

哈尔滨工业大学 (深圳)

嵌入式系统软件设计基础 实验报告

实验所属课程： 电子工艺

姓名： 李木晗

学号： SZ170210119

专业： 电子信息

评分：

批阅老师：

实验内容：

实验代码要求：关键代码要注释，整个工程打包发到指定邮箱：jingjing.yang@163.com

邮件名称要求：专业_姓名_学号_软件基础实验报告

1.1 实验目的

设计一个灯光控制系统，该系统主要用于展览馆等需要解说员解说，且需要调节光线以达到最佳演示效果的场合。系统检测到外界声音后打开灯光，系统根据周边环境光的情况自动调整 LED 灯的亮度，以达到最佳展示效果。一定时间后，如果没有检测到声音信号，则自动关闭 LED 灯，以达到节能目的。

1.2 实验原理及用户手册查找

（文字、图表等阐述要实现的实验效果，阐述相关模块的使用方法和有关理论知识和寄存器使用等方法）

1.2.1 LED 灯

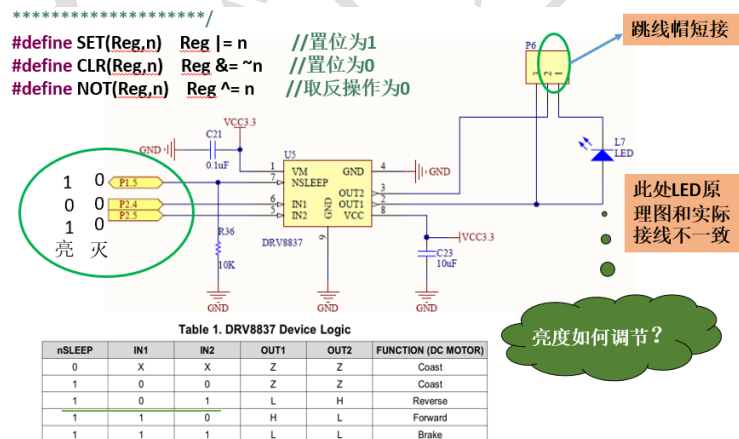
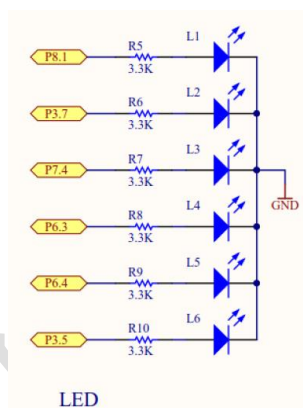


Table 1. DRV8837 Device Logic

nSLEEP	IN1	IN2	OUT1	OUT2	FUNCTION (DC MOTOR)
0	X	X	Z	Z	Coast
1	0	0	Z	Z	Coast
1	0	1	L	H	Reverse
1	1	0	H	L	Forward
1	1	1	L	L	Brake

```
void light_on(void) {
    P8OUT |= 0x02;
    P3OUT |= 0x80;
    P7OUT |= 0x10;
    P6OUT |= 0x08;
    P6OUT |= 0x10;
    P3OUT |= 0x20;
}
```

```
void light_off(void) {
    P8OUT &= ~ 0x02;
    P3OUT &= ~ 0x80;
    P7OUT &= ~ 0x10;
```

```
P6OUT &= ~ 0x08;
P6OUT &= ~ 0x10;
P3OUT &= ~ 0x20;
}

void led_on(void) {
    P1OUT |= BIT5;
    P2OUT &= ~ BIT5;
}
```

```
void led_off(void) {
    P1OUT &= ~ BIT5;
    P2OUT |= BIT5;
}
```

```
void initLED(void) {
    P8DIR |= 0x02;
    P3DIR |= 0x80;
    P7DIR |= 0x10;
    P6DIR |= 0x08;
    P6DIR |= 0x10;
    P3DIR |= 0x20;

    P1DIR |= BIT5;
    P2DIR |= BIT2 + BIT4 + BIT5;
}
```

1.2.2 看门狗定时器

利用看门狗定时器控制 LED 开关。当声音强度达到一定的值则喂狗，否则进入中断时关闭大功率 LED。

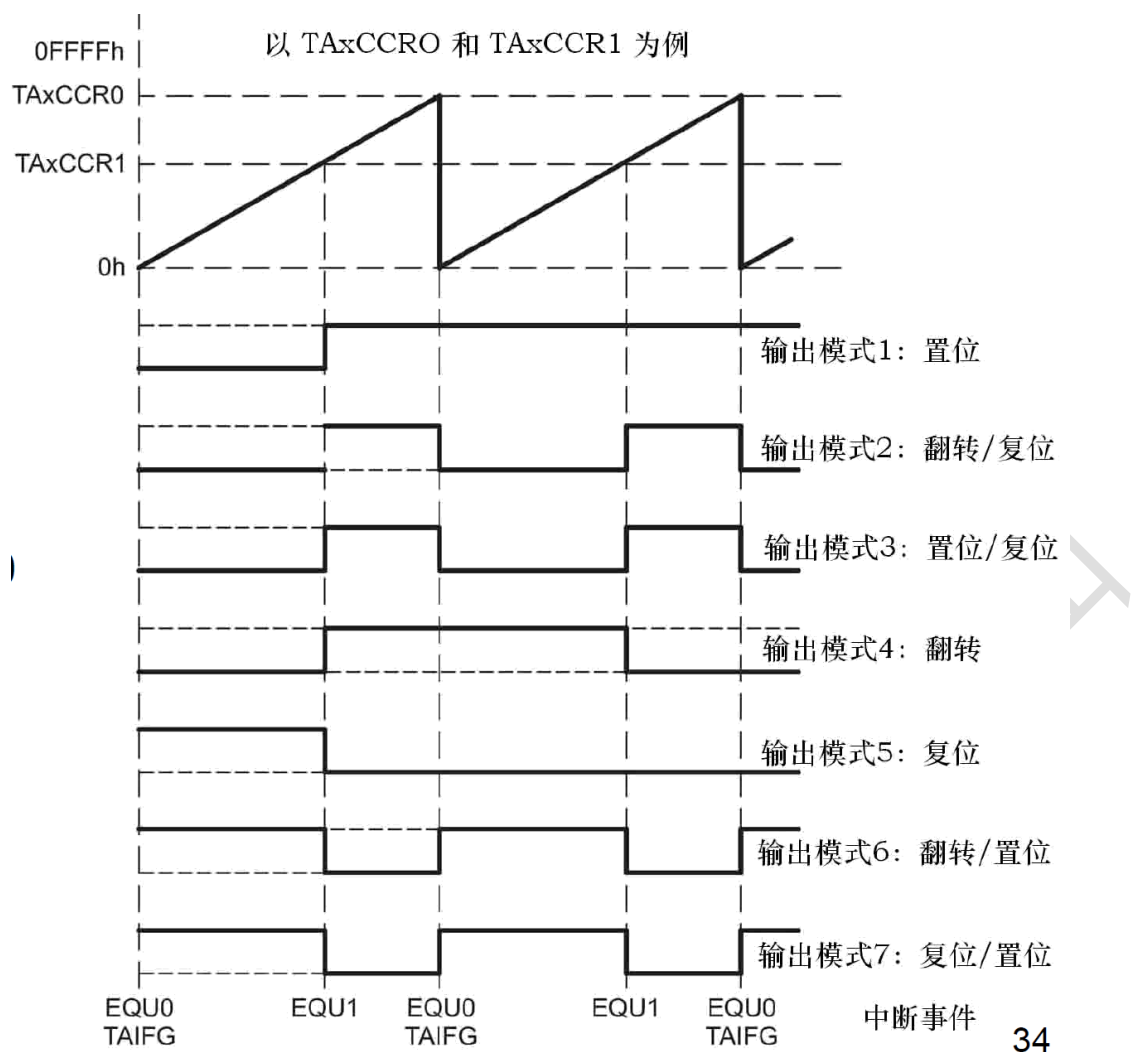
Watchdog Timer Control Register

Figure 16-2. WDTCTL Register

15		14		13		12		11		10		9		8	
WDTPW															
7		6		5		4		3		2		1		0	
WDTHOLD		WDTSEL				WDTTMSSEL		WDTCNTCL		WDTIS					
rw-0		rw-0		rw-0		rw-0		r0(w)		rw-1		rw-0		rw-0	
								0b = Watchdog timer is not stopped. 1b = Watchdog timer is stopped.							
6-5		WDTSEL		RW		0h		Watchdog timer clock source select 00b = SMCLK 01b = ACLK 10b = VLOCLK 11b = X_CLK; VLOCLK in devices that do not support X_CLK							
4		WDTTMSSEL		RW		0h		Watchdog timer mode select 0b = Watchdog mode 1b = Interval timer mode							
3		WDTCNTCL		RW		0h		Watchdog timer counter clear. Setting WDTCNTCL = 1 clears the count value to 0000h. WDTCNTCL is automatically reset. 0b = No action 1b = WDTCNT = 0000h							
2-0		WDTIS		RW		4h		Watchdog timer interval select. These bits select the watchdog timer interval to set the WDTIFG flag and/or generate a PUC. 000b = Watchdog clock source /(2^31) (18h:12m:16s at 32.768 kHz) 001b = Watchdog clock source /(2^27) (01h:08m:16s at 32.768 kHz) 010b = Watchdog clock source /(2^23) (00h:04m:16s at 32.768 kHz) 011b = Watchdog clock source /(2^19) (00h:00m:16s at 32.768 kHz) 100b = Watchdog clock source /(2^15) (1 s at 32.768 kHz) 101b = Watchdog clock source /(2^13) (250 ms at 32.768 kHz)							

2019/3/1

52



34

1.2.3 ADC

利用 ADC12 模块进行模数转换，查阅手册，采用序列通道多次查询的方式，利用中断获得转换结果并进行相应的处理。

[1] ADC12CTL0 转换控制寄存器 0

15	14	13	12	11	10	9	8
SHT1x				SHT0x			
7	6	5	4	3	2	1	0
MSC	REF2_5V	REFON	ADC12ON	ADC12OVIE	ADC12TOVIE	ENC	ADC12SC

[2] ADC12CTL1 转换控制寄存器 1

15	14	13	12	11	10	9	8
CSTARTADDx				SHSx		SHP	ISSH
7	6	5	4	3	2	1	0
ADC12DIVx				ADC12SSELx		CONSEQx	ADC12 BUSY

[3] ADC12MCTLx 通道储存控制寄存器

7	6	5	4	3	2	1	0
EOS	SREFx			INCHx			

[4] ADC12MEMx 通道储存寄存器

15	14	13	12	11	10	9	8
0	0	0	0	Conversion Results			
7	6	5	4	3	2	1	0
Conversion Results							

该组寄存器为 12 位寄存器，用来存放 A/D 转换结果，其中只用到了低 12 位，高 4 位为 0。

[5] ADC12IFG 中断标志寄存器

15	14	13	12	11	10	9	8
ADC12IFG15	ADC12IFG14	ADC12IFG13	ADC12IFG12	ADC12IFG11	ADC12IFG10	ADC12IFG9	ADC12IFG8
7	6	5	4	3	2	1	0
ADC12IFG7	ADC12IFG6	ADC12IFG5	ADC12IFG4	ADC12IFG3	ADC12IFG2	ADC12IFG1	ADC12IFG0

ADC12IFGx: 中断标志位

对应于 ADC12MEMx，当 A/D 转换完成后，数据被存入 ADC12MEMx，此时 ADC12IFGx 标志置位。

[6] ADC12IE 中断控制寄存器

15	14	13	12	11	10	9	8
ADC12IE15	ADC12IE14	ADC12IE13	ADC12IE12	ADC12IE11	ADC12IE10	ADC12IE9	ADC12IE8
7	6	5	4	3	2	1	0
ADC12IE7	ADC12IE6	ADC12IE5	ADC12IE4	ADC12IE3	ADC12IE2	ADC12IE1	ADC12IE0

ADC12IEx: 中断允许位

对应于 ADC12IFGx，如果 ADC12IEx 允许，则当 ADC12IFGx 置位时会进入 ADC12 的中断服务程序。

[7] ADC12IV 中断向量寄存器

由于 ADC12 是一个多源中断，有 18 个中断标志，但是只有一个中断向量。则 18 个中断标志按照优先级安排对中断标志的响应。

ADC12IV 内容	中断源	中断标志	优先级
0x0000	无中断	无	无
0x0002	ADC12MEMx 溢出	ADC12OV	最高
0x0004	转换时间溢出	ADC12TOV	
0x0006	ADC12MEM0	ADC12IFG0	
0x0008	ADC12MEM1	ADC12IFG1	
0x000a	ADC12MEM2	ADC12IFG2	
0x000c	ADC12MEM3	ADC12IFG3	

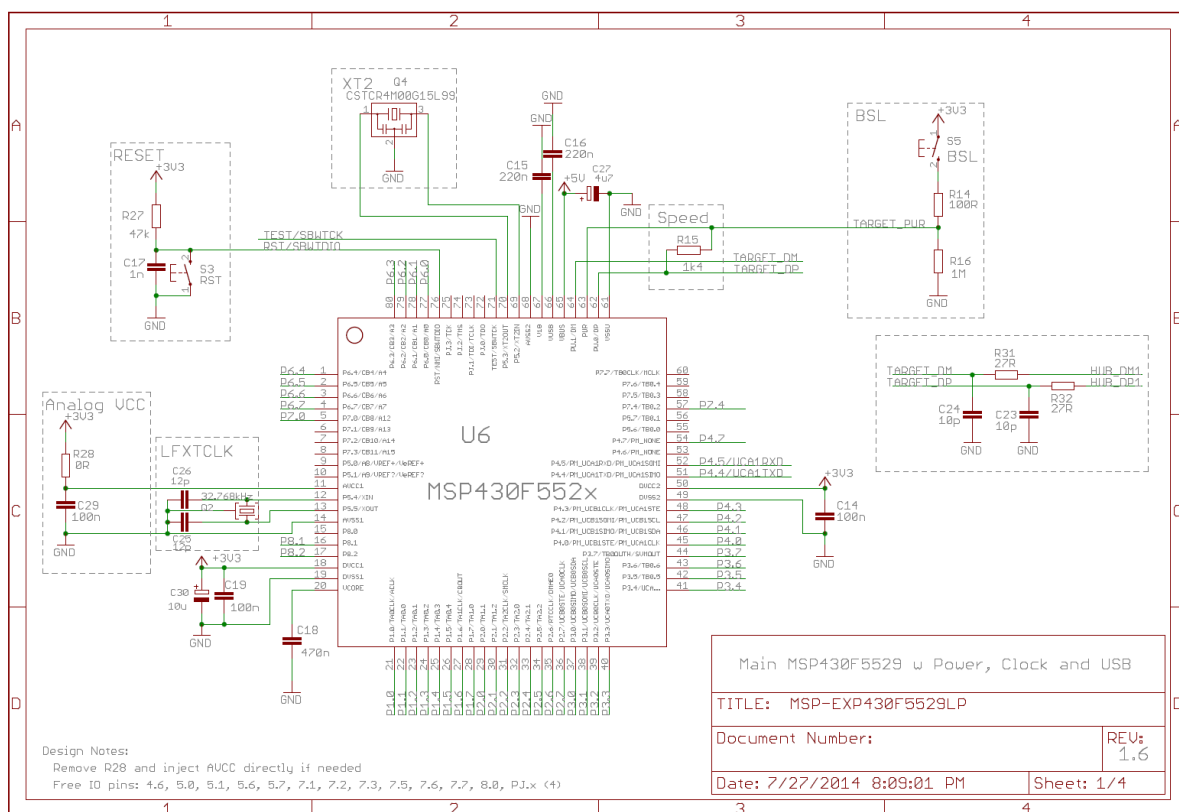
```
void ADC12_SCSC(void) {
    ADC12CTL0 |= ADC12SHT03 + ADC12MSC + ADC12ON;
    ADC12CTL1 |= ADC12SHP + ADC12CONSEQ_3 + ADC12CSTARTADD_0;
    ADC12MCTL0 |= ADC12INCH_0; // 麦克风
    ADC12MCTL1 |= ADC12INCH_1 + ADC12EOS; // 光敏电阻
    //ADC12MCTL2 |= ADC12INCH_5 + ADC12EOS; // 拨盘
    //ADC12IE = 0x0007;
    ADC12IE = 0x0003;
    ADC12CTL0 |= ADC12ENC + ADC12SC;
}

#pragma vector = ADC12_VECTOR
__interrupt void ADC12(void) {
    volatile unsigned int value_buf;

    switch(ADC12IV) { // 判断中断向量
        case 6: // 0x0006为MEM0 // 麦克风
            value_buf = ADC12MEM0;
            lightCtrl((value_buf - 1800) / 400);
            //ledCtrl((value_buf - 1800) / 400);
            break;
        case 8: // 0x0008为MEM1 // 光敏电阻
            value_buf = ADC12MEM1;
            //lightCtrl(value_buf / 700);
            ledCtrl(value_buf / 2000);
            break;
        // case 10: // 0x0008为MEM2
        //     value_buf = ADC12MEM2; // 拨盘
        //     break;
        default:
            break;
    }
}
```

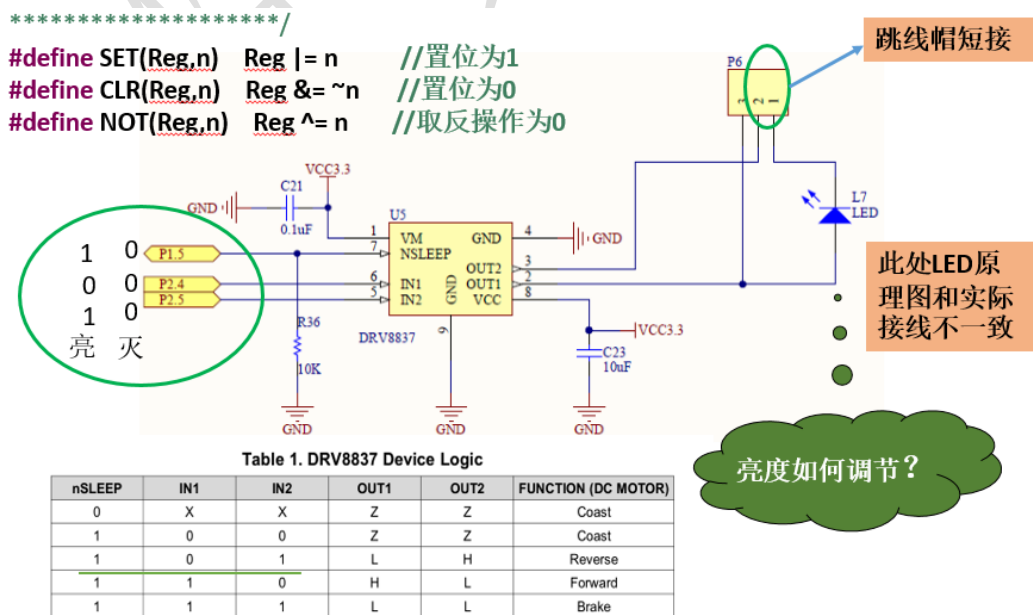
1.3 单片机硬件

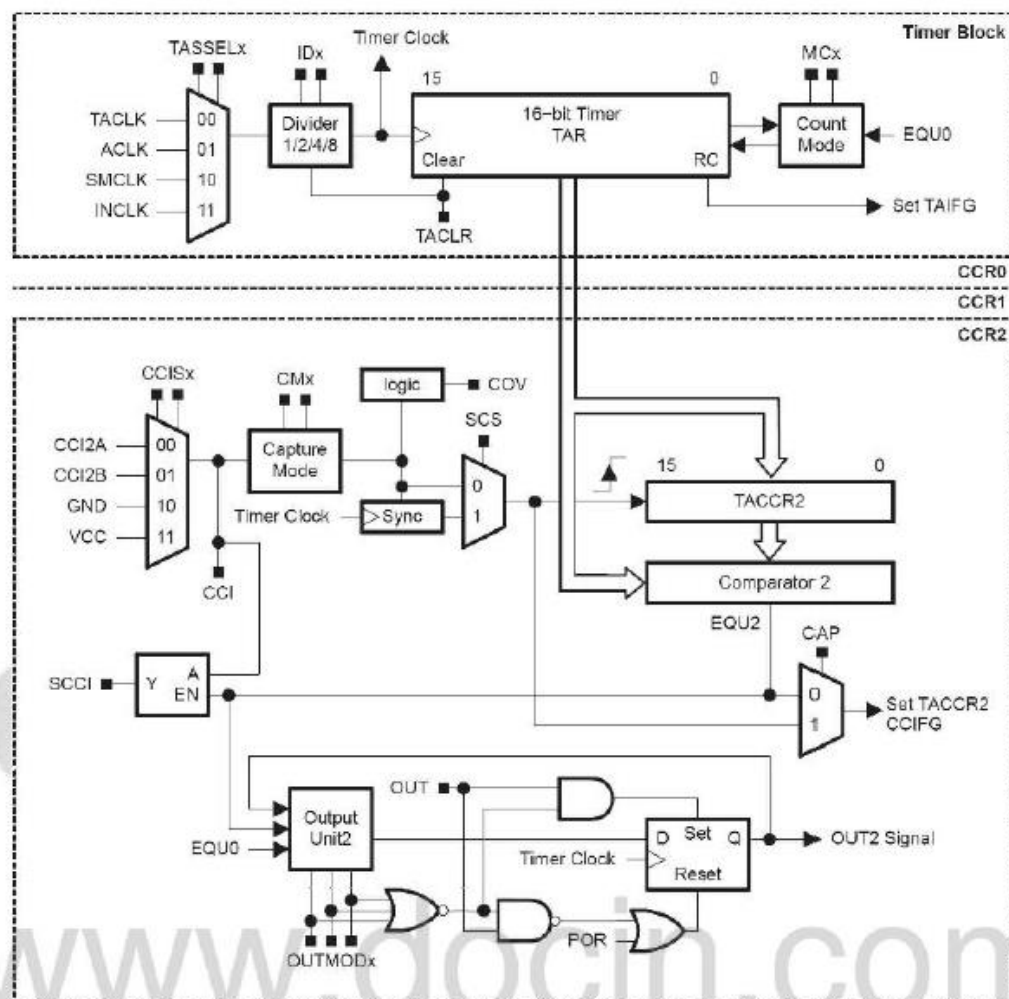
（原理图、接线图等，要阐明具体引脚并与程序对应）



1.3.1 LED 和定时器 A

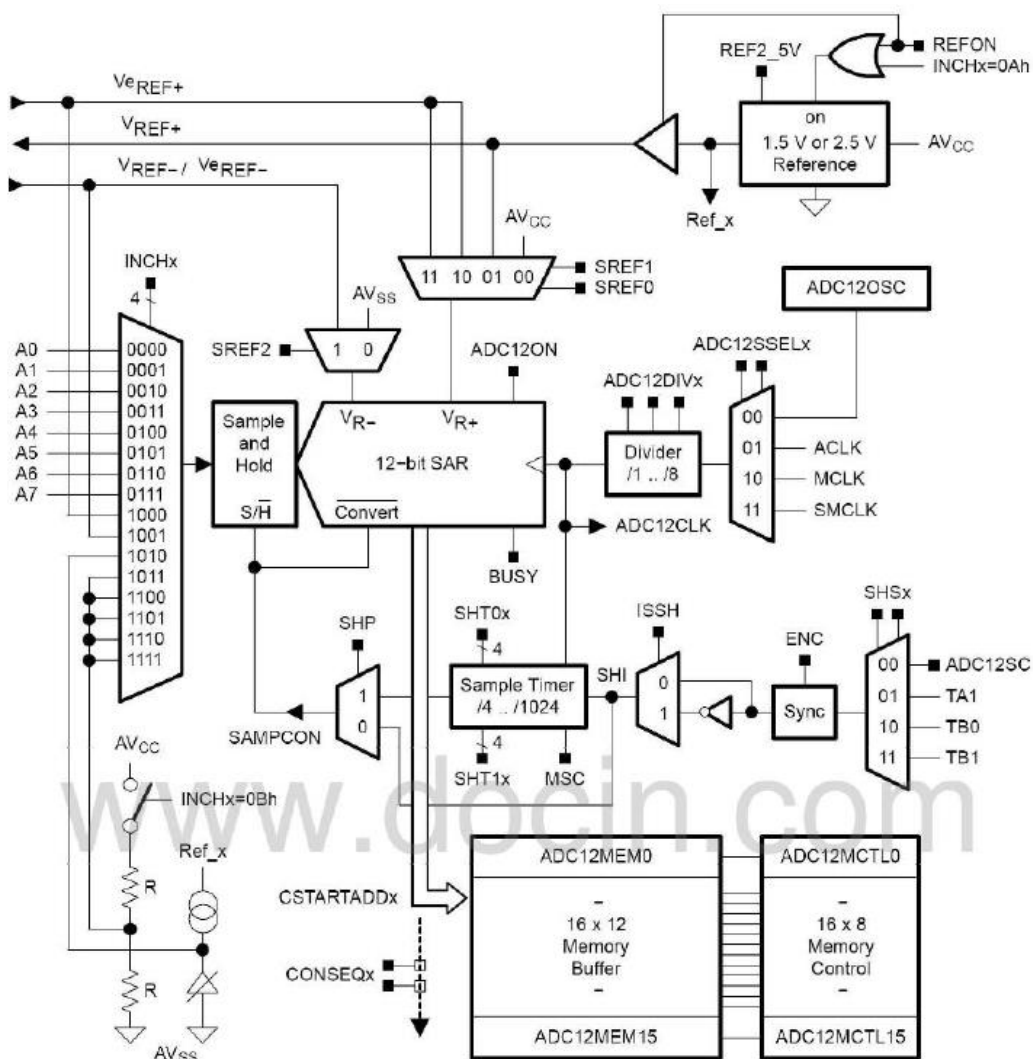
TA2CCTL1 的输出为 P2.4, TA2CCTL2 的输出为 P2.5, 通过设定输出使能即可利用 PWM 波的占空比控制大功率 LED 的亮度。





1.3.2 ADC12

利用跳冒线和杜邦线将麦克风输出接到 P6.0, 将光敏电阻接到 P6.1 端口, 查阅上面的接线图可以知道, 这两个端口分别对应着 ADC12INCH_0 和 ADC12INCH_1。将数据分别储存在 ADC12MCTL0 和 ADC12MCTL1 对应的 ADC12MEM0 和 ADC12MEM1 中, 即可获得当前的音量和光强。



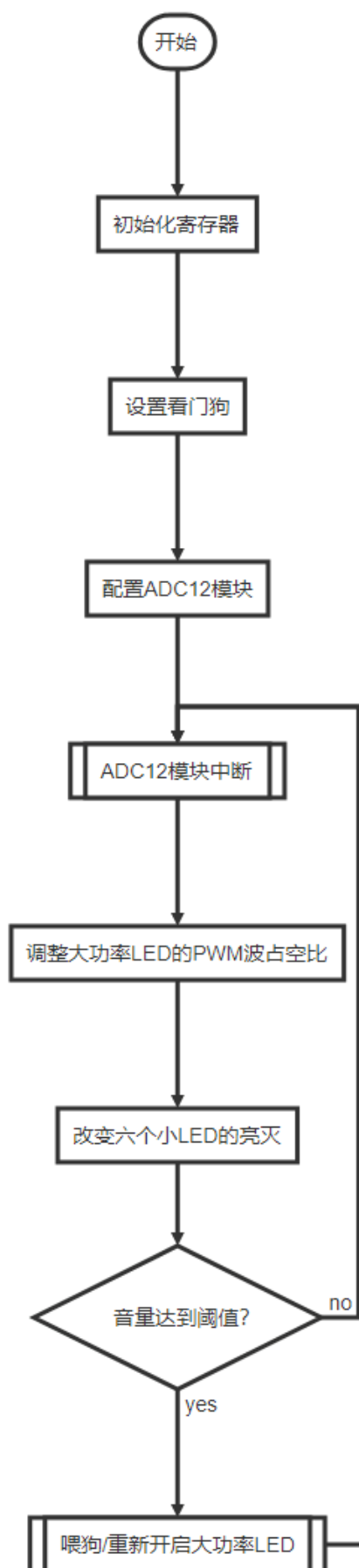
1.4 程序设计思路

(详细叙述程序设计思路和流程图)

第一，利用看门狗定时器做一个 16s 的定时，控制大功率 LED 的亮灭，当声音音量大小达到某一阈值时，喂狗，则重新计时，当大功率 LED 关闭后，只有音量大小达到阈值后，LED 才会重新开启。另外也做了两个按键开关辅助控制大功率 LED 的亮灭。

第二，使 ADC12 模块工作在序列通道多次查询模式，利用中断法获得转换结果并进行相应的处理。对光强信号，根据其大小控制 PWM 波的输出占空比，以控制大功率 LED 的亮度；对声音信号，其音量大小实时显示在六个小 LED 上，当达到 3 格响度时即喂狗，达到 4 格响度时即可尝试重新开启大功率 LED。

下面是主体的流程图（省去看门狗中断和按键中断控制 LED 开关部分）：



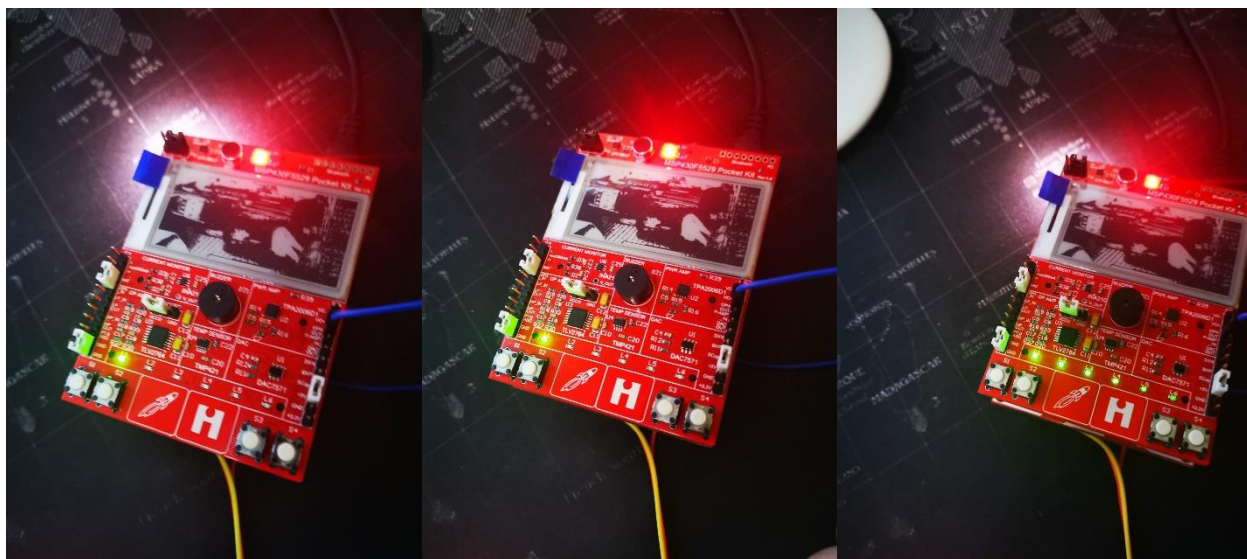
1.5 实验结果

（实验结果文字阐述，按键、灯效果展示，示波器图形展示等，手画亦可）

安静较暗状态

看门狗关灯

响度阈值开灯



1.6 实验中遇到的问题和解决方法？

首先，根据 PPT 内的例程，利用 ADC12 模块的单通道单次转换模式完成了单个影响因素的 LED 控制，但是将两个程序合并到一起以后发现两路信号产生了干扰，重新接线改变输入端口无明显效果。考虑到序列通道单次转换模式下没有加时间控制，可能导致混叠，遂改用中断查询方式做 ADC12 模块的多次查询模式。查阅资料发现可以指定采样时间，于是去掉了多次采样取平均值的滤波函数。一(做)顿(到)操(心)作(态)以(爆)后(炸)，把两块单通道多次转换的代码合到一起，实现了序列通道多次查询的方式，利用中断获得转换结果并进行相应的处理。

然而出现了一个新的小问题，PWM 波的占空比参数有点不对了，LED 工作在正常状态时占空比调的却很小，改了一下也没改正常。有两个可能，一是采样时间的改变同时会影响输出的值，但是 debug 模式下查看寄存器发现应该是差不多的；而是 TA2CCTL2 通道没有用，而且输出接到了 P2.5，但是去掉二者中任一 LED 灯都不亮。没(又)有(一)分(次)析(做)出(到)来(心)为(态)什(爆)么(炸)。

1.7 实验体会与建议

这次实验，我们一步一步完成了展馆灯光综合实验。首先利用 PWM 调光实现了呼吸灯的功能，然后接入光敏电阻，使灯的亮度随外界环境亮度而变化，再更改输入信号，使可以用声音大小进行控制，最后将上述模块综合起来，实现了用光强信号控制 PWM 波的输出占空比，以控制大功率 LED 的亮度，并用声音音量大小控制大功率 LED 的开关的功能，锻炼了我们自主学习查阅资料实现各个功能的综合能力。