

定时器 T1-中断方式

1. 实验目的

- 1) 通过实验掌握 CC2540 芯片 GPIO 的配置方法
- 2) 掌握 Led 驱动电路及开关 Led 的原理
- 3) 掌握定时器 T3(8 位)通过中断方式控制 LED1 周期性闪烁

2. 实验设备

硬件:

PC 机一台

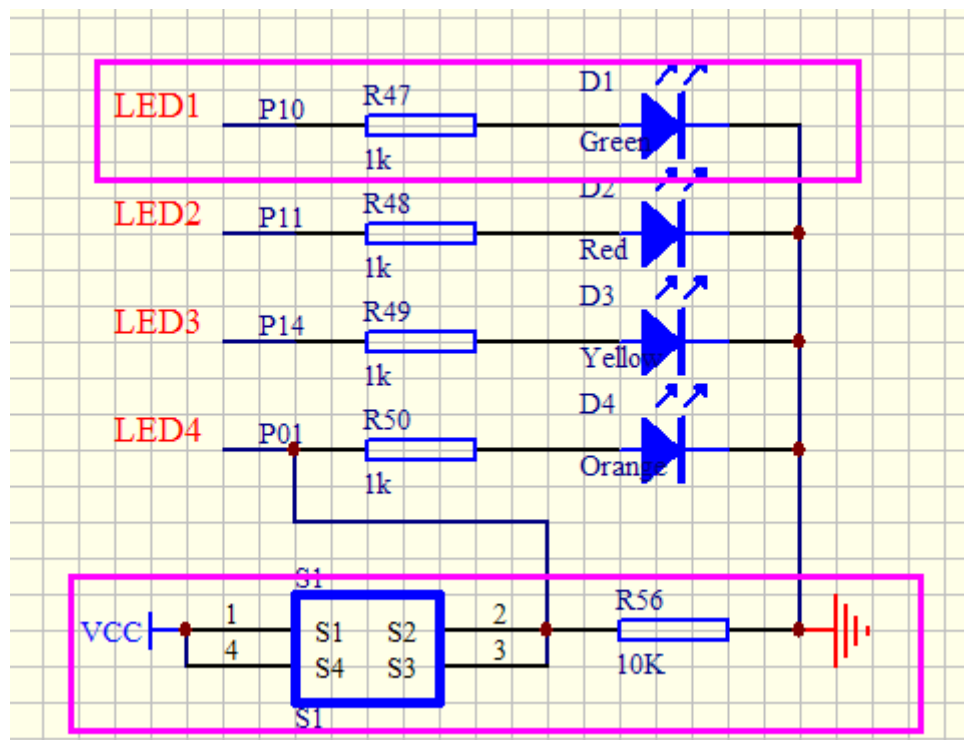
SmartRF cc254x (底板、核心板、仿真器、USB 线) 一套

软件:

XP/win7 系统, IAR 8.10 集成开发环境

3. 实验相关电路图

由于发光二极管单向导电特性, 即只有在正向电压 (二极管的正极接正, 负极接负) 下才能导通发光。



4. 实验相关寄存器

CC2530 的 T3 定时器（8 位）需要了解 T3CTL, T3CCTL0, T3CC0, T3CCTL1, T3CC 寄存器。如下表所示：

寄存器	作用	描述
T3CTL(0XCB)	定时器 3 的控制和状态	<p>Bit[7:5]：定时器时钟分频倍数选择： 000：不分频 001：2 分频 010：4 分频 011：8 分频 100：16 分频 101：32 分频 110：64 分频 111：128 分频</p> <p>Bit4：T3 起止控制位</p> <p>Bit3：溢出中断掩码 0：关溢出中断 1：开溢出中断</p> <p>Bit2：清计数值 高电平有效</p> <p>Bit[1:0] T3 模式选择 00：自动重装 0x00-0xFF 01：DOWN (从 T3CC0 到 0X00 计数一次) 10：模计数（反复从 0X00 到 T3CC0 计数） 11：UP/DOWN(反复从 0X00 到 T3CC0 计数再到 0X00)</p>

T3CCTL0(0xCC)	T3 通道 0 捕获 / 比较控制寄存器	<p>Bit6: 通道 0 中断屏蔽 0: 中断禁止 1: 中断使能</p> <p>Bit[5: 3] T3 通道 0 比较输出模式选择</p> <p>Bit2: T3 通道 0 模式选择: 0: 捕获 1: 比较</p> <p>Bit[1:0] T3 通道 0 捕获模式选择 00 没有捕获 01 上升沿捕获 10 下降沿捕获 11 边沿捕获</p>
T3CC0(0xCD)	定时器 3 通道 0 捕获/比较值	定时器捕获 / 比较值通道 0。当 T3CCTL0.MODE=1 (比较模式) 时写该寄存器会导致 T3CC0.VAL[7:0]更新到写入值延迟到 T3CNT.CNT[7:0]=0x00。
T3CCTL1(0xCE)	T3 通道 1 捕获 / 比较控制寄存器	<p>Bit6: 通道 1 中断屏蔽 0: 中断禁止 1: 中断使能</p> <p>Bit[5: 3] T3 通道 1 比较输出模式选择</p> <p>Bit2: T3 通道 1 模式选择: 0: 捕获 1: 比较</p> <p>Bit[1:0] T3 通道 1 捕获模式选择 00 没有捕获 01 上升沿捕获 10 下降沿捕获 11 边沿捕获</p>
T3CC1(0xCF)	定时器 3 通道 1 捕获/比较值	定时器捕获 / 比较值通道 1。当 T3CCTL1.MODE=1 (比较模式) 时写该寄存器会导致 T3CC1.VAL[7:0]更新写入值延迟到 T3CNT.CNT[7:0]=0x00。

按照表格寄存器的内容，对 T3 进行配置，由于定时器 T3 为 8 位所以配置稍有不同。

```

T3CTL |= 0x08;      //开溢出中断
T3IE = 1;           //开总中断和 T3 中断
T3CTL |= 0xE0;      //128 分频,128/16000000*N=0.5S,N=62500
T3CTL &= ~0x03;     //自动重装 00—>0xff 62500/255=245(次)
T3CTL |= 0x10;      //启动
EA = 1;             //开总中断

```

5. 源码分析

/******

技术支持与项目开发合作(Tel) 13510953260 QQ11940507 阿莫

```

* 文 件 名: main.c
* 作    者: Amo [ www.amoMcu.com 阿莫单片机]
* 修    订: 2014-04-08
* 版    本: 1.0
* 描    述: 定时器 T3 通过中断方式控制 LED1 周期性闪烁
*****/

#include <ioCC2540.h>

typedef unsigned char uchar;
typedef unsigned int  uint;

#define LED1 P1_0          // P1.0 口控制 LED1

uint count;                //用于定时器计数

/*****
* 名    称: InitLed()
* 功    能: 设置 LED 灯相应的 IO 口
* 入口参数: 无
* 出口参数: 无
*****/
void InitLed(void)
{
    P1DIR |= 0x01;          //P1.0 定义为输出
    LED1 = 0;               //使 LED1 灯下电默认为熄灭
}

/*****
* 名    称: InitT3()
* 功    能: 定时器初始化, 系统不配置工作时钟时默认是 2 分频, 即 16MHz
* 入口参数: 无
* 出口参数: 无
*****/
void InitT3()
{
    T3CTL |= 0x08;          //开溢出中断
    T3IE = 1;               //开总中断和 T3 中断
    T3CTL |= 0xE0;          //128 分频, 128/16000000*N=0.5S, N=62500
    T3CTL &= ~0x03;         //自动重装 00—>0xff 62500/255=245(次)
    T3CTL |= 0x10;          //启动
    EA = 1;                 //开总中断
}

//定时器 T3 中断处理函数

```

```

#pragma vector = T3_VECTOR
__interrupt void T3_ISR(void)
{
    IRCON = 0x00;           //清中断标志, 也可由硬件自动完成
    if(count++ > 245)       //245 次中断后 LED 取反, 闪烁一轮 (约为 0.5 秒时间)
    {
        //经过示波器测量确保精确
        count = 0;         //计数清零
        LED1 = ~LED1;      //改变 LED1 的状态
    }
}

/*****
* 程序入口函数
*****/

void main(void)
{
    InitLed();              //设置 LED 灯相应的 IO 口
    InitT3();               //设置 T3 相应的寄存器
    while(1)
    {};
}

```