

### 使用 PY32F030\_003 微控制器的 LED 控制器模块

#### 前言

LED 数码管是由多个发光二极管封装在一起组成的器件，在生活中随处可见，洗衣机，电饭煲，热水器，微波炉，这些最基本的家用电器基本都用到了这种 LED 数码管。本应用笔记提供了含有配置 LED 数码管的代码例程等内容。

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 | PY32F030、PY32F003 |

## 目录

|   |                   |           |
|---|-------------------|-----------|
| 1 | LED 控制器功能简介 ..... | 错误!未定义书签。 |
| 2 | LED 控制器应用例程 ..... | 错误!未定义书签。 |
| 3 | 版本历史 .....        | 5         |

PUYA CONFIDENTIAL

## 1 LED 控制器功能简介

- 本项目支持 1~4 个 8 段式共阴极 LED 数码管的控制器功能。该控制器通过 4 个支持超大灌电流 (80mA/60mA/40mA/20Ma 可配置)的管脚(PB)，输出对应点亮 4 个 7-segment 数码管，同一时间只点亮一个数字。

PUYA CONFIDENTIAL

## 2 LED 控制器应用例程

- 配置 LED 控制器步骤:

| 步骤 | 操作                            |
|----|-------------------------------|
| 1  | LED 控制器硬件初始化(时钟, GPIO 引脚, 中断) |
| 2  | 初始化 LED 控制器                   |
| 3  | 在中断中控制数码管显示                   |

- LED 控制器代码介绍: 打开 LED\_IT 例程代码, 此样例演示了 LED 的控制数码管显示功能, 样例中同时控制 4 个数码管, 4 个数码管的显示内容可以在中断中实时修改。

1. 打开例程代码, 在 py32f030\_hal\_msp.c 文件中, HAL\_LED\_MspInit 函数初始化了底层硬件驱动, 使能了时钟, 配置了 GPIO 的复用功能, 使能了 LED 中断。

```
void HAL_LED_MspInit (LED_HandleTypeDef *hled)
{
    GPIO_InitTypeDef GPIO_InitStructure;

    __HAL_RCC_LED_CLK_ENABLE();

    __HAL_RCC_GPIOA_CLK_ENABLE();
    //开启 GPIOA 时钟
    __HAL_RCC_GPIOB_CLK_ENABLE();

    //com1 2 3
    GPIO_InitStructure.Pin= GPIO_PIN_3 |GPIO_PIN_4 |GPIO_PIN_5;

    GPIO_InitStructure.Mode=GPIO_MODE_AF_PP;
    GPIO_InitStructure.Pull=GPIO_PULLUP;
    GPIO_InitStructure.Alternate = GPIO_AF6_LED ;
    GPIO_InitStructure.Speed=GPIO_SPEED_FREQ_VERY_HIGH;
    HAL_GPIO_Init(GPIOB,&GPIO_InitStructure);

    //com0
    GPIO_InitStructure.Pin=GPIO_PIN_15;
    GPIO_InitStructure.Mode=GPIO_MODE_AF_PP;
    GPIO_InitStructure.Pull=GPIO_PULLUP;
    GPIO_InitStructure.Alternate = GPIO_AF6_LED;
    GPIO_InitStructure.Speed=GPIO_SPEED_FREQ_VERY_HIGH;
    HAL_GPIO_Init(GPIOA,&GPIO_InitStructure);

    //SEG B C D E F G DP
    GPIO_InitStructure.Pin= GPIO_PIN_0 |GPIO_PIN_1 |GPIO_PIN_2 | //
        GPIO_PIN_3 |GPIO_PIN_4 |GPIO_PIN_5 |GPIO_PIN_6;

    GPIO_InitStructure.Mode=GPIO_MODE_AF_PP;
    GPIO_InitStructure.Pull=GPIO_PULLUP;
    GPIO_InitStructure.Alternate = GPIO_AF3_LED ;
    GPIO_InitStructure.Speed=GPIO_SPEED_FREQ_VERY_HIGH;
    HAL_GPIO_Init(GPIOA,&GPIO_InitStructure);

    //SEG A
    GPIO_InitStructure.Pin=GPIO_PIN_8;
    GPIO_InitStructure.Mode=GPIO_MODE_AF_PP;
    GPIO_InitStructure.Pull=GPIO_PULLUP;
```

```
GPIO_Initure.Alternate = GPIO_AF3_LED ;
GPIO_Initure.Speed=GPIO_SPEED_FREQ_VERY_HIGH;
HAL_GPIO_Init(GPIOB,&GPIO_Initure);
```

```
//使能 LED 中断
HAL_NVIC_SetPriority(LED_IRQn,0,0);
HAL_NVIC_EnableIRQ(LED_IRQn);
__HAL_LED_ENABLE_IT(hled, LED_CR_IE);
}
```

2. 在 main.c 中初始化 LED 控制器，配置数码管个数，LED 时钟分频系数等内容。

```
hled.Instance = LED;
hled.Init.Com_sel = LED_COM_SEL_COM0_1_2_3;
hled.Init.EHS = LED_COM_DRIVER_LOW;
hled.Init.PR = 9; /* Fpclk/(PR+1) */
hled.Init.TR_T1 = 0xF0;
hled.Init.TR_T2 = 0x10;
HAL_LED_Init(&hled);
```

3. LED 中断发生后，代码运行到中断回调函数 HAL\_LED\_LightComplateCallback 中，我们在这里显示数码管的数值。

```
void HAL_LED_LightComplateCallback(LED_HandleTypeDef *hled)
{
    LED->DR0 = dispArr[(dispNum)%10];
    LED->DR1 = dispArr[(dispNum+1)%10];
    LED->DR2 = dispArr[(dispNum+2)%10];
    LED->DR3 = dispArr[(dispNum+3)%10];
}
```

3 版本历史

| 版本   | 日期         | 更新记录       |
|------|------------|------------|
| V0.1 | 2021.10.20 | 初版         |
| V1.0 | 2022.06.25 | 修改了应用例程的内容 |
| V1.1 | 2023.08.16 | 增加 002A 内容 |
| V1.2 | 2023.08.24 | 更新声明       |
|      |            |            |
|      |            |            |
|      |            |            |



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。