



基于定时器模式的间隔定时

课程目标与重难点

知识目标

1

掌握

模模式的工作原理，定时器最大计数值的计算，TICCTL0寄存器的定义。

能力目标

2

能够

在IAR环境中，编写模模式定时的初始化函数与中断服务函数，实现1秒定时功能。

素质目标

3

培养

硬件工程师系统分析、独立思考、动手实践、协同合作的基本素质。

重点：模模式定时原理及最大计数值的计算。

难点：定时器的初始化函数与中断服务函数的设计与实现。

项目内容与任务分解

设计程序，实现**基于定时器1模模式的秒闪**。功能如下：

- 【1】系统上电后，全部LED灯关闭。
- 【2】选择内部**16MHz时钟的128分频**作为定时器1的计数信号。
- 【3】在定时器1的**模模式**中实现**0.1秒**的间隔定时。
- 【4】在中断服务函数中实现**1秒的间隔定时**。
- 【5】控制**D4灯**实现**秒闪**功能，即**D4灯亮1秒，灭1秒...**

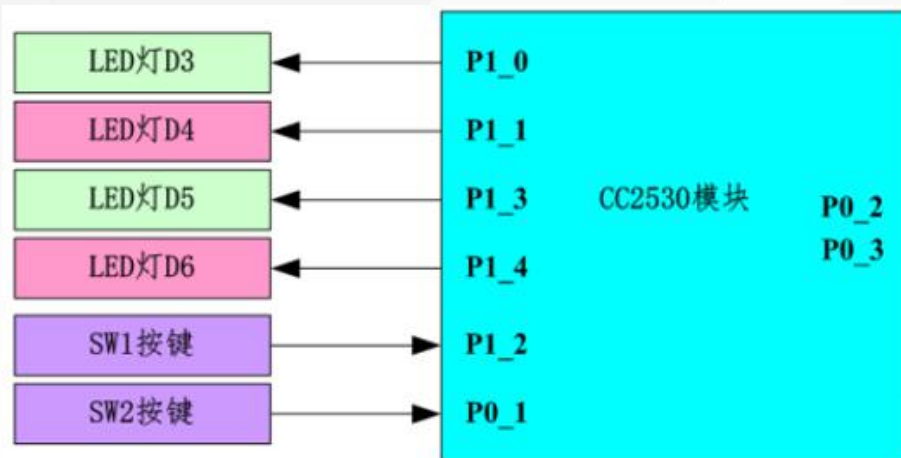
任务01：学习相关知识。

任务02：构思项目实施思路。

任务03：动手编写代码。

任务04：对比老师讲解。

任务05：修正完善项目。



任务

01

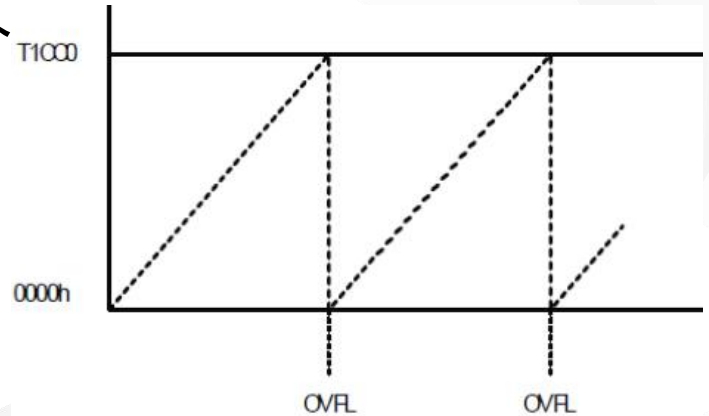
学习定时器的相关知识

CC2530的定时/计数器

- CC2530共有**5个定时/计数器**，分别是：
- **定时器1：16位定时器**，是CC2530中**功能最全**的一个定时/计数器，在应用中应优先选用。该定时器支持输入捕获、输出比较、PWM输出、能触发DMA功能，具有**5个独立的捕获/比较通道**，每个通道使用一个I/O引脚。具有**自由运行、模模式和正计数/倒数**三种不同工作模式。
- **定时器2**：用于CSMA-CA算法提供定时，用户**一般不使用**该定时器。
- **定时器3和定时器4：8位定时器**，支持输入捕获、输出比较、具有2个独立的捕获/比较通道，每个通道使用一个I/O引脚。
- **睡眠定时器：24位正计数定时器**，主要用于设置系统进入和退出低功耗睡眠模式之间的周期。

模模式的工作原理

- **模模式**：计数器从**0x0000**开始，在每个活动时钟边沿**加1**，计数器达到**T1CC0**寄存器保存的值时产生**溢出**，计数器又将从**0x0000**开始**新一轮的递增**计数。



- 计数溢出后，自动将相应的标志位置1，如果设置了相关的中断使能则会产生一个中断请求。
- 模模式的**计数周期不是固定值**，可由用户自行设定。
- 定时器1的**模模式**，需要开启其**通道0**的**输出比较模式**。

最大计数值的计算与设置

- 定时器1有**5对T1CCxH和T1CCxL**寄存器，分别对应**通道0到通道4**。在使用**定时器1的间隔定时功能**时，使用**T1CC0H和T1CC0L**两个寄存器存放**最大计数值的高8位和低8位**。
- 最大计数值 = 定时时长 / 定时器计数周期。

$$\text{最大计数值} = \frac{\text{定时时长}}{\text{定时器计数周期}} = \frac{0.1\text{S}}{\frac{1}{16\text{M}} \times 128} = 12500 = 0\text{x}30\text{D}4$$

- 在程序设计中，要**先写低8位寄存器**，**后写高8位寄存器**。

T1CC0L = 0xD4;

T1CC0H = 0x30;

T1CCTL0寄存器

- 定时器1的**模模式**，需要开启其**通道0的输出比较模式**。

T1CCTL0 寄存器（更多内容请看 CC2530 的数据手册）

位	位名称	复位值	操作	描述
7	RFIRQ	0	R/W	当设置时，使用 RF 中断捕获，而不是常规捕获输入。
6	IM	1	R/W	通道 0 中断屏蔽。 0 : 禁止通道 0 中断。 1 : 使能通道 0 中断。
5:3	CMP[2:0]	000	R/W	通道 0 比较模式选择。 当定时器的值等于 T1CC0 中的比较值，选择操作输出。 有 8 种输出操作，具体详看数据手册。
2	MODE	0	R/W	定时器 1 通道 0 的模式选择。 0 : 捕获模式。 1 : 比较模式 。
1:0	CAP[1:0]	00	R/W	通道 0 捕获模式选择。 00 : 为捕获。 01 : 上升沿捕获。 10 : 下降沿捕获。 11 : 所有沿捕获。

```
P1CCTL0 |= 0x04;
```

```
//将定时器 1 通道 0 的模式选择为比较模式
```


1+X证书“传感网应用开发”考题

- **2020年1月份，单选题第10题。**

- 下面哪一项不是单片机中的定时/计数器具有的功能（ ）。
 - A. 定时功能
 - B. 计数功能
 - C. 中断功能
 - D. AES加密功能

- **2020年1月份，多选题第5题。**

- 以下属于CC2530定时/计数器的工作模式的有（ ）。
 - A. 自由运行模式
 - B. 模模式
 - C. 时间戳模式
 - D. 正计数/倒数模式

任务

02

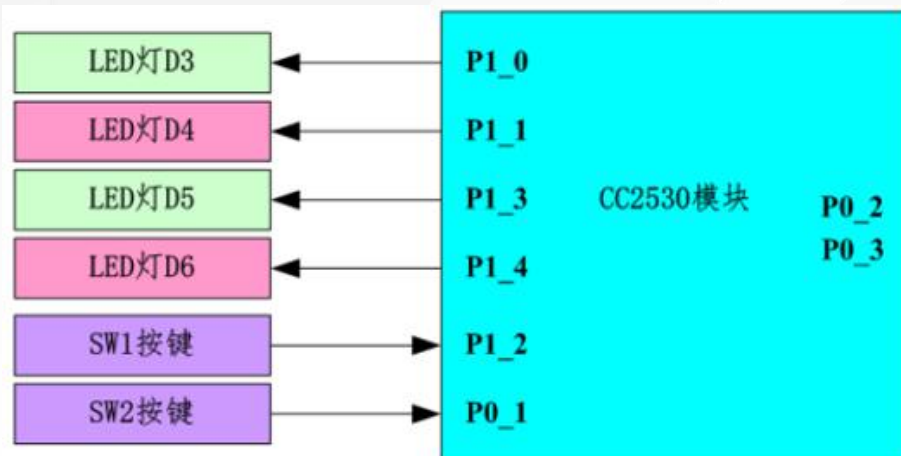
构思项目的实现思路

项目实施思路提示

设计程序，实现**基于定时器1模模式的秒闪**。功能如下：

- 【1】系统上电后，全部LED灯关闭。
- 【2】选择内部**16MHz时钟的128分频**作为定时器1的计数信号。
- 【3】在定时器1的**模模式**中实现**0.1秒**的间隔定时。
- 【4】在中断服务函数中实现**1秒的间隔定时**。
- 【5】控制**D4**灯实现**秒闪**功能，即**D4**灯亮1秒，灭1秒...

- 【1】计算定时器最大计数值。
- 【2】设计定时器初始化函数。
- 【3】设计定时器中断服务函数。
- 【4】实现1秒的间隔定时。
- 【5】实现D4的秒闪功能。



模模式间隔定时程序设计思路关键

- 定时器的程序设计，需要定时器**初始化**和**中断服务**这两个函数。
- 在定时器的**初始化函数**中：
 - 1-计算**最大计数值**，并赋值给**T1CC0L**和**T1CC0H**寄存器。
 - 2-设置**T1CCTL0寄存器**，开启通道0的输出比较模式。
 - 3-使能**定时器1中断**和**总中断**。
 - 4-设置**T1CTL寄存器**，设置128分频并以模模式开始工作。
- 在定时器的**中断服务函数**中：

对全局变量**count**累计，

当**count**累计到10，即1秒的间隔定时到，对D4灯进行状态切换，同时将**count**清零，进入下一秒的定时工作。

任务

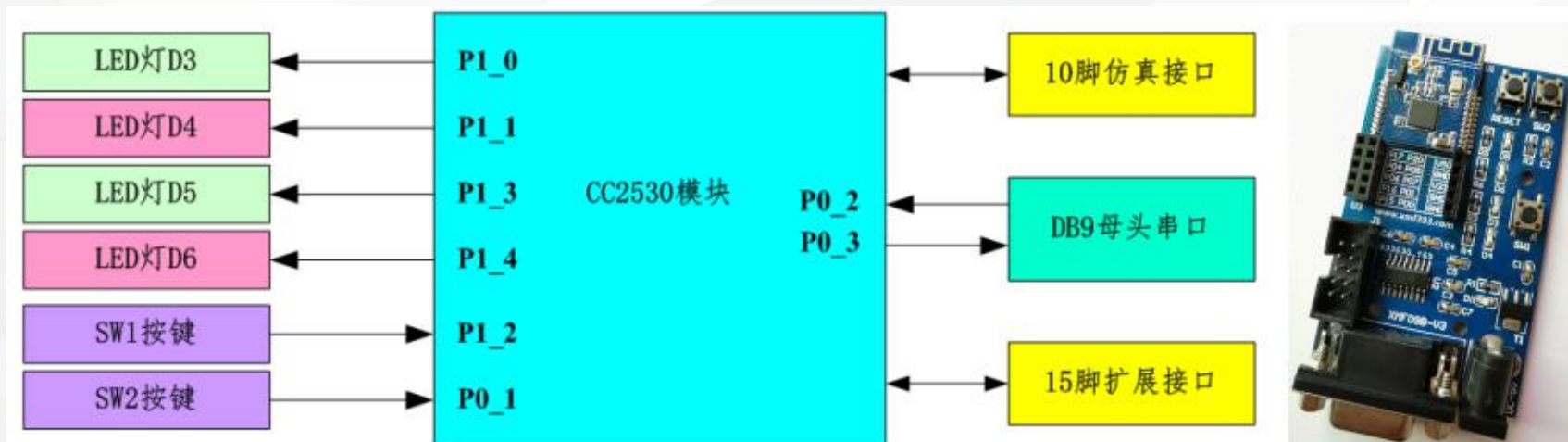
03

动手编写程序实现项目功能

项目实施思路提示

设计程序，实现**基于定时器1模模式的秒闪**。功能如下：

- 【1】系统上电后，全部LED灯关闭。
- 【2】选择内部**16MHz时钟的128分频**作为定时器1的计数信号。
- 【3】在定时器1的**模模式**中实现**0.1秒**的间隔定时。
- 【4】在中断服务函数中实现**1秒的间隔定时**。
- 【5】控制**D4灯**实现**秒闪**功能，即**D4灯亮1秒，灭1秒...**



任务

04

对比老师演示，反思实现过程

项目实施难点解析

- 定时器的**初始化函数**：

```
5 void Init_Timer1()  
6 {  
7     T1CC0L = 0xd4;    //设置最大计数值的低8位  
8     T1CC0H = 0x30;    //设置最大计数值的高8位  
9     T1CCTL0 |= 0x04; //开启通道0的输出比较模式  
10    T1IE = 1;         //使能定时器1中断  
11    EA = 1;          //使能总中断  
12    T1CTL = 0x0e;     //分频系数是128,模模式  
13 }
```

- 定时器的**中断服务函数**：

```
30 unsigned int count = 0;  
31 #pragma vector = T1_VECTOR  
32 __interrupt void Service_Timer1()  
33 {  
34     count++;  
35  
36     if((count % 10) == 0)  
37     {  
38         D4 = ~D4;  
39     }  
40 }
```

任务

05

修正完善项目，巩固知识技能

项目实现的参考代码

```
1 #include "ioCC2530.h"
2
3 #define D4 P1_1
4
5 void Init_Port()
6 {
7     P1SEL &= ~0x1B;
8     P1DIR |= 0x1B;
9     P1 &= ~0x1B;
10 }
11 //=====
12 void Init_Timer1()
13 {
14     //1-设置最大计数值
15     T1CCOL = 0xD4;
16     T1CCOH = 0x30;
17     //2-开启通道0的比较模式
18     T1CTL0 |= 0x04;
19     //3-使能定时器1的中断
20     T1IE = 1;
21     //4-使能总中断
22     EA = 1;
23     //5-设置定时器1的分频系数和工作模式
24     T1CTL = 0x0E;
25 }
```

```
27 unsigned int count = 0;
28 #pragma vector = T1_VECTOR
29 __interrupt void Service_Timer1()
30 {
31     count++;
32
33     if((count % 10) == 0)
34     {
35         D4 = ~D4;
36     }
37 }
38
39 //=====
40 void main()
41 {
42     Init_Port();
43     Init_Timer1();
44     while(1);
45 }
```

谢谢大家

