

# 一起学习 CC3200 系列教程之跑马灯

阿汤哥

序:

能力有限, 英语不怎么好, 难免有错, 有问题请联系我,

QQ1519256298 [hytga@163.com](mailto:hytga@163.com)

实现跑马灯实验需要对 CC3200 设置, 分成 4 个步骤

- 1、引脚映射成 GPIO 模式, 上拉下拉等
- 2、使能 GPIO 的时钟
- 3、配置 GPIO: 输入输出,
- 4、向 GPIO 的端口写入数据

CC3200 共有 64 引脚, 共 32 个 GPIO, 分成 4 组, 每组有 8 个,

所以 CC3200 的 1 个引脚有 3 中名称,

一种是基于引脚的, 编号从 0-63

一种是是基于总的 GPIO, 编号从 0-31

一种是基于分组的 GPIO, 编号 A0-A3 (或者 A B C D) 加 (0-7)

引脚 64 的对应关系 : PIN\_64 == GPIO09 == GPIO A1\_1

- 1、设置引脚的映射,

库函数示例 PinModeSet(PIN\_64, PIN\_MODE\_0);

PIN\_64 就是引脚编号, 这里是引脚 64

PIN\_MODE\_0 就是映射的功能, 这里是 GPIO 模式

设置引脚的电流强度、上拉、下拉、开漏等

PinConfigSet(PIN\_64, PIN\_STRENGTH\_2MA, PIN\_TYPE\_STD);

PIN\_64 就是引脚编号, 这里是引脚 64

PIN\_STRENGTH\_2MA: 电流强度, 2ma, 记得 CC3200 的电流强度大部分都是 2ma, 好像只有几根是 4ma 的,

PIN\_TYPE\_STD: 没有上拉和下拉, 没有开漏

- 2、使能 GPIO 的时钟

CC3200 的 GPIO 分成 4 组, 每组都有自己的一根时钟, 所以配置时需要查阅 datasheet, 引脚是对应于哪一个 GPIO, 这个 GPIO 又是哪一组。

库函数示例: PRCMPeripheralClkEnable(PRCM\_GPIOA1, PRCM\_RUN\_MODE\_CLK);

PRCM\_GPIOA1: GPIOA2 的时钟线

PRCM\_RUN\_MODE\_CLK: 当 cc3200 在运行状态时, 使能这个时钟, CC3200 好像有 3 种运行模式, 运行, 睡眠, 深度睡眠, 这里是运行状态

- 3、设置 GPIO 的输出输出

GPIODirModeSet(GPIOA1\_BASE, GPIO\_PIN\_1, GPIO\_DIR\_MODE\_OUT);

GPIOA1\_BASE: GPIO A1 组的 GPIO

GPIO\_PIN\_1: GPIO A1 的第 1 个 GPIO

GPIO\_DIR\_MODE\_OUT: 输出模式

#### 4、向 GPIO 的端口写入数据

```
GPIOPinWrite(GPIOA1_BASE,GPIO_PIN_1,GPIO_PIN_1);
```

GPIOA1\_BASE:GPIO A1 组的 GPIO

参数的第 1 个 GPIO\_PIN\_1: GPIO A1 的第 1 个 GPIO

参数的第 2 个 GPIO\_PIN\_1: GPIO A1\_1 输出高电平, 为什么这里也是 GPIO\_PIN\_1,

其实这里我是取巧了, 看一下这个函数的原型

```
GPIOPinWrite(unsigned long ulPort, unsigned char ucPins, unsigned char ucVal);
```

ucPins 可以等于 GPIO\_PIN\_1 | GPIO\_PIN\_2,

ucVal 可以是任意值, 这里我用一段代码表示

```
if(ucPins& GPIO_PIN_1) {
```

```
    if (ucVal & GPIO_PIN_1) {
```

```
        GPIO_PIN_1 输出高
```

```
    } else {
```

```
        GPIO_PIN_1 输出低
```

```
    }
```

```
}
```

```
If(ucPins& GPIO_PIN_1 == 0) {
```

```
    GPIO_PIN_1 不受影响
```

```
}
```

其他的 GPIO 依次类推

代码

```
//#define BLINKY_MODE 01
```

```
//#define BLINKY_MODE 02
```

```
#define BLINKY_MODE 03
```

```
void main(void) {
```

//初始化板子, 我看了一下大概是中断向量表的映射, 和其他的一些东西, 这里我们不关心

```
BoardInit();
```

```
//使能时钟
```

```
PRCMPeripheralClkEnable(PRCM_GPIOA1,PRCM_RUN_MODE_CLK);
```

```
//设置成 GPIO 模式
```

```
PinModeSet(PIN_64, PIN_MODE_0);
```

```
//设置电流驱动能力 2ma, 普通
```

```
PinConfigSet(PIN_64,PIN_STRENGTH_2MA,PIN_TYPE_STD);
```

```
//设置成 GPIO 模式
```

```
PinModeSet(PIN_01, PIN_MODE_0);
```

```
//设置电流驱动能力 2ma, 普通
```

```
PinConfigSet(PIN_01,PIN_STRENGTH_2MA,PIN_TYPE_STD);
```

```

//设置成 GPIO 模式
PinModeSet(PIN_02, PIN_MODE_0);
//设置电流驱动能力 2ma, 普通
PinConfigSet(PIN_02,PIN_STRENGTH_2MA,PIN_TYPE_STD);

GPIODirModeSet(GPIOA1_BASE,GPIO_PIN_1,GPIO_DIR_MODE_OUT);
GPIODirModeSet(GPIOA1_BASE,GPIO_PIN_2,GPIO_DIR_MODE_OUT);
GPIODirModeSet(GPIOA1_BASE,GPIO_PIN_3,GPIO_DIR_MODE_OUT);
while(1) {
    #if BLINKY_MODE == 01 //现象:跑马灯
        GPIOPinWrite(GPIOA1_BASE,GPIO_PIN_1,GPIO_PIN_1);
        delay(0xfffff);
        GPIOPinWrite(GPIOA1_BASE,GPIO_PIN_1,0);
        delay(0xfffff);
        GPIOPinWrite(GPIOA1_BASE,GPIO_PIN_2,GPIO_PIN_2);
        delay(0xfffff);
        GPIOPinWrite(GPIOA1_BASE,GPIO_PIN_2,0);
        delay(0xfffff);
        GPIOPinWrite(GPIOA1_BASE,GPIO_PIN_3,GPIO_PIN_3);
        delay(0xfffff);
        GPIOPinWrite(GPIOA1_BASE,GPIO_PIN_3,0);
        delay(0xfffff);
    #endif
    #if BLINKY_MODE == 02 //现象全亮后全灭

        GPIOPinWrite(GPIOA1_BASE,GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3,GPIO_PIN_1|GPIO
_PIN_2|GPIO_PIN_3);
        delay(0xfffff);
        GPIOPinWrite(GPIOA1_BASE,GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3,0);
        delay(0xfffff);

    #endif
    #if BLINKY_MODE == 03//现象 2 个亮一个灭 后 2 个灭 1 个亮

        GPIOPinWrite(GPIOA1_BASE,GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3,GPIO_PIN_1|GPIO
_PIN_2&(~GPIO_PIN_3));
        delay(0xfffff);

        GPIOPinWrite(GPIOA1_BASE,GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3,(~GPIO_PIN_1)&(~
GPIO_PIN_2)|GPIO_PIN_3);
        delay(0xfffff);

    #endif
}

```

}  
}