

### 使用 PY32C613 微控制器的 CRC 计算单元进行数据校验

#### 前言

CRC(Cyclic Redundancy Check, 循环冗余校验)计算单元将输入的 32 位数据, 根据生成的多项式, 运算产生一个 CRC 结果。

本应用笔记介绍了 CRC32 的多项式, CRC 的特性以及如何如何进行软件配置等内容。

在本文档中, PY32 仅指表 1 中列出的产品系列。

表 1. 适用产品

类型	产品系列
微型控制器系列	PY32C613

## 目录

1	CRC 简介 .....	错误!未定义书签。	3
1.1	CRC 多项式.....		3
1.2	CRC 特性.....		3
2	CRC 功能描述.....	错误!未定义书签。	4
3	CRC 例程代码.....		5
4	版本历史.....		7

## 1 CRC 简介

### 1.1 CRC 多项式

PY32 微控制器 CRC 使用 CRC32 多项式： $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^8+X^7+X^5+X^4+X^2+X+1$ 。

其中  $X^{32}$  对应 bit 32 为 1,  $X^{26}$  对应 bit 26 为 1, 以此类推。

所以得出一个多项式的值： $(1 \ll 32) | (1 \ll 26) | (1 \ll 23) | \dots | (1 \ll 1) | 1 = 0x104C11DB7$ ,

取低 32 位得出多项式的值：0x4C11DB7。

### 1.2 CRC 特性

- 支持 32 位数据输入
- 单个输入/输出 32 数据和结果输出公用一个寄存器
- General purpose 的 8 位寄存器(可被用作临时存储)
- 计算时间：32 位数据 4 个 AHB 时钟

## 2 CRC 功能描述

CRC 计算单元含有 1 个 32 位数据寄存器：

- 对该寄存器进行写操作时，作为输入寄存器，输入要进行 CRC 计算的数据。
- 对该寄存器进行读操作时，返回上一次 CRC 计算的结果。

每一次写入数据寄存器，其计算结果是前一次 CRC 计算结果和新计算结果的组合（对整个 32 位字进行 CRC 计算，而不是逐字节地计算）。

当 CRC 正在计算时，写操作会被阻止，直到 CRC 计算结束。

可以通过设置寄存器 CRC\_CR 的 RESET 位来重置寄存器 CRC\_CR 为 0xFFFF FFFF。该操作不影响寄存器 CRC\_IDR 内的数据。

### 3 CRC 例程代码

打开 PY32F030\_HAL\_Firmware\_V0.0.1 文件夹，打开 CRC 例程：本例程定义了若干个 32 位数据进行 CRC 校验，最后输出一个校验码与期望的校验码进行比对。若校验错误，则 LED 灭灯，否则 LED 亮灯。

- 在 main.c 中定义了 CRC 数据声明

```
/* CRC handler declaration */
CRC_HandleTypeDef  CrcHandle;

/* Used for storing CRC Value */
__IO uint32_t uwCRCValue = 0;

static const uint32_t aDataBuffer[BUFFER_SIZE] =
{
    0x00001021, 0x20423063, 0x408450a5, 0x60c670e7, 0x9129a14a, 0xb16bc18c,
    0xd1ade1ce, 0xf1ef1231, 0x32732252, 0x52b54294, 0x72f762d6, 0x93398318,
    0xa35ad3bd, 0xc39cf3ff, 0xe3de2462, 0x34430420, 0x64e674c7, 0x44a45485,
    0xa56ab54b, 0x85289509, 0xf5cfc5ac, 0xd58d3653, 0x26721611, 0x063076d7,
    0x569546b4, 0xb75ba77a, 0x97198738, 0xf7dfe7fe, 0xc7bc48c4, 0x58e56886,
    0x78a70840, 0x18612802, 0xc9ccd9ed, 0xe98ef9af, 0x89489969, 0xa90ab92b,
    0x4ad47ab7, 0x6a961a71, 0x0a503a33, 0x2a12dbfd, 0xfbbfeb9e, 0x9b798b58,
    0xbb3bab1a, 0x6ca67c87, 0x5cc52c22, 0x3c030c60, 0x1c41edae, 0xfd8fcddec,
    0xad2abd0b, 0x8d689d49, 0x7e976eb6, 0x5ed54ef4, 0x2e321e51, 0x0e70ff9f,
    0xefbedfdd, 0xcffcbf1b, 0x9f598f78, 0x918881a9, 0xb1caa1eb, 0xd10cc12d,
    0xe16f1080, 0x00a130c2, 0x20e35004, 0x40257046, 0x83b99398, 0xa3fbb3da,
    0xc33dd31c, 0xe37ff35e, 0x129022f3, 0x32d24235, 0x52146277, 0x7256b5ea,
    0x95a88589, 0xf56ee54f, 0xd52cc50d, 0x34e224c3, 0x04817466, 0x64475424,
    0x4405a7db, 0xb7fa8799, 0xe75ff77e, 0xc71dd73c, 0x26d336f2, 0x069116b0,
    0x76764615, 0x5634d94c, 0xc96df90e, 0xe92f99c8, 0xb98aa9ab, 0x58444865,
    0x78066827, 0x18c008e1, 0x28a3cb7d, 0xdb5ceb3f, 0xfb1e8bf9, 0x9bd8abbb,
    0x4a755a54, 0x6a377a16, 0x0af11ad0, 0x2ab33a92, 0xed0fdd6c, 0xcd4dbdaa,
    0xad8b9de8, 0x8dc97c26, 0x5c644c45, 0x3ca22c83, 0x1ce00cc1, 0xef1fff3e,
    0xdf7caf9b, 0xbfba8fd9, 0x9ff86e17, 0x7e364e55, 0x2e933eb2, 0x0ed11ef0
};

/* Expected CRC Value */
uint32_t uwExpectedCRCValue = 0x379E9F06;
```

- 在 main 函数里进行 CRC 初始化

```
/*-1- Init CRC module */
CrcHandle.Instance = CRC;
if (HAL_CRC_Init(&CrcHandle) != HAL_OK)
{
    while (1);
}
```

- 在 main 函数里调用 HAL 库的 CRC 计算函数进行数据计算，返回值即计算结果

```
/*-2- Compute the CRC of "aDataBuffer" */
uwCRCValue = HAL_CRC_Accumulate(&CrcHandle, (uint32_t *)aDataBuffer, BUFFER_SIZE);
```

- 在 main 函数中获取 CRC 计算结果后，与期待结果比对

```
/*-3- Compare the CRC value to the Expected one */
if (uwCRCValue != uwExpectedCRCValue)
{
    /* Wrong CRC value: LED OFF */
    BSP_LED_Off(LED_GREEN);
}
else
{
    /* Right CRC value: Turn LED on */
    BSP_LED_On(LED_GREEN);
}
```

4 版本历史

版本	日期	更新记录
V1.0	2023.11.21	初版



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