
Timer 教程

——疯壳·开发板系列

Wolverine-Team

2015/7/24

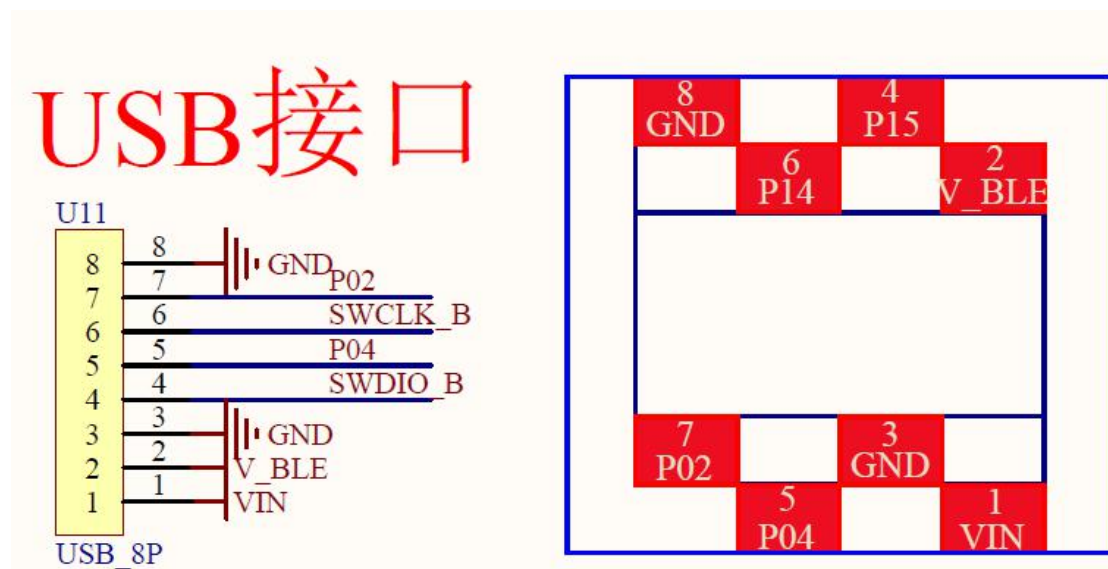
目录

第一节 硬件电路.....	3
第二节 Timer 寄存器.....	4
2.1 Timer 介绍.....	4
2.1.1 Timer0.....	4
2.1.2 Timer2.....	4
2.2 寄存器介绍.....	4
2.2.1 定时器 0 控制寄存器.....	4
2.2.2 定时器 0 计数控制寄存器.....	4
2.2.3 定时器 0 高电平重新装载值.....	4
2.2.4 定时器 0 低电平重新装载值.....	5
2.2.5 PWM2 占空比.....	5
2.2.6 PWM3 占空比.....	5
2.2.7 PWM4 占空比.....	5
2.2.8 PWM2,3,4 的频率.....	5
2.2.9 PWM2,3,4 控制寄存器.....	5
2.3 寄存器配置讲解.....	5
第三节 Timer 实验.....	7

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

第一节 硬件电路

定时器 Timer 为 MCU 内部模块资源，该实验中使用到串口，串口通过 USB 接口引出，分别连接 MCU 的 P02 与 P04，如下图所示：



第二节 Timer 寄存器

2.1 Timer 介绍

软件定时器模块包含两个定时器模块，它们可以通过软件控制、编程并用于各种任务。

2.1.1 Timer0

定时器 0 有 16 位的通用定时器；可以产生两路脉宽调制信号；可编程的输出频率；可编程的占空比；可编程的软件中断。

2.1.2 Timer2

定时器 2 有 14 位的通用定时器；可以产生 3 路脉宽调制信号；输入时钟频率为 16MHz；可编程的输入频率；占空比可调；用于白色 LED 的亮度控制。

2.2 寄存器介绍

2.2.1 定时器 0 控制寄存器

Table 287: TIMER0_CTRL_REG (0x50003400)

Bit	Mode	Symbol	Description	Reset
15:4	-	-	Reserved	0x0
3	R/W	PWM_MODE	0 = PWM signals are '1' during high time. 1 = PWM signals send out the (fast) clock divided by 2 during high time. So it will be in the range of 1 to 8 MHz.	0x0
2	R/W	TIM0_CLK_DIV	1 = Timer0 uses selected clock frequency as is. 0 = Timer0 uses selected clock frequency divided by 10. Note that this applies only to the ON-counter.	0x0
1	R/W	TIM0_CLK_SEL	1 = Timer0 uses 16, 8, 4 or 2 MHz (fast) clock frequency. 0 = Timer0 uses 32 kHz (slow) clock frequency.	0x0
0	R/W	TIM0_CTRL	0 = Timer0 is off and in reset state. 1 = Timer0 is running.	0x0

15:4 位：保留不使用；

3 位：PWM 模式选择，'0'表示 PWM 信号为高时输出'1'，'1'表示 PWM 信号为高时输出快时钟信号的二分频，则输出频率范围为 1~8MHz；

2 位：定时器 0 的时钟分频，为'1'使用选择的时钟源频率，为'0'使用选择的时钟源进行 10 分频，注意这个值适用于计数寄存器；

1 位：时钟源选择，'1'使用快时钟（16、8、4、2MHz），'0'使用慢时钟（32KHz）；

0 位：定时器 0 控制位，'0'表示定时器 0 关闭并处于复位状态，'1'定时器 0 运行。

2.2.2 定时器 0 计数控制寄存器

Table 288: TIMER0_ON_REG (0x50003402)

Bit	Mode	Symbol	Description	Reset
15:0	R0/W	TIM0_ON	Timer0 On reload value: If read the actual counter value ON_CNTER is returned	0x0

定时器 0 计数重新装载值，如果读取该寄存器则返回当前计数器的值。

2.2.3 定时器 0 高电平重新装载值

Table 289: TIMER0_RELOAD_M_REG (0x50003404)

Bit	Mode	Symbol	Description	Reset
15:0	R0/W	TIM0_M	Timer0 'high' reload value if read the actual counter value T0_CNTER is returned	0x0

定时器 0 高电平重新装载值，如果读取该寄存器则返回定时器 0 的计数值。

2.2.4 定时器 0 低电平重新装载值

Table 290: TIMER0_RELOAD_N_REG (0x50003406)

Bit	Mode	Symbol	Description	Reset
15:0	R0W	TIM0_N	Timer0 'low' reload value: If read the actual counter value T0_CNTER is returned	0x0

定时器 0 低电平重新装载值，如果读取该寄存器则返回定时器 0 的计数值。

2.2.5 PWM2 占空比

Table 291: PWM2_DUTY_CYCLE (0x50003408)

Bit	Mode	Symbol	Description	Reset
13:0	RW	DUTY_CYCLE	duty cycle for PWM	0x0

PWM2 信号的占空比。

2.2.6 PWM3 占空比

Table 292: PWM3_DUTY_CYCLE (0x5000340A)

Bit	Mode	Symbol	Description	Reset
13:0	RW	DUTY_CYCLE	duty cycle for PWM	0x0

PWM3 信号的占空比。

2.2.7 PWM4 占空比

Table 293: PWM4_DUTY_CYCLE (0x5000340C)

Bit	Mode	Symbol	Description	Reset
13:0	RW	DUTY_CYCLE	duty cycle for PWM	0x0

PWM4 信号的占空比。

2.2.8 PWM2,3,4 的频率

Table 294: TRIPLE_PWM_FREQUENCY (0x5000340E)

Bit	Mode	Symbol	Description	Reset
13:0	RW	FREQ	Freq for PWM 2 3 4	0x0

PWM2、3、4 信号的频率，三个信号频率相同。

2.2.9 PWM2,3,4 控制寄存器

Table 295: TRIPLE_PWM_CTRL_REG (0x50003410)

Bit	Mode	Symbol	Description	Reset
2	RW	HW_PAUSE_EN	'1' = HW can pause PWM 2,3,4	0x1
1	RW	SW_PAUSE_EN	'1' = PWM 2 3 4 is paused	0x0
0	RW	TRIPLE_PWM_ENABLE	'1' = PWM 2 3 4 is enabled	0x0

2 位：硬件暂停使能，为'1'则硬件可以停止 PWM2、3、4。

1 位：软件暂停使能，为'1'则 PWM2、3、4 暂停。

0 位：PWM 信号使能，为'1'则 PWM2、3、4 使能。

2.3 寄存器配置讲解

```
#define CLK_PER_REG          (* (volatile uint16*)0x50000004)
#define TIMER0_CTRL_REG     (* (volatile uint16*)0x50003400)
#define TIMER0_ON_REG       (* (volatile uint16*)0x50003402)
#define TIMER0_RELOAD_M_REG (* (volatile uint16*)0x50003404)
#define TIMER0_RELOAD_N_REG (* (volatile uint16*)0x50003406)
```

```
#define PWM2_DUTY_CYCLE      (* (volatile uint16*)0x50003408)
#define PWM3_DUTY_CYCLE      (* (volatile uint16*)0x5000340A)
#define PWM4_DUTY_CYCLE      (* (volatile uint16*)0x5000340C)
#define TRIPLE_PWM_FREQUENCY (* (volatile uint16*)0x5000340E)
#define TRIPLE_PWM_CTRL_REG  (* (volatile uint16*)0x50003410)
启动 Timer0 模块的时钟: CLK_PER_REG |= 0x0008;
定时器 0 初始化, 选择高频时钟, 正常 PWM 模式, 无分频, 则
TIMER0_CTRL_REG = 0X0E;
设置计数值为 1000, 高电平为 500, 低电平为 200, 则 TIMER0_ON_REG=1000;
TIMER0_RELOAD_M_REG=500; TIMER0_RELOAD_N_REG=200;
开 定 时 器 中 断  NVIC_SetPriority (SWTIM_IRQn, 254);
NVIC_EnableIRQ(SWTIM_IRQn);
启动定时器 TIMER0_CTRL_REG = (TIMER0_CTRL_REG|0x01);
```

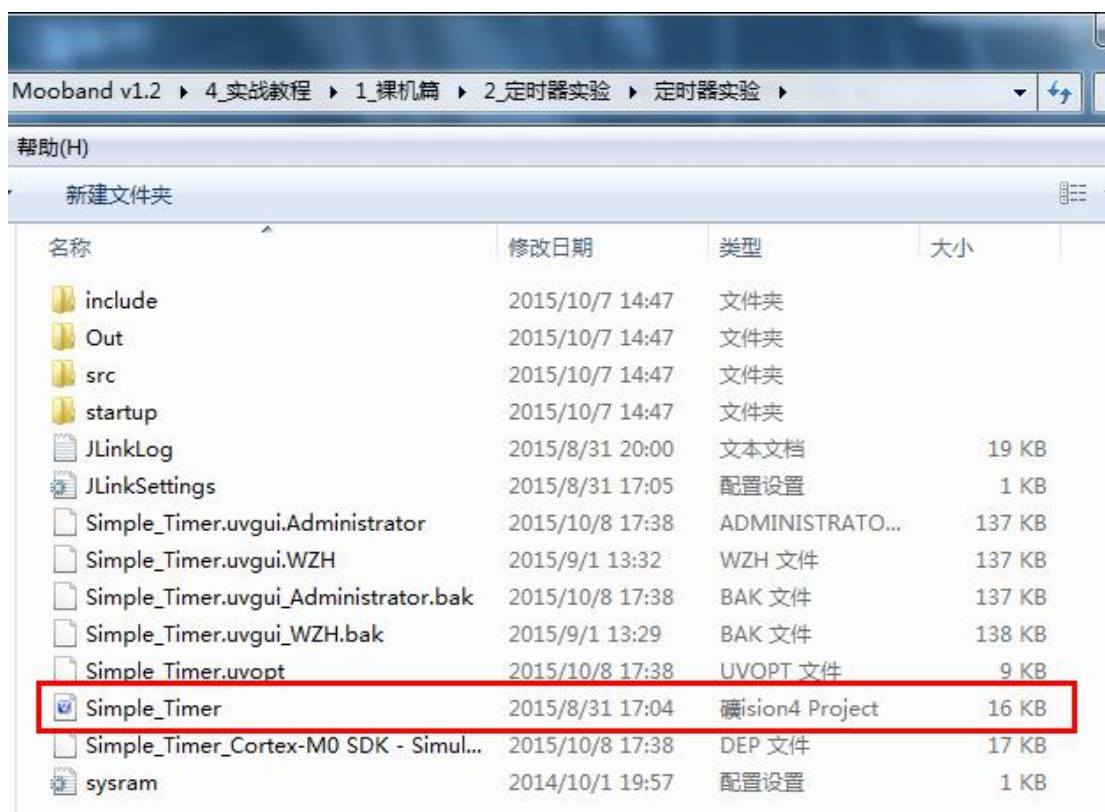
第三节 Timer 实验

实验需要使用的模块有：带屏手环，Jlink 调试工具，USB 转串模块，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。将 USB 转串模块插在手环现在调试线的另一端，注意丝印标注一一对应，然后将 USB 转串模块插在电脑的 USB 接口。如下图所示：



打开 Timer 实验的 Keil 工程 Simple_Timer.uvproj，位于目录：..\4_实战教程\1_裸机篇\2_定时器实验\定时器实验，如下图所示：



打开串口调试助手连接串口，波特率为 115200。打开 KEIL 工程之后，编译代码，点击 DEBUG，然后点击全速运行，就可以看到串口调试助手打印出的信息。如下图所示：



```
*****  
***** 金刚狼团队 *****  
***** Mooband v1.2 *****  
***** 官网:www.mooband.net *****  
***** 淘宝:shop115904315.taobao.com *****  
***** 官方Q Q群: 193836402 *****  
*****  
***** Timer TEST *****  
*****  
Timer Test!  
Timer Test!  
Timer Test!
```

Wolverine