GPIO 教程

——疯壳·开发板系列

Wolverine-Team

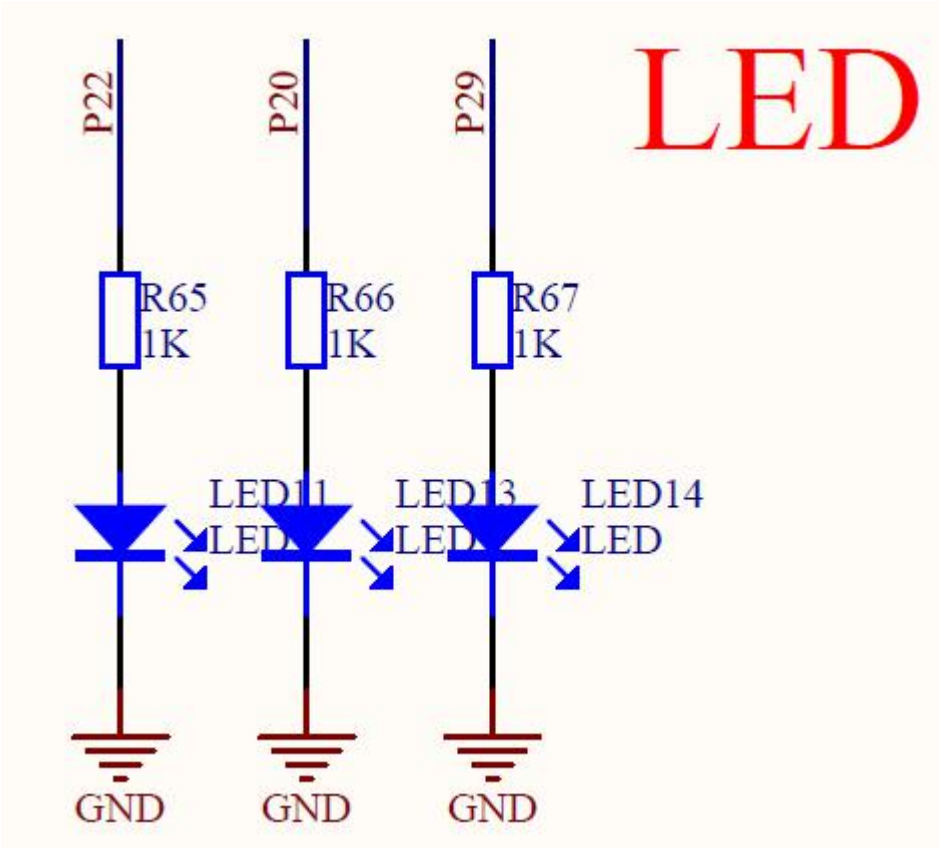
2015/7/16

目录

官网地址：http://www.fengke.club
购买链接：http://shop115904315.taobao.com/
官方 QQ 群：193836402

第一节 LED 硬件电路

手环有三个可控 LED，分别接在 MCU 的 P22、P20、P29 这三个引脚，每个 LED 串联 1K 的限流电阻，如下图所示：



第二节 GPIO 寄存器

2.1 GPIO 引脚介绍

DA14580 的 I/O 引脚功能可以通过软件配置，分为 4 组，分别为 Port0、Port1、Port2、Port3，其中 Port2 只在 QFN40 与 QFN48 封装的芯片中，Port3 只在 QFN48 封装的芯片中。

Port0 有 8 个引脚，Port1 有 6 个引脚（其中包括 DEBUG 引脚 SW_CLK 与 SWDIO），Port2 有 10 个引脚，Port3 有 8 个引脚；

每个引脚都可以选择上拉或者下拉 25KOhm 的电阻；

每个引脚上拉电压在 VBAT3V(降压模式)与 VBAT1V(升压模式)两者可选；

4 路模数转换的引脚固定分配为 Port0 中的 0:3 引脚；

当系统进入睡眠模式时，引脚保持最后的状态。

2.2 寄存器介绍

2.2.1 P0 数据寄存器

Table 223: P0_DATA_REG (0x50003000)

Bit	Mode	Symbol	Description	Reset
15:8	-	-	Reserved	0x0
7:0	R/W	P0_DATA	Set P0 output register when written; Returns the value of P0 port when read	0x0

15:8 位：保留不使用；

7:0 位：写该寄存器则设置 P0 输出寄存器的值，读该寄存器则返回 P0 口的值。

2.2.2 P0 设置数据寄存器

Table 224: P0_SET_DATA_REG (0x50003002)

Bit	Mode	Symbol	Description	Reset
15:8	-	-	Reserved	0x0
7:0	R/W	P0_SET	Writing a 1 to P0[y] sets P0[y] to 1. Writing 0 is discarded; Reading returns 0	0x0

15:8 位：保留不使用；

7:0 位：写’1’到对应位则对应的引脚置’1’，写’0’无效，读该寄存器则返回 0。

2.2.3 P0 复位数据寄存器

Table 225: P0_RESET_DATA_REG (0x50003004)

Bit	Mode	Symbol	Description	Reset
15:8	-	-	Reserved	0x0
7:0	R/W	P0_RESET	Writing a 1 to P0[y] sets P0[y] to 0. Writing 0 is discarded; Reading returns 0	0x0

15:8 位：保留不使用；

7:0 位：写’1’到对应位则对应的引脚置’0’，写’0’无效，读该寄存器则返回 0。

2.2.4 P00 模式寄存器

Table 226: P00_MODE_REG (0x50003006)

Bit	Mode	Symbol	Description	Reset
15:10	-	-	Reserved	0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, Pull-down selected 11 = Output, no resistors selected In ADC mode, these bits are don't care	0x2
7:5	-	-	Reserved	0x0
4:0	R/W	PID	Function of port 0 = Port function, PUPD as set above 1 = UART1_RX 2 = UART1_TX 3 = UART2_RX 4 = UART2_TX 5 = SPI_DI 6 = SPI_DO 7 = SPI_CLK 8 = SPI_EN 9 = I2C_SCL 10 = I2C_SDA 11 = UART1_IRDA_RX 12 = UART1_IRDA_TX 13 = UART2_IRDA_RX 14 = UART2_IRDA_TX 15 = ADC (only for P0[3:0]) 16 = PWM0 17 = PWM1 18 = BLE_DIAG (only for P0[7:0]) 19 = UART1_CTSN 20 = UART1_RTSN 21 = UART2_CTSN 22 = UART2_RTSN 23 = PWM2 24 = PWM3 25 = PWM4 Note: when a certain input function (like SPI_DI) is selected on more than 1 port pin, the port with the lowest index has the highest priority and P0 has higher priority than P1.	0x0

15:10 位：保留不使用；

9:8 位：设置引脚的上拉、下拉电阻的模式，00 为输入，无电阻；01 为输入，上拉电阻；10 为输入，下拉电阻；11 为输出，无电阻；在 ADC 模式中该位的设置无效。

7:5 位：保留不使用；

4:0 位：设置引脚功能，0~25 分别对应不同的功能，具体看上图，注：当一个输入功能设置为多个引脚时，引脚序号越低优先级越高，即优先选用序号低的引脚设置为该功能。P0 的优先级比 P1 的高。

2.2.5 P1~P3 的寄存器配置

参照 P0 寄存器的配置，需要注意的是 P14 与 P15 两个引脚模式的默认值是 1，即默认上拉电阻。

注：P14 与 P15 为 SWD 调试接口的时钟与数据接口，在使用 Jlink 调试过程中不要使用这两个引脚。

2.3 寄存器配置讲解

```
#define P0_DATA_REG              (* ( volatile uint16*)0x50003000)
#define P0_SET_DATA_REG          (* ( volatile uint16*)0x50003002)
#define P0_RESET_DATA_REG        (* ( volatile uint16*)0x50003004)
```

```
#define P00_MODE_REG                    (* ( volatile uint16*)0x50003006)
#define P01_MODE_REG                    (* ( volatile uint16*)0x50003008)
#define P02_MODE_REG                    (* ( volatile uint16*)0x5000300A)
#define P03_MODE_REG                    (* ( volatile uint16*)0x5000300C)
#define P04_MODE_REG                    (* ( volatile uint16*)0x5000300E)
#define P05_MODE_REG                    (* ( volatile uint16*)0x50003010)
#define P06_MODE_REG                    (* ( volatile uint16*)0x50003012)
#define P07_MODE_REG                    (* ( volatile uint16*)0x50003014)
```

如果对 P0 整体操作，可以使用 P0_DATA_REG 寄存器，若使 P0[7:0] = 0xaa，因为以上宏定义是定义的地址，则寄存器操作为 P0_DATA_REG=0xaa;

如果要读取 P0 的状态值，则通过读取 P0_DATA_REG 寄存器，P0_STATUS = P0_DATA_REG，若要读取某一位的状态值，则通过与操作之后返回逻辑值即可。

作为输出时 P0_DATA_REG 适合对整体 P0 口操作，而对于某一位进行操作则需要经过一些逻辑操作保持其它位的值不变，不是很方便。而 P0_SET_DATA_REG 和 P0_RESET_DATA_REG 这两个寄存器对位操作就十分方便。对某一位置'0'或置'1'只需对应位赋 1，其它位赋 0 即可，因为该寄存器忽视写 0 操作。若使 P00 = 1，P01 = 0 则 P0_SET_DATA_REG =0x01;　P0_RESET _DATA_REG=0x02;

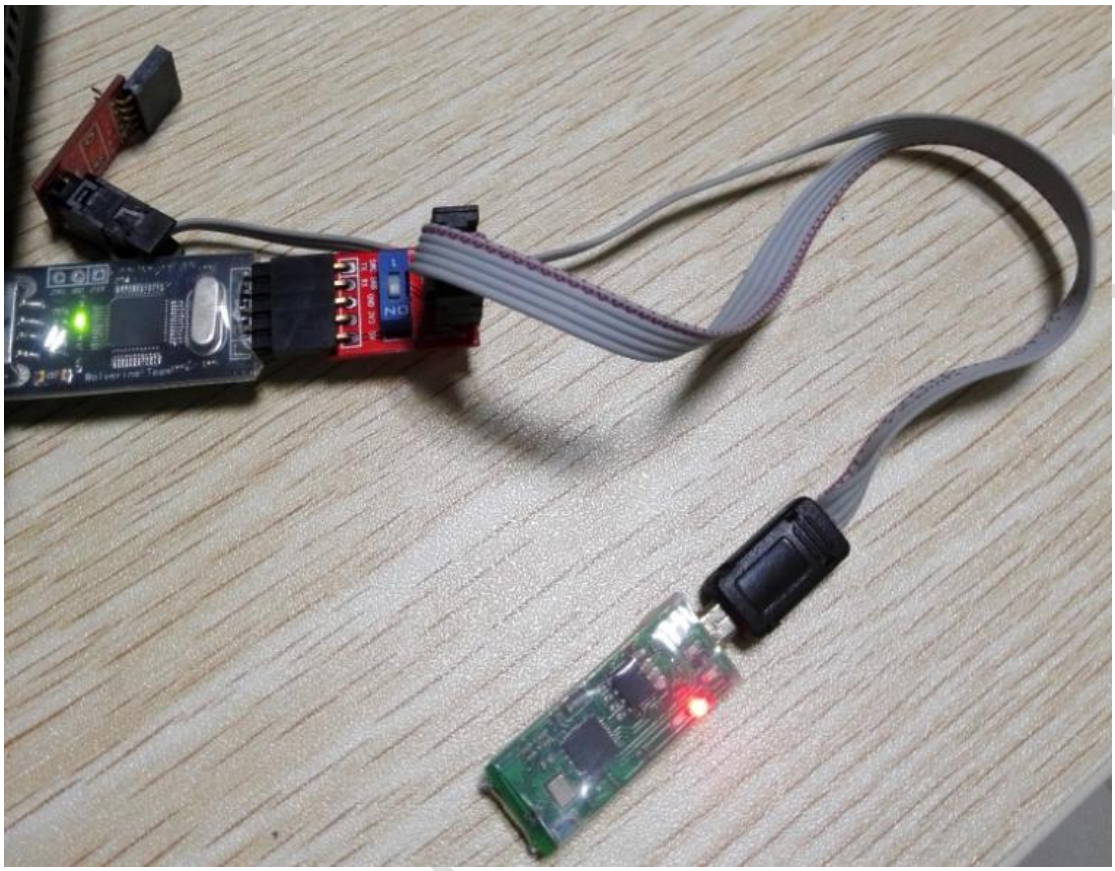
P0[x]_MODE_REG 是 P0 口某一位的操作模式的设置寄存器，例如配置 P00 为输入口，不上拉电阻，P01 为 I2C_SDA，P02 为输出口，则

```
P00_ MODE _DATA_REG=0x00;
P01_ MODE _DATA_REG=0x0a;
P02_ MODE _DATA_REG=0x30;
```

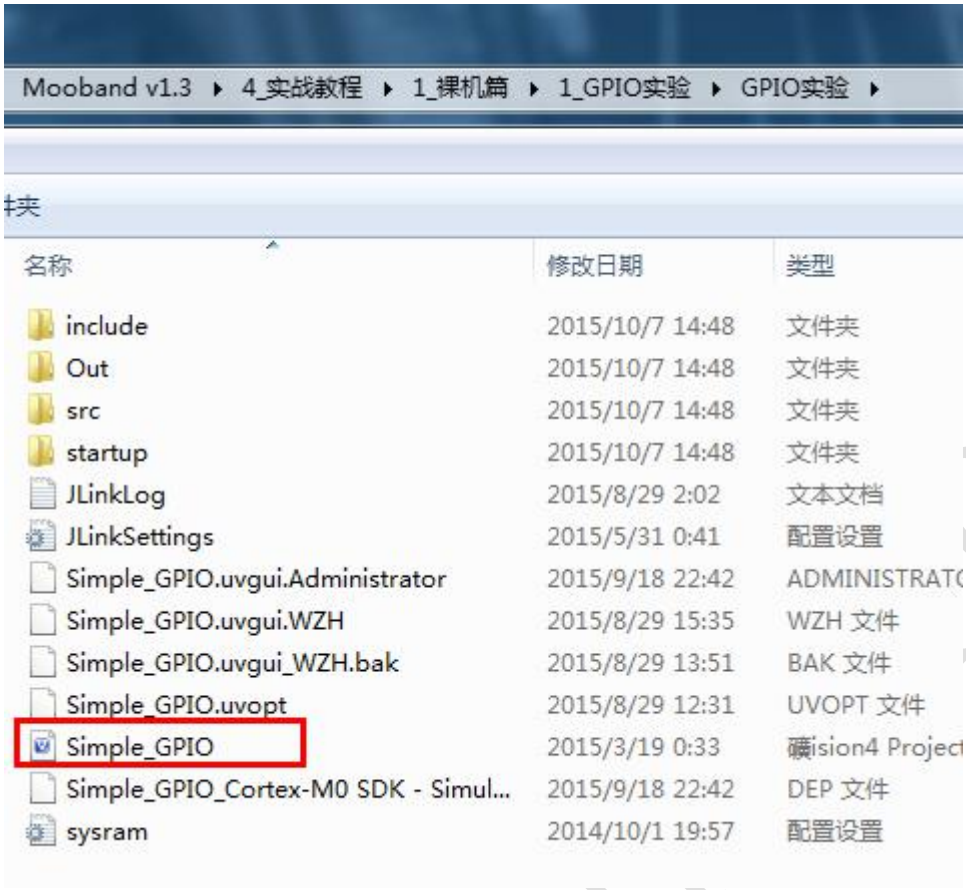
第三节 GPIO 实验

实验需要使用的模块有：手环，Jlink 调试工具，一根手环下载调试线。

将 JLINK 通过下载调试线连接到手环的 USB 调试接口，JLINK 插在有拨码开关的一端，注意丝印标注一一对应，将 JLINK 插上电脑的 USB 口，如下图所示：



打开 GPIO 实验的 Keil 工程 Simple_GPIO.uvproj，位于目录：..\4_实战教程\1_裸机篇\1_GPIO 实验\GPIO 实验。如下图所示：



使用手环调试下载线连接好 Jlink，编译，点击 DEBUG，然后点击全速运行，则可以看到三个 LED 灯交替闪烁的流水灯，如下图所示：