          | ADC 使用注意事项.....       | 7 |
| 13                                                          | Flash 使用注意事项 .....    | 7 |
| 14                                                          | RCC 使用注意事项.....       | 7 |
| 15                                                          | 版本历史.....             | 8 |
| 附录 1                                                        | .....                 | 9 |
| 1.PY32T020 读取 information 区域中存放的内部参考电压 1.2V 实测值(具体地址见 12.2) | .....                 | 9 |

## 1 LSE 配置注意事项

- 先配置 LSE\_DRIVER，再使能 LSE(LSEON=1)。

## 2 SPI 使用注意事项

| SPI 模式 | 收/发模式 | SPI 最快速度 |
|--------|-------|----------|
| 从机全双工  | 收     | PCLK/16  |
| 从机全双工  | 发     | PCLK/16  |
| 主机全双工  | 收     | PCLK/2   |
| 主机全双工  | 发     | PCLK/2   |

- SPI 按 8bit 发送数据时，须将 SPI 寄存器强转成 uint8\_t 类型的变量((\_\_IO uint8\_t\*)SPI->DR);
- SPI 从机发送时，SPI 的 DR 寄存器写入一个值后，未发送出去之前，再次写的值不能覆盖前面写的值；如果想覆盖，需要先复位整个 SPI 模块(通过 RCC\_APB1RSTR2 中的 SPI1RST 来进行对应模块的复位)，重新写入 DR 值；
- SPI 半双工主机接收模式，CPHA=0，CPOL=1，256 分频下，会多一个 CLK；
- SPI 通信时，SPI->SR.BSY 位在最后一个时钟期间被清除，Polling 模式时，下一帧数据更新需确保上一帧数据传输完成。

## 3 PWR 使用注意事项

- MCU 进 STOP 模式时必须使能看门狗；
- 推荐使用硬件看门狗替代软件看门狗，提高程序可靠性；
- 一旦使能看门狗，软件无法关闭，在低功耗下，需使用 RTC 定时唤醒，对看门狗进行喂狗。

## 4 GPIO 配置注意事项

- 当 GPIO 用作触摸时，禁止同时配置为通用输出模式及复用功能模式；
- 初始化 GPIO 等其他结构体都需要赋值为 0，避免初始值不固定；

- 所有 GPIO 不能有超过-0.3 V 的负压。

## 5 IO 倒灌电流使 MCU 工作

### 5.1 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。

### 5.2 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。

### 5.3 代码示例

```
int main(void)
{
    LL_IOP_GRP1_EnableClock(LL_IOP_GRP1_PERIPH_GPIOA);/*Enable the GPIOA clock*/
    /*Configure pin PA1 as open-drain output */
    LL_GPIO_SetPinMode(GPIOA, LL_GPIO_PIN_1, LL_GPIO_OUTPUT_OPENDRAIN);
    LL_mDelay(5);                                     /*Delay for 5ms*/
}
```

## 6 STOP 模式操作注意事项

- A 版本芯片推荐做法：进入 STOP 模式前关闭 BOR 功能。（[B 版本已修复](#)）
- A 版本芯片若必须开启 BOR 功能，需要在进入 STOP 模式前开启看门狗，在 STOP 模式下定时唤醒进行喂狗操作。（[B 版本已修复](#)）

## 7 TK 使用注意事项

- 在电源干扰较大或需 EFT 认证的应用，建议使用 PF5 以外 IO 作为 TK。
- 在 PCB 设计时，开关电源储能电感应尽量远离弹簧按键摆放，间距至少 4cm 以上。

## 8 LED 恒流输出配置注意事项

- 使用 PB2 作为恒流输出时，必须首先使能 PA0 的恒流功能。
- 使用 PB3 作为恒流输出时，必须首先使能 PA1 的恒流功能。

## 9 I2C 从机使用注意事项

### 9.1 操作流程：

每当从机收到 STOP 信号后进行如下操作：

- 关闭 I2C 模块使能；
- 打开 I2C 模块使能；
- 应答使能。

### 9.2 代码示例：

```
LL_I2C_Disable(I2C1);  
LL_I2C_Enable(I2C1);  
LL_I2C_AcknowledgeNextData(I2C1, LL_I2C_ACK);
```

## 10 NRST 复位注意事项

- 必须确保 NRST 引脚的低电平有效复位脉冲宽度大于 300  $\mu$ s。

## 11 ADC 上电校准

### 11.1 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。

### 11.2 操作流程

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

### 11.3 代码示例

```
static void APP_AdcConfig()
{
    LL_APB1_GRP2_EnableClock(LL_APB1_GRP2_PERIPH_ADC1);           //使能 ADC1 时钟

    if (LL_ADC_IsEnabled(ADC1) == 0)
    {
        LL_ADC_StartCalibration(ADC1);                             //使能校准
        #if (USE_TIMEOUT == 1)
            Timeout = ADC_CALIBRATION_TIMEOUT_MS;
        #endif
        while (LL_ADC_IsCalibrationOnGoing(ADC1) != 0)
        {
            #if (USE_TIMEOUT == 1)                                  //检测校准是否超时
                if (LL_SYSTICK_IsActiveCounterFlag())
                {
                    if(Timeout-- == 0)
                    {
                        // ... (code omitted) ...
                    }
                }
            #endif
        }
        #endif
        LL_mDelay(1);
    }
}
```

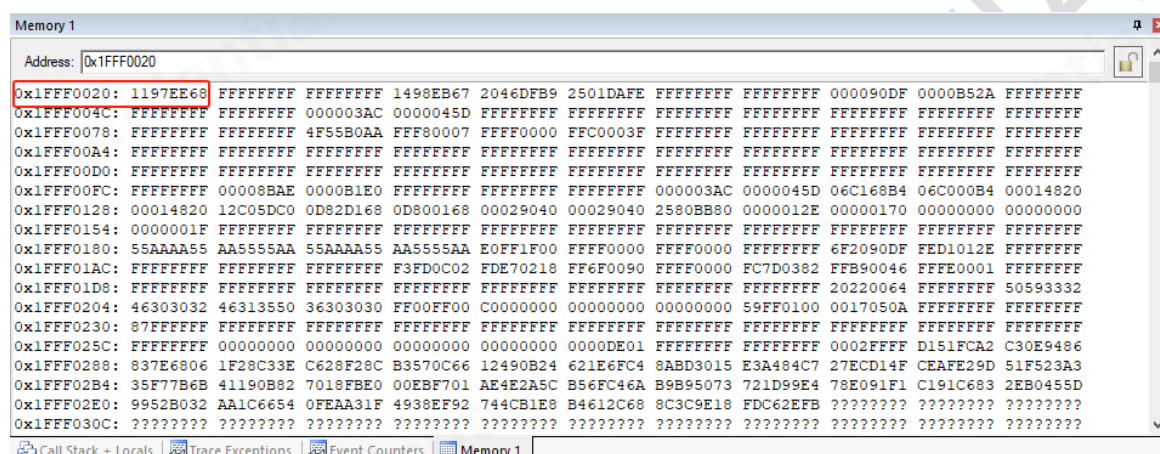
## 12 ADC 使用注意事项

### 12.1 ADC 软件配置

- GPIO 直接驱动大功耗器件会影响 ADC 采样结果(例如数码管显示, 建议数码管显示的时候不采样 ADC, 或者在数码管的各个数据线上串入 10-100  $\Omega$  电阻, 可根据实际情况进行调整)。

### 12.2 Vreferint 1.2V

- 芯片的 Vreferint 1.2 V 实测值放置在 FLASH 中的 information 区域(0x1FFF0020)(高 16 位是实际值, 低 16 位是反码) 读取 Vreferint 1.2 V 的程序见附录 1。



## 13 Flash 使用注意事项

- FLASH 只支持 Page 擦和 Page 写, 一个 Page 是 128 字节, 起始地址只能 Page 对齐(如起始地址 0x08000000, 0x08000080 等)。
- 每次 Page 写之前必须先 Page 擦。

## 14 RCC 使用注意事项

- APB 分频系数大时, 执行 (APB 总线上的) 模块复位后, 不能立马对模块寄存器进行读写操作, 如需操作, 则需要增加\_\_NOP()空指令, 空指令数量要大于 APB 分频数, 例如 APB 8 分频, 建议增加 10 个以上的\_\_NOP 指令。

15 版本历史

| 版本   | 日期         | 更新记录                                |
|------|------------|-------------------------------------|
| V1.0 | 2024.3.20  | 初版                                  |
| V1.1 | 2024.5.11  | 添加 GPIO 配置注意事项                      |
| V1.2 | 2025.04.17 | 增加 STOP、TK、LED、I2C 章内容              |
| V1.3 | 2025.06.04 | 添加 NRST 复位注意事项                      |
| V1.4 | 2025.06.23 | 增加 ADC、IWDG、Flash、RCC、SPI、GPIO 章节内容 |
| V1.5 | 2025.07.10 | 1、合并 SPI 章内容<br>2、添加 STOP 章节内容      |



Puya Semiconductor Co., Ltd.

声 明

普冉半导体(上海)股份有限公司（以下简称：“Puya”）保留更改、纠正、增强、修改 Puya 产品和/或本文档的权利，恕不另行通知。用户可在下单前获取产品的最新相关信息。

Puya 产品是依据订单时的销售条款和条件进行销售的。

用户对 Puya 产品的选择和使用承担全责，同时若用于其自己或指定第三方产品上的，Puya 不提供服务支持且不对此类产品承担任何责任。

Puya 在此不授予任何知识产权的明示或暗示方式许可。

Puya 产品的转售，若其条款与此处规定不一致，Puya 对此类产品的任何保修承诺无效。

任何带有 Puya 或 Puya 标识的图形或字样是普冉的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代并替换先前版本中的信息。

## 附录1

### 1.PY32T020读取information区域中存放的内部参考电压1.2V实测值(具体地址见12.2)

```
#define HAL_VREF_INT          (*(uint8_t*)(0x1fff0023))
#define HAL_VREF_DEC          (*(uint8_t*)(0x1fff0022))
#define vref_int              (*(uint8_t*)(HAL_VREF_INT))    //存放参考电压整数部分
#define vref_dec              (*(uint8_t*)(HAL_VREF_DEC))    //存放参考电压小数部分
float vref;              //参考电压值

static uint8_t Bcd2ToByte(uint8_t Value)
{
    uint32_t tmp = 0U;
    tmp = ((uint8_t)(Value & (uint8_t)0xF0) >> (uint8_t)0x4) * 10U;
    return (tmp + (Value & (uint8_t)0x0F));
}

float read_1_2V(void)
{
    uint8_t data_vref_int,data_vref_dec;
    data_vref_int = Bcd2ToByte(HAL_VREF_INT);
    data_vref_dec = Bcd2ToByte(HAL_VREF_DEC);

    //初始化所有外设, flash接口, systick
    vref = data_vref_int/10;    //计算参考电压
    vref = vref + ((data_vref_int%10)*0.1 + data_vref_dec*0.001);
    return vref;
}
```