

PY32F005 系列各模块的 应用注意事项

前言

PY32F005 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 32 Kbytes Flash 和 3 Kbytes SRAM 存储器，最高工作频率 48 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 PY32F005 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	PY32F005

目录

1	ADC 使用注意事项.....	3
2	FLASH 使用注意事项	3
3	Option 操作	3
4	版本历史.....	6
附录 1	7
1.	PY32F005 读取 information 区域中存放的 Vreferint 1.2V 实测值(具体地址见 1.2)	7

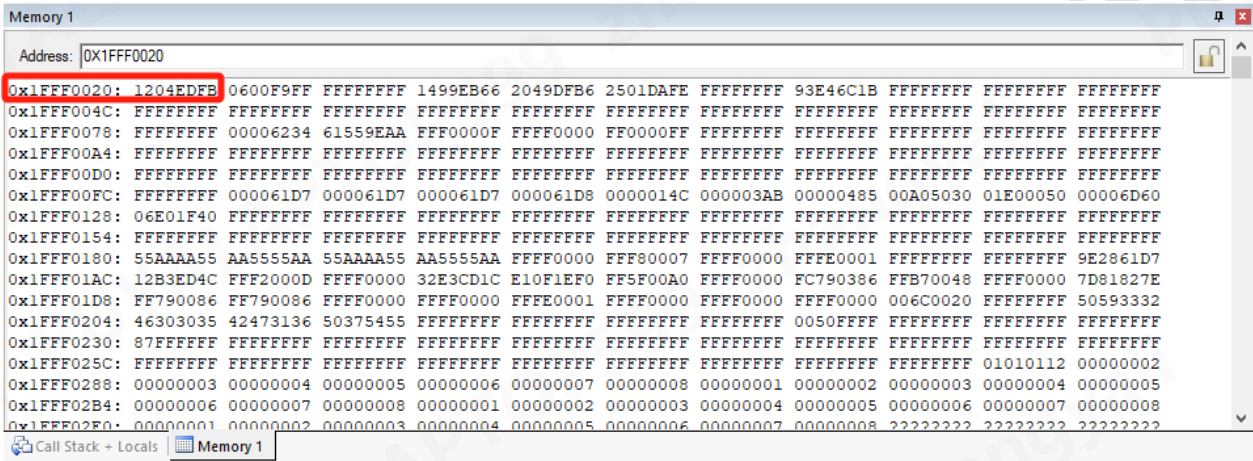
1 ADC 使用注意事项

1.1 ADC 软件配置

- ADC 采集 0V 电压时会有 1~2 个 AD 值的抖动。

1.2 Vreferint 1.2V

- 芯片 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:



- 在采样 Vreferint 1.2 V 的时候，通过 ADC 采样时间转换公式算出来的结果至少需要 20 us，方法如下：
 - a) 降低分辨率；
 - b) 降低ADC的时钟频率；
 - c) 提高ADC采样周期。

2 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 0x08005000, 0x08005080 等)。
- 每次 Page 写之前必须先 Page 擦。

3 Option 操作

- 量产时，Option 操作必须在烧写器选项字节中配置，并把程序中操作 Option 的函数屏蔽；
- 建议客户程序使能写保护，写保护在 Option 中设置，具体步骤如图 3-1、图 3-2 所示；



图 3-1 轩微操作 Option 写保护



图 3-2 创芯工坊操作 Option 写保护

- 烧写器配置 Option 时，需勾选智能复位功能/编程后重启芯片(烧写器均有类似选项需要勾选)，具体步骤如图 3-3、图 3-4 所示。



图 3-3 轩微操作“智能复位”



图 3-4 创芯工坊操作勾选“编程后重启芯片”

4 版本历史

版本	日期	更新记录
V1.0	2026.01.09	初版
V1.1	2026.03.24	修改 ADC 使用注意事项



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。

普冉半导体(上海)股份有限公司 - 保留所有权利

附录1

1. PY32F005读取information区域中存放的Vreferint 1.2V实测值(具体地址见1.2)

```
#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;
}
```