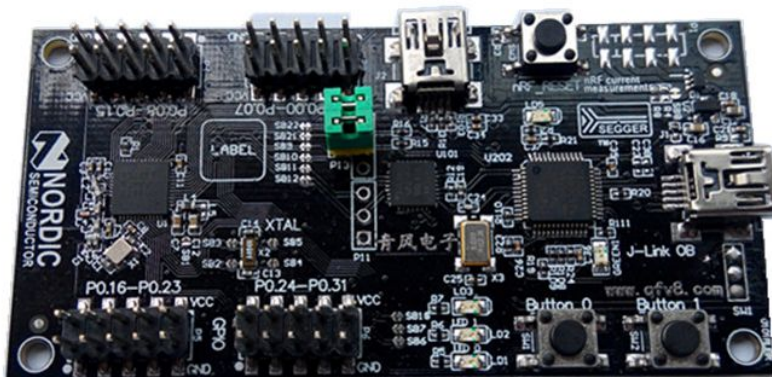


青风带你玩蓝牙 nRF51822 系列教程

-----作者: 青风

出品论坛: www.qfv8.com 青风电子社区

nrf51822蓝牙4.0开发板



青风出品

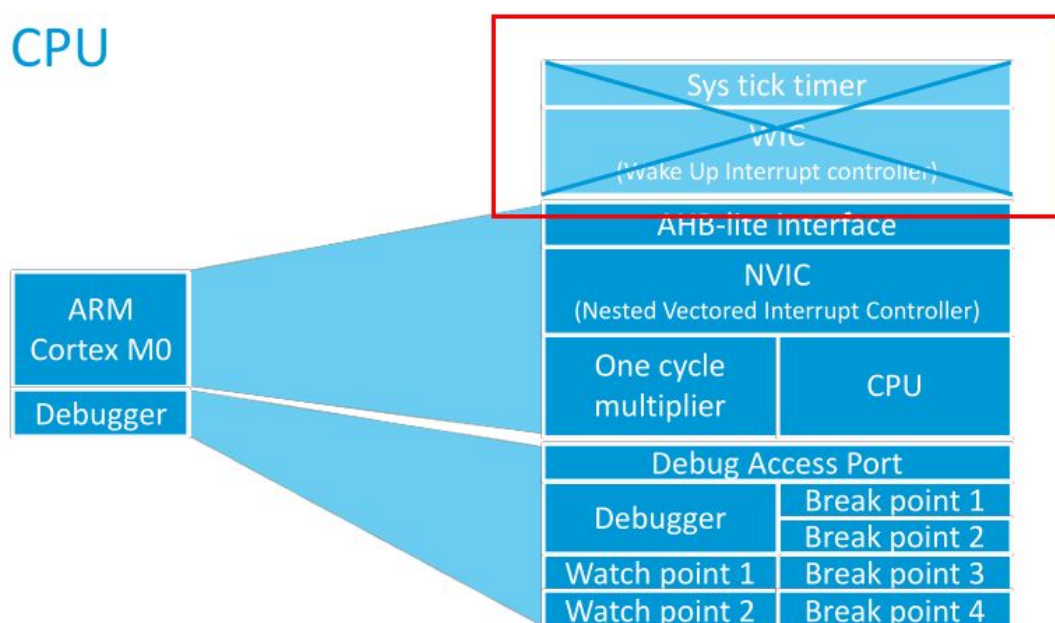


作者: 青风**出品论坛: www.qfv8.com****淘宝店: <http://qfv5.taobao.com>****QQ 技术群: 346518370****硬件平台: 青云 QY-nRF51822 开发板**

2.3 定时器定时

nRF51822 虽然是 cortex m0 内核, 但是为了节省内部资源, 芯片把唤醒中断和滴答时钟 systick 给省略掉了, 如下图所示。那么为了精确定时, 我们就直接采用定时器进行定时了。

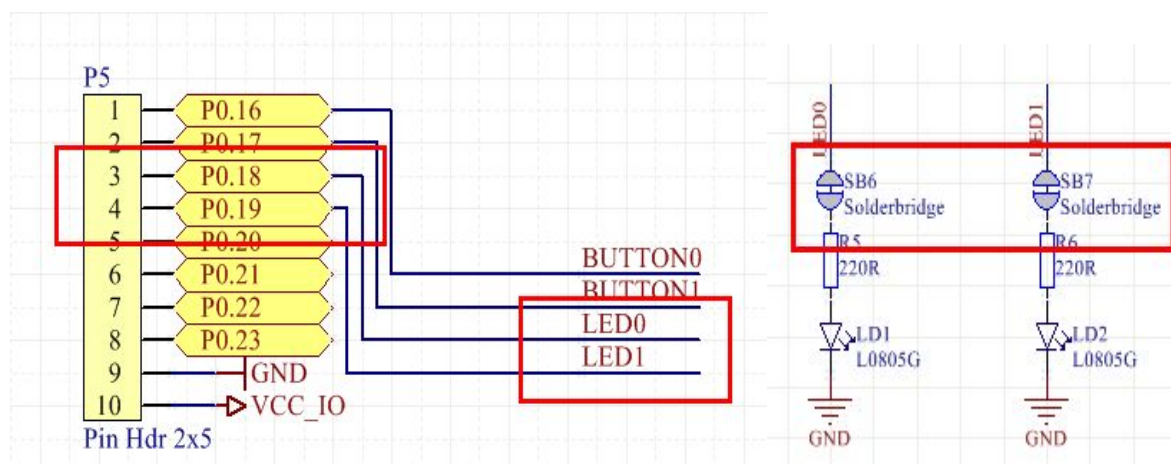
The CPU



在使用 nRF51822 时, 你会发现现在 startup 文件里有设置 systick 中断声明, 但是本人编写了 systick 函数证明 nRF51822 的系统滴答无法开启, 这里特此提示一下。

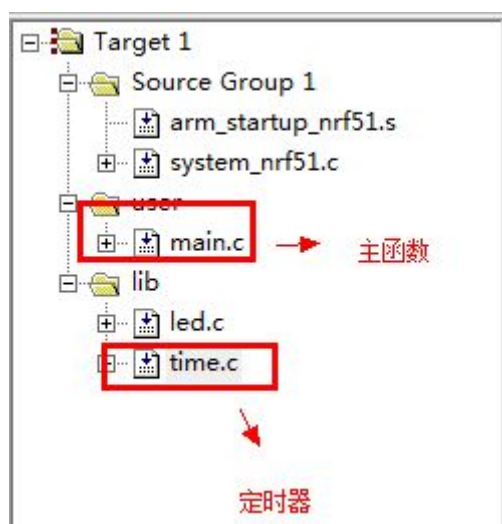
2.3.1 硬件准备:

如下图所示: 青云 QY-nRF51822 开发板上, 通过管脚 PA11 和管脚 PA12 连接 2 个 LED 灯, 我们下面的任务通过 led 灯来指示不同定时器的定时时间。IO 管脚接分别接一个发光二极管, 因此当把 IO 管脚定义为输出高电平的时候, 就可以点亮发光二极管了。图中 SB6 和 SB7 分别为两个外设隔离点, 当不需要使用 LED 灯的时候可以把这两个点割开。



2.3.2 软件准备:

在代码文件中, 实验三建立了一个演示历程, 我们打开看看需要那些库文件。打开 user 文件夹中的 time 工程:



如上图所示: 只需要自己编写红色框框里的两个文件就 OK 了, 因为采用子函数的方式其中 led.c 在上一节控制 LED 灯的时候已经写好, 现在我们就来讨论下如何编写 time.c 这个驱动子文件。

time.c 文件主要是要起到两个作用: 第一: 初始化定时器参数。第二: 设置定时时

间函数。完成这两个功能就可以在 `main.c` 文件中直接调用本驱动了。

下面我们就结合寄存器来详细分析下定时器的设置:

Register	Offset	Description
TASKS		
START	0x000	Start Timer
STOP	0x004	Stop Timer
COUNT	0x008	Increment Timer (Counter mode only)
CLEAR	0x00C	Clear timer
CAPTURE[0]	0x040	Capture Timer value to CC0 register
CAPTURE[1]	0x044	Capture Timer value to CC1 register
CAPTURE[2]	0x048	Capture Timer value to CC2 register
CAPTURE[3]	0x04C	Capture Timer value to CC3 register
EVENTS		
COMPARE[0]	0x140	Compare event on CC[0] match
COMPARE[1]	0x144	Compare event on CC[1] match
COMPARE[2]	0x148	Compare event on CC[2] match
COMPARE[3]	0x14C	Compare event on CC[3] match
REGISTERS		
SHORTS	0x200	Shortcuts
INTENSET	0x304	Write-only - configures which events generate a Timer interrupt
INTENCLR	0x308	Write-only - configures which events do not generate a Timer interrupt
MODE	0x504	Timer mode selection
BITMODE	0x508	Configure the number of bits used by the TIMER
PRESCALER	0x510	Timer prescaler register
CC[0]	0x540	Capture/Compare register 0
CC[1]	0x544	Capture/Compare register 1
CC[2]	0x548	Capture/Compare register 2
CC[3]	0x54C	Capture/Compare register 3

上面红色框框的几个寄存器大家注意了, 这个是下面设置要用的几个寄存器, 英文解释很清楚, 关键是搞清楚怎么用, 我们之间对着代码段分析:

```

01.    p_timer->MODE = TIMER_MODE_MODE_Timer;           // 设置为定时器模式
02.    p_timer->PRESCALER = 9; //9 分频
03.    p_timer->BITMODE = TIMER_BITMODE_BITMODE_16Bit; // 16 bit 模式.
04.    p_timer->TASKS_CLEAR = 1;                          // 清定时器.
05.
06.    // 分频后的时钟*31.25 后为 1ms
07.    p_timer->CC[0] = number_of_ms * 31;
08.    p_timer->CC[0] += number_of_ms / 4;
09.    p_timer->TASKS_START = 1;                          // 开始定时器
10.

```

上面一段代码的编写严格按照了寄存器要求进行, 首先是 **MODE**, 也就是模式设置, 我们设置定时器模式。模式类型也很简单, 和单片机类似, 分为定时器模式和计数器模式, 如下图所示:

17.2.2 MODE

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID (Field ID)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	RW	Field	Value	Description																												
A	RW			Timer mode																												
		TIMER	0	Select timer mode																												
		COUNTER	1	Select counter mode																												

PRESCALER 寄存器设置预分频计数器，这个寄存器是干什么的？我们首先看看定时器内部结构，如下图所示，要产生 **Ftimer** 时钟，必须把外部提供的高速时钟 **HFCLK** 首先进行分频：

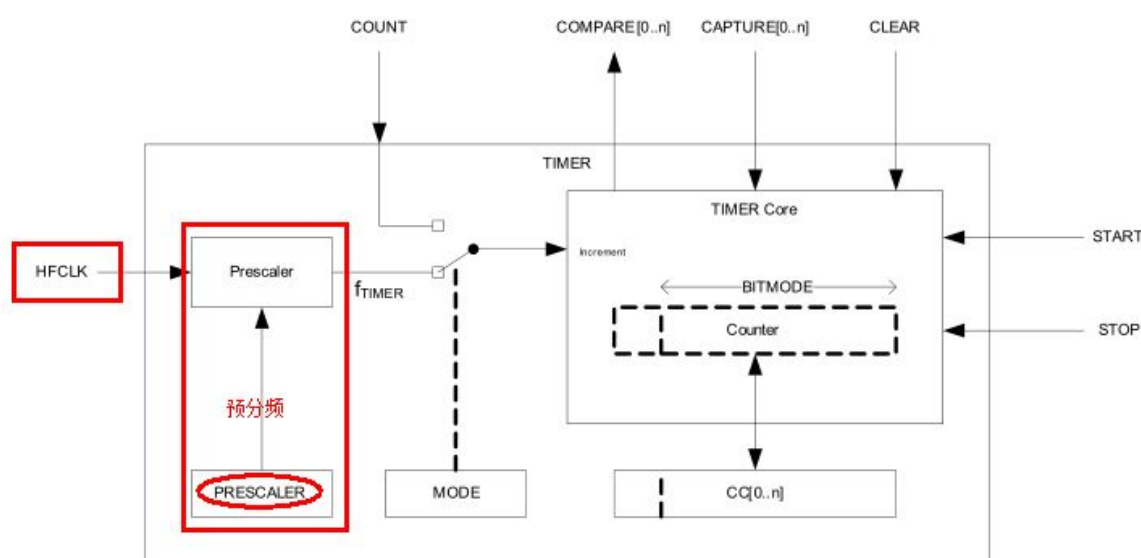


Figure 28 Block schematic for Time/counter

那么这个计算关系是什么？我们之间查数据手册。看到下面的公式：

$$f_{TIMER} = \frac{HFCLK}{2^{PRESCALER}}$$

代码里，我们设置为 9 分频，大家可以计算一下，**Ftimer** 时钟为 31250 Hz。然后是 **BITMODE** 寄存器，我们设置为 16bit。定时器计数之前需要清空，然后设定定时比较的值。满了溢出。设置和 51 单片机差不多。设置好了就开始启动。

外部提供的高速时钟 **HFCLK** 我们必须首先初始化进行开启：

```

11. // 开始 16 MHz 晶振.
12. NRF_CLOCK->EVENTS_HFCLKSTARTED = 0;
13. NRF_CLOCK->TASKS_HFCLKSTART     = 1;
14.
15. // 等待外部振荡器启动

```



```
16. while (NRF_CLOCK->EVENTS_HFCLKSTARTED == 0)
17. {
18.     // Do nothing.
19. }
```

并且加入定时器选择的功能。

```
20. switch (timer)
21. {
22.     case TIMER0:
23.         p_timer = NRF_TIMER0;
24.         break;
25.
26.     case TIMER1:
27.         p_timer = NRF_TIMER1;
28.         break;
29.
30.     case TIMER2:
31.         p_timer = NRF_TIMER2;
32.         break;
33.
34.     default:
35.         p_timer = 0;
36.         break;
37. }
```

那么主函数就是十分的简单了，直接调用我们写好的驱动函数，LED 灯指示定时器相应的变化。函数如下所示：

```
38. /******* (C) COPYRIGHT 2014 青风电子 *****/
39. * 文件名   : main
40. * 描述     :
41. * 实验平台: 青云 nRF51822 开发板
42. * 描述     : 定时器定时
43. * 作者     : 青风
44. * 店铺     : qfv5.taobao.com
45. *****/
46. #include "nrf51.h"
47. #include "led.h"
48. #include "time.h"
49.
50.
51. int main(void)
52. {
53.     //
54.     LED_Init();
55.
```

```
56.     while (1)
57.     {
58.         LED1_Toggle();
59.         //使用定时器 0 产生 1s 定时
60.         nrf_timer_delay_ms(TIMER0, TIMER_DELAY_MS);
61.
62.         LED1_Toggle();
63.         // 使用定时器 1 产生 1s 定时
64.         nrf_timer_delay_ms(TIMER1, TIMER_DELAY_MS);
65.
66.         LED1_Toggle();
67.         // 使用定时器 2 产生 1s 定时
68.         nrf_timer_delay_ms(TIMER2, TIMER_DELAY_MS);
69.     }
70. }
71.
72.
```

实验下载到青云 nRF51822 开发板后的实验现象如下, 参数 1s 的定时闪烁:

