

讯联电子nRF51822蓝牙4.0开发实战

TIMER

V:1.0





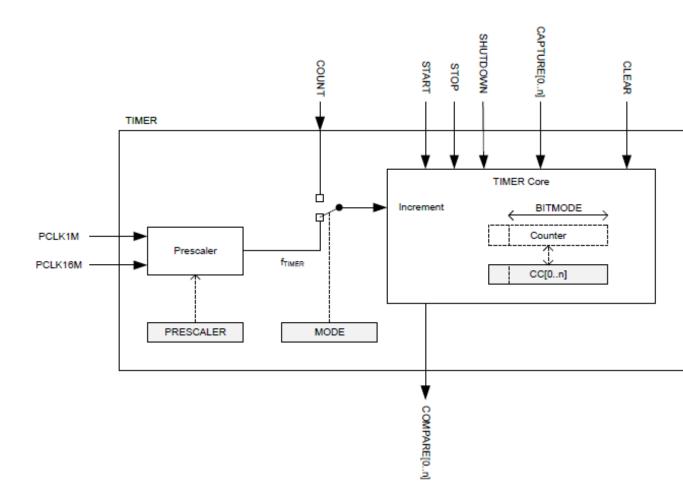
申明:本教程版权归讯联电子所有。本教程仅供内部客户交流之用。如需引用,请注明出处。由于工程师水平有限,文档难免有所疏漏和错误,由此造成的损失,讯联电子不承担任何责任。





该讲介绍 51822 的 Timer/Counter 模块工作在 timer 模式下(定时器模式,还可以工作为计数器模式)如何操作

51822 的 Timer/Counter 结构如下图所示



Timer 模块从 PCLK16M/PCLK1M 处获得时钟源,然后经分频后得到的时钟作为 timer 模块的时钟 (上图 Ftimer)。当 timer 模块选择为 timer 模式时,Counter 会在 Ftimer 的每个 tick 计数一次当计数值与 cc[n](n 为 0,1,2,3)寄存器中的值相等时就会触发对应的 Compare[n]事件,如果我们设置了 compare[n]事件产生时触发中断(关于事件与中断看前一篇 GPIOTE),那么就可以在 counter 计数到与 cc[n]寄存器中的值相等时触发中断,也就能实现我们需要的定时器功能了



所以根据上面的模块结构图和说明想要实现定时器,我们需要做如下几个步骤:

- 1 选择 Timer/Countermo 模块为 timer 模式, 并设置 bitmode (8, 16, ,2,32 位)
- 2 通过设置分屏来设置 timer 的时钟
- 3 设置 cc[n](后面我们的例子选择使用 cc0),来设置计数到多少产生compares[n]事件(当计数值技术到cc[n]的值时对应产生compare[n]事件)
- 4 设置 compare 事件产生时触发中断。
- 5 通过 NVIC 函数启动 MCU 的 timer 中断
- 6 最后启动 timer 就可以了。

通过查看 数据手册看以看到

1

设置模式 通过 寄存器 MODE 设置 0为 timer 模式 1为 counter 模式。

2

设置 timer 的时钟 通过以下公式设置

fTIMER = 16 MHz / (2 PRESCALER)

这里可能有个疑问,上面的图解中不是有两个时钟源 16M 和 1M 吗,怎么这个公式只能通过 16M 来分频获得 timer 时钟。这是因为 51822 为了降低功耗内部自动 做了时钟源切换,当 Ftimer <= 1M 时会自动切换成 1M 时钟源

举两个例子解释下

如果需要 timer 的时钟为 4M, 那么 $4 = Ftimer = 16M/2^2$ 即我们只需设置分频寄存器 PRESCALER 为 2,就能或得 4M 的时钟给 timer 了

当需要 timer 的时钟为 500 Khz 时,根据公式 我们设置 PRESCALER 寄存器的值为 5, $500 \text{kHZ} = \text{Ftimer} = 16 \text{M}/2^5$ 。 这个时候 Ftimer <= 1 M,所以 51882 内部会自动切换成 1 M 的时钟源然后分频后获得 500 K 的 timer 时钟。不过这些都是 51822 自动切换的了

也就是说设置 timer 时钟只要根据上面的公式设置就可以了,时钟源的切换 是 51822 自动完成的

3

设置 cc[n]寄存器的值,定时就是通过这个值来设置的。下面的例子会做一个一秒定时亮灯/灭灯的程序,我们设置timer时钟为1M,即分频寄存器PRESCALER



写值为 4。 1M 的时钟源则一个 tick 为 1us, 所以要定时 1s, 则 cc [0] 的值我们填入 1000000 就行了。(这里也可以选择 cc [1], cc [2], 或 cc [3], 只要下面对应的 compare 事件产生中断设置成对应的就可以了)

4

通过寄存器 INTENSET 第 16bit 位设置 compare[0]事件产生时触发中断。

5

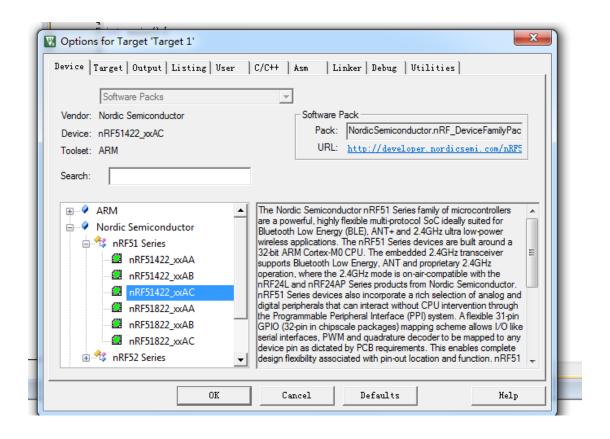
通过 NVIC 的功能函数 NVIC Enable IRQ 来使能 MCU 的 Timer 0 中断

6

最后通过 Timer/Counter 模块的 START 启动 timer。

注意: 这里裸板例子代码我们用的是 timer0, 如果跑了协议栈就不能用 timer0, 可以使用 timer1 和 timer2

新建一个工程选择自己的芯片型号



作者: 不离不弃 qq 574912883



下面的代码我们不使用 nrf 提供的库函数,而按照上面说明的顺序直接设置寄存器来使用 timer0,因为没用别的功能。所以下面勾选一个 core 和 startup。因为用到了点灯所以勾选一下 nrf_drivers 下的 Nrf_gpio

Software Component	Sel.	Variant		Version	Description
🖭 💠 Board Support					Generic Interfaces for Evaluation and Development Boards
□ � CMSIS					Cortex Microcontroller Software Interface Components
CORE	~			3.40.0	CMSIS-CORE for Cortex-M, SC000, and SC300
∳ DSP				1.4.2	CMSIS-DSP Library for Cortex-M, SC000, and SC300
⊕ 💠 RTOS (API)				1.0	CMSIS-RTOS API for Cortex-M, SC000, and SC300
					Unified Device Drivers compliant to CMSIS-Driver Specification
Device					Startup, System Setup
Startup	~			8.0.3	System Startup and header files for nRF51 Series
🛨 💠 File System		MDK-Pro		6.2.0	File Access on various storage devices
± ♦ Graphics		MDK-Pro		5.26.1	User Interface on graphical LCD displays
Network		MDK-Pro		6.2.0	IP Networking using Ethernet or Serial protocols
⊕ ◆ RTOS		Clarinox		1.0.0	Clarinox implementation of uC/OS
⊕ ❖ USB		MDK-Pro		6.2.0	USB Communication with various device classes
⊕ ♦ Wireless		Clarinox		2.0.0	Clarinox Wireless Libraries
⊕ • nRF_ANT					
⊕ � nRF_BLE					
nRF_Drivers					
app_uart		default	•	3.0.1	UART driver
♦ ble_flash				3.0.1	Flash driver for BLE examples
nrf_adc				3.0.1	ADC Hardware Abstraction Layer
orf_clock				3.0.1	Software Component
onf_delay				3.0.1	simple delay module

下面是源代码,因为都是直接操作寄存器,所以更简单直观

```
Main.c

#include "nrf51.h"

#include "nrf_gpio.h"

//定义自己板子上的 LED 灯

#define LED 22

int main() {
```



```
nrf gpio cfg output(LED);
   //NRF TIMERO 定义在 nrf51.h 中, 该指针指向 timer0 中的寄存器组
   NRF_TIMERO->PRESCALER = 4; //2<sup>4</sup> 16 分频得到 1M timer 时钟
   NRF TIMERO -> MODE = 0;
                                 //timer 模式
   NRF_TIMERO->BITMODE = 3;  // 设置 32bit
NRF_TIMERO->CC[0] = 1000000;  //一个 tick 是 1us, 1000000 代表 1s
   NRF TIMERO->INTENSET = 1<<16;//设置 compare[0]事件产生时触发中断
  //该设置使 timer 模块中的 conter 计数到 cc[0]值时会自动清零,以带到重
  //新计数的目的
   NRF TIMERO -> SHORTS = 1;
//启动 timer 模块
   NRF_TIMERO->TASKS_START = 1;
//开启 MCU 的 timer0 中断
   NVIC SetPriority(TIMERO IRQn, 3);
   NVIC ClearPendingIRQ(TIMERO IRQn);
   NVIC EnableIRQ(TIMERO IRQn);
   while (1);
   return 0;
}
//中断函数中翻转灯状态
void TIMERO IRQHandler() {
   if(NRF TIMERO->EVENTS COMPARE[0] == 1) {
      NRF_TIMERO->EVENTS_COMPARE[0] = 0; //清除事件,不然会导致一
                                           //直产生中断
      nrf_gpio_pin_toggle(LED);
   }
}
```