浙江大学

课程名称 电子电路系统设计与调试实践

任课教师 _____

论文题目 太阳能景观灯控制器设计

姓　　名 _____

学　　号 _

年级专业 _____

所在学院 信息与电子工程学院

目录

1 任务目标：

本实验旨在使用合适的外设器件与 STM32 单片机，实现太阳能景观灯控制器的设计，具体要求如下：

•实现对电池的充电控制，电池电压达到 8.2V 时停止充电，电池电压低于 7.8V 时继续充电。

• 实现对 LED 灯的用电控制，当电池电压低于 6V 时，停止供电，电池电压高于 6.4V 时继续供电。

• 对 LED 进行恒流驱动，电流 300mA，控制精度小于 15mA。

• 系统工作状态有显示(LCD 屏)。

• 当电池电压达到 8.2V 时，采用 PWM 方式控制进行恒压限流充电，当电流低于 100mA 时停止充电。

• 通过设定实现 LED 的亮度调节，调节步进小于 50mA。

• 实现景观灯的光控功能，有白天灭，夜间点亮。

除上述课程要求的功能之外，本小组设计的太阳能景观灯控制器外接了蓝牙模块，可以通过蓝牙远程遥控景观灯的亮灭以及景观灯时控等需求。

2 系统方案

2.1 系统总体框图：

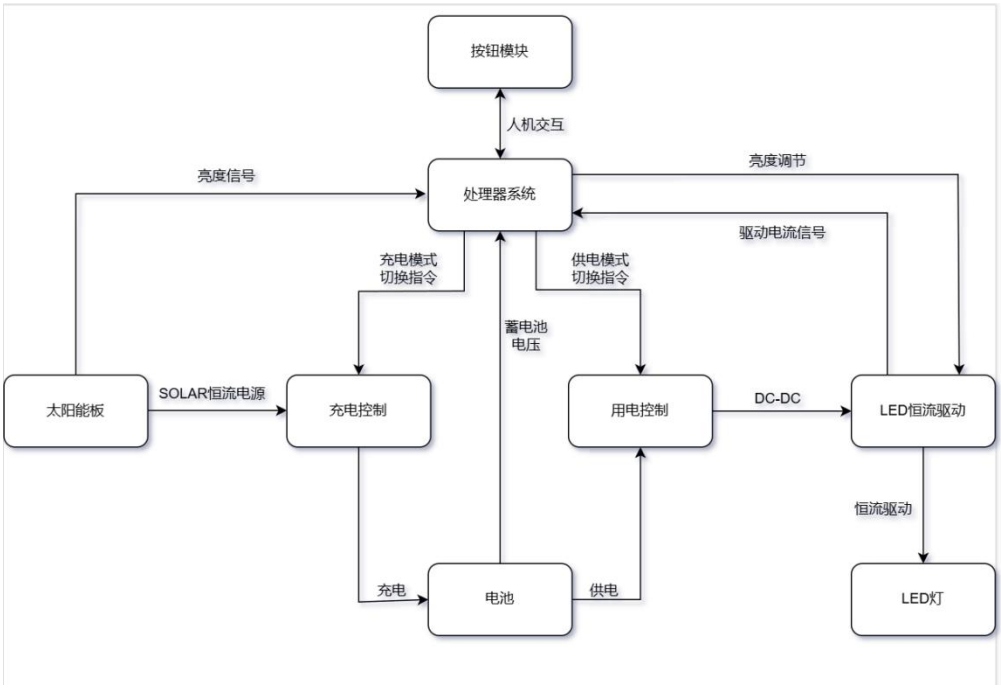

图 1 系统总体框图

上述系统框图为景观灯控制器设计任务要求的解读。通过上述框图，我们可以基本了解到充电控制、供电控制、人机交互几大主要模块的工作流程。

2.2 方案比较与选择

在上述系统框图的基础上，小组同学进行了调研，结合理论知识，基本确定了 4 个 ADC

信号采集端口，2 个 PWM 输入控制端口。

在电流控制方面，为方便观测，本小组采用了采样电阻结合运放的观测方案，并合理添加滤波电容，旨在解决小信号易受噪声干扰的问题。与此同时，使用 PWM 信号结合 MOS 管对电流进行限流控制，从处理器到被控对象之间的所有信号都是数字形式的，无需再进行数模转换过程，对噪声的抗干扰能力大大增强。

在电压控制方面，为满足单片机输入电压的限制要求，本小组采用了电阻分压的电压采集方式，通过电阻分压采集，再经软件数据校准，我们可以通过单片机较为准确地采集与处理电压值，供后续控制过程使用。

在人机交互方面，本小组采用了 4 个按键控制系统，并选择使用蓝牙控制的方式使得太阳能景观灯可控性更强：夜间无人时段，可操控景观灯停止供电，系统消耗的能源更少，更加的绿色环保。

2.3 方案描述

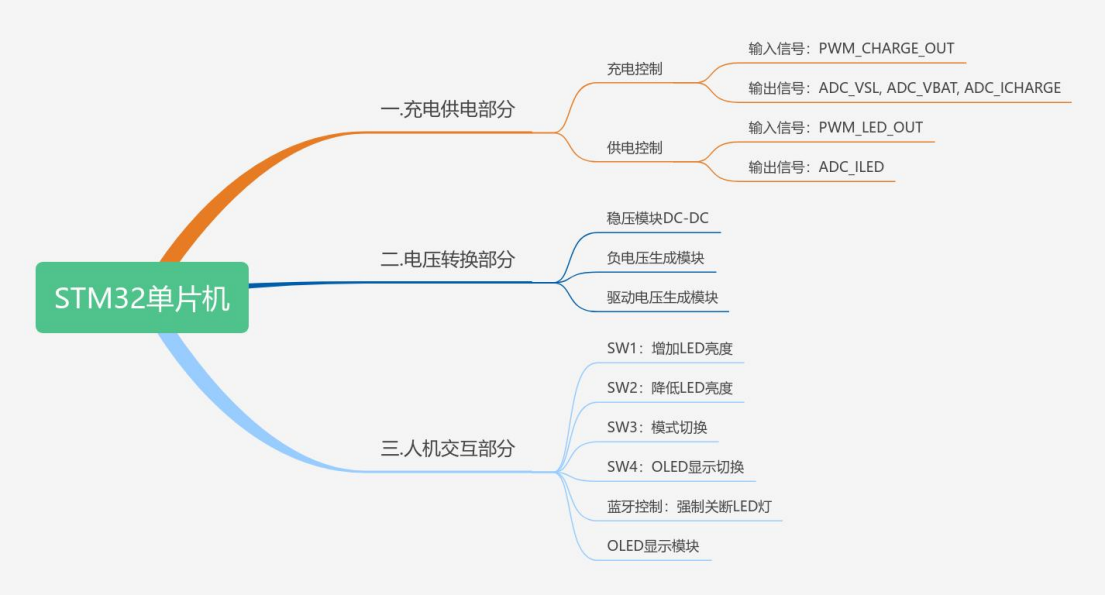

图 2 电路方案描述

电路总体设计方案包括以下三个部分：充电供电部分是主要部分，负责控制充放电状态及其电流；电压转换部分是基础部分，负责为不同器件的工作提供供电电压与驱动电压；人机交互部分则主要为使用者操纵提供方便。

在充电供电部分，我们先采集太阳能电池板两端的电压，判断其是否处于白天。若处于白天，则停止供电，检测电源电压，电源电压低于 7.8V 继续恒流充电，充电过程中充电电压达到 8.2V 时改为恒压限流充电，再利用采样电阻检测充电电流，电流小于 100mA 时停止充电。若处于黑夜，则停止充电开始供电，并检测电源电压，电压低于 6V 时停止供电，电压高于 6.4V 时重新恢复供电；在黑夜供电状态下，可使用蓝牙模块切换模式，强制其停止供电。这样就可以完整实现一个太阳能景观灯控制器系统。

3 理论分析与计算

3.1 器件选取

• 分压电阻：用于采样电压，为满足电压由 8V 降到 3.3V 以下的要求，选取的分压电阻阻值为 100kΩ与 47kΩ。

• 采样电阻：已提供 100mΩ的采样电阻

• 运放选择：采样充电与供电电流时，采样电阻两端电压较小，因此设置放大系数分别为 56 倍与 120 倍的双运放 LM358 进行放大。

• 太阳能电池板：已提供，开路电压为 12V，恒流 0.5A.

3.2 Multisim 关键电路仿真

• PWM 信号控制仿真

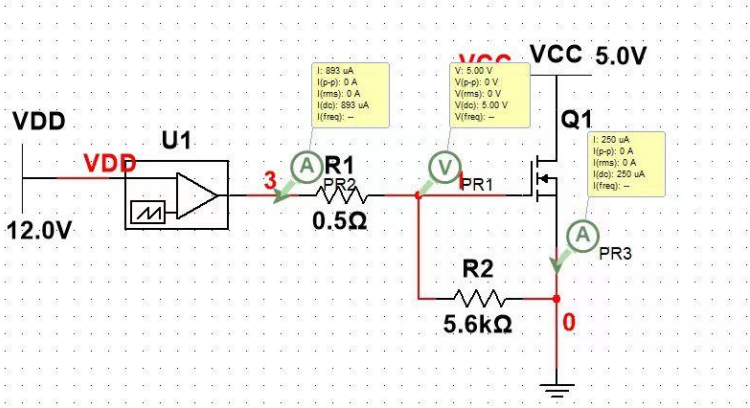

图 3 PWM 控制仿真

通过构造 PWM 信号控制电流，修改 PWM 的占空比，可以观测主干电路的电流变化，检验 PWM 信号对电流的控制作用。

• 采样电路仿真

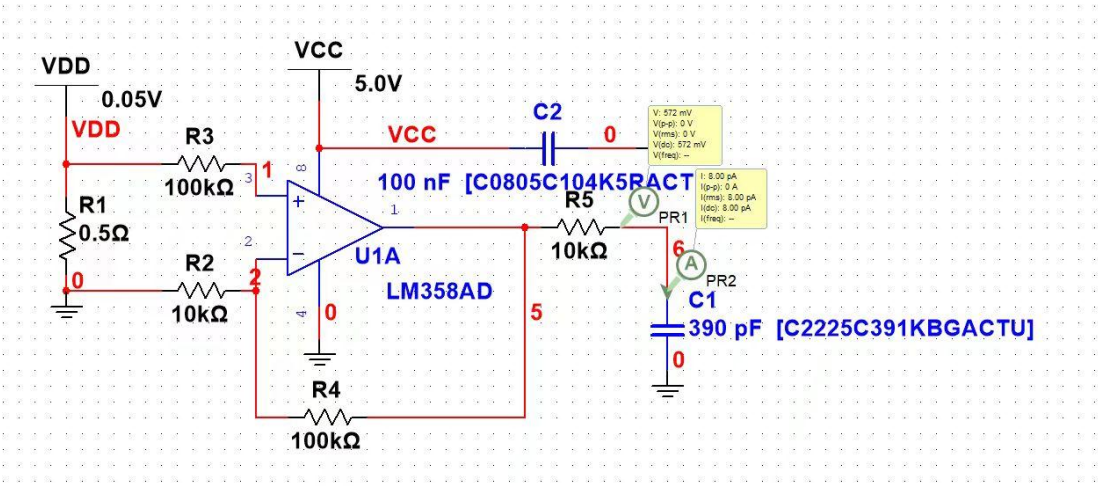

图 4 采样电路

通过对构造的采样电路进行测试，可以验证运放的放大功能与电路的采样功能。

• 驱动电压转换仿真

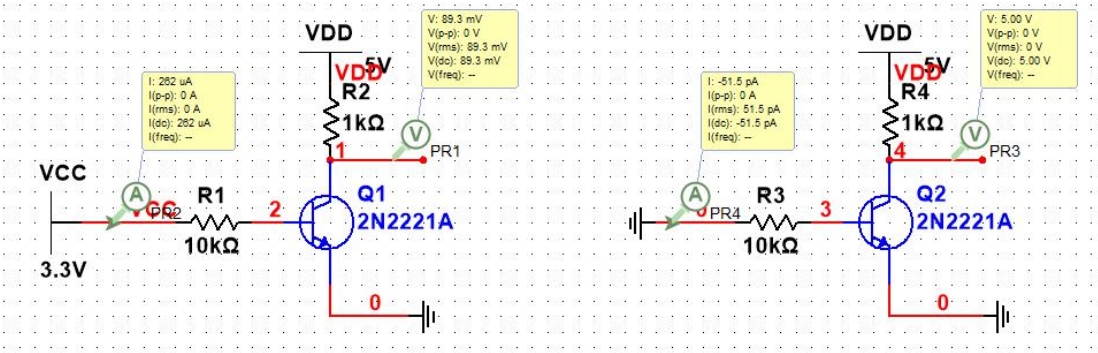

图 5 驱动电压转换仿真

通过三极管的转换，电路可以实现反相输出 0V 与 5V，驱动 MOS 管。

• PWM 滤波仿真：

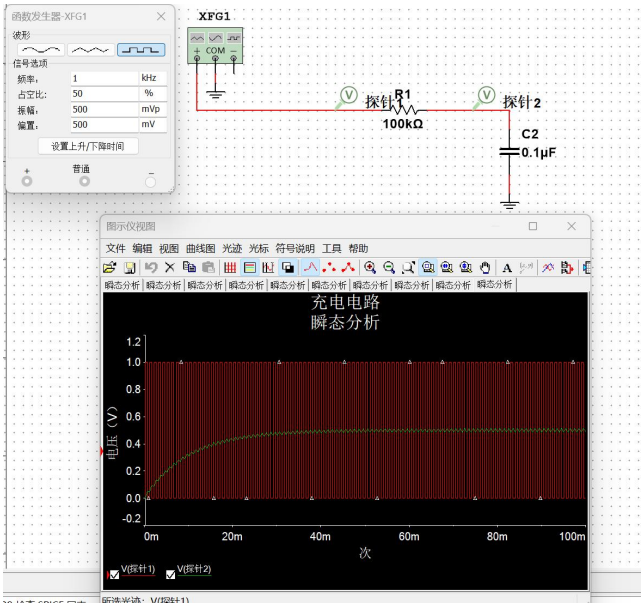

图 6 滤波器仿真

在低通滤波器的作用下，PWM 信号作用下，电路电压稳定在最大值×占空比的位置。

3.3 输出信号（单片机输入）理论值计算

• ADC_VSL：反映太阳能电池板两端电压

查询数据手册，二极管导通压降：

$$V_F = 0.6V$$

所以可得出太阳能电池板两端电压范围：

$$V_{SOLAR+} - V_{SOLAR-} = [8.4, 8.8]$$

根据如下公式可计算充电状态下太阳能电池板两端电压：

$$V_{ADC\_VSL} = (V_{SOLAR+} - V_{SOLAR-}) \times \frac{R_4}{R_3 + R_4}$$

$$V_{ADC\_VSL} = 2.75V$$

因此，可配置光控模式下停止供电的阈值电压为：

$$V_{ADC\_VSL} \approx 2V$$

• ADC_VBAT：反映电池两端电压

根据如下公式，可计算并配置 PWM 控制电池充放电时的电压

$$V_{ADC\_VBAT} = (V_{CC} - V_{BAT-}) \times \frac{R_7}{R_6 + R_7}$$

开启恒流充电时（≦7.8V）：

$$V_{ADC\_VABT} \leq 2.49V$$

切换为恒压限流充电时（≧8.2V）：

$$V_{ADC\_VABT} \geq 2.62V$$

停止供电时（≦6V）：

$$V_{ADC\_VABT} \leq 1.92V$$

继续供电时（≧6.4V）：

$$V_{ADC\_VABT} \geq 2.05V$$

• ADC_ICHARGE：通过采样电阻反映充电电流

$$V_{ADC\_ICHARGE} = A_{CHARGE} \times I_{CHARGE} \times R_{SAMPLE}$$

恒压限流充电（≦100mA）:

$$V_{ADC\_CHARGE} \leq 1.2V$$

• ADC_ILED：对 LED 进行恒流驱动

$$V_{ADC\_ILED} = A_{LED} \times I_{LED} \times R_{SAMPLE}$$

恒流范围（285mA≦I≦315mA）:

$$1.596\text{V} \leq V_{ADC\_ILED} == 1.68V \leq 1.764V$$

4 电路与程序设计

4.1 硬件电路设计

4.1.1 各模块电路

• 充电电路模块

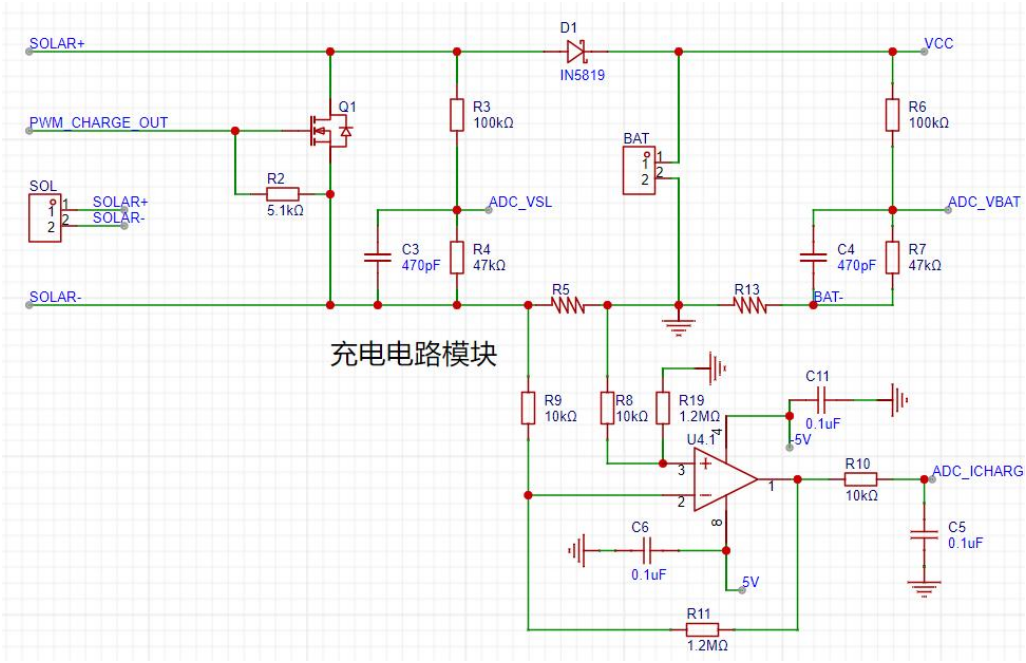

图 7 充电电路模块

充电电路模块利用太阳能电池板对电池进行充电，设计 ADC_VSL 与 ADC_VBAT 两个信号分别检测充电电压与电池电压，通过 ADC 接口引脚传回到单片机，便于单片机通过其传回的值做出相应的判断。

设计一个 MOS 管控制充电的开始与结束，充电时 MOS 管关断，太阳能板电压加到电池两端；反之则 MOS 管导通，0.5A 电流在 MOS 管中恒定导通，此时 MOS 管漏源电压接近于 0。与此同时，MOS 管栅极采用 PWM 信号，通过调节信号的占空比，可以控制充放电电流的大小。

设计二极管 D1 作保护，在未充电状态下保护电池电压，避免电池两端电压流失。

设计采样电阻 R5 采集充电电流，两端使用 LM358 作差分放大，将采样电阻两端电压放大 120 倍数，并在信号出口作低通滤波，防止 PWM 方波信号的干扰使得 ADC 电压采集不稳定。

• 恒流 LED 模块（供电模块）

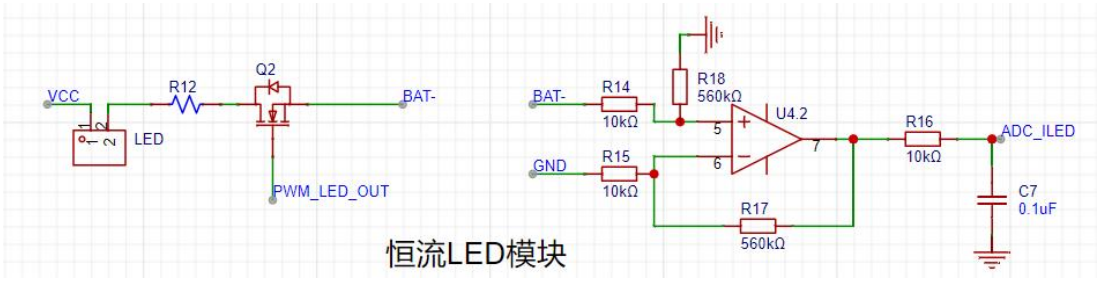

图 8 恒流 LED 模块

恒流 LED 模块主要实现 LED 供电，电流恒定在 300mA 左右，LED 发光功率 1W 左右，因此设计 LED 限流保护电阻 R12 为 4Ω左右。

设计一个 MOS 管控制 LED 灯的亮灭，采用 PWM 信号控制，通过调节 PWM 信号的占空比，改变流过 LED 的电流，从而改变 LED 的亮度。

设计采样电阻 R13（见图 3），通过运放 LM358，电压放大 56 倍，并通过低通滤波稳定电压信号，传递给单片机作处理

• DC - DC 模块

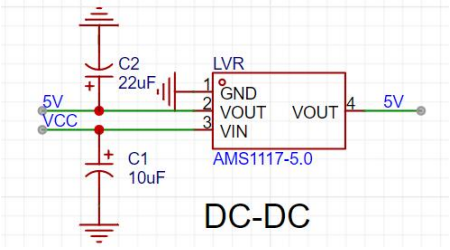

图 9 VCC—5V 转换

利用 AMS1117 稳压芯片，将电池电压转换为稳定的 5V 电压，为单片机与运放供电。

• 负电压生成模块

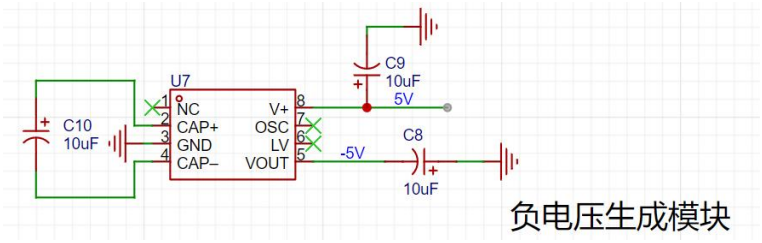

图 10 负电压生成模块

通过电压转换芯片 ICL7660，可将 5V 电压转换为-5V 电压，用于给双运放 LM358 供电。

• 电压驱动模块

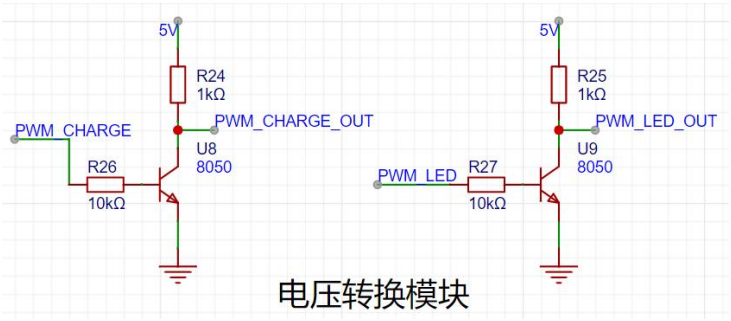

图 11 电压转换模块

由于单片机提供的电压最大值为 3.3V，而 MOS 管驱动电压V_{TH}在 4V 左右，为确保 MOS 管可以正常工作，需要利用三极管 8050 将 3.3V 电压转换为 5V，在此过程中需要注意，PWM_..._OUT 是 PWM_...的反相。

• 按键控制模块

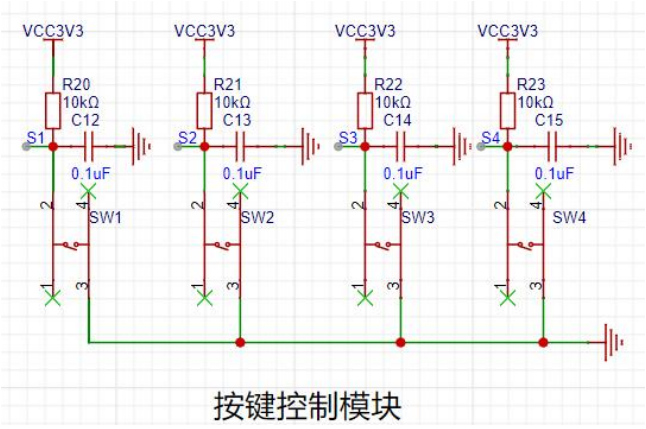

图 12 按键控制模块

设计四个按键，分别上拉接高电平，并采用 0.1uF 电容滤波。四个按键的功能分别为向上调节 LED 亮度、向下调节 LED 亮度、模式切换、OLED 显示切换。

• OLED 显示模块

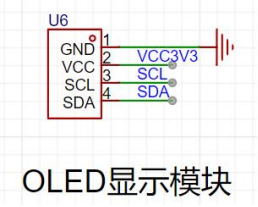

图 13 OLED 显示模块

• 蓝牙模块

实验附加功能，并未在原理图中画出，后期通过排针与杜邦线连接。

• 核心控制器部分与引脚配置

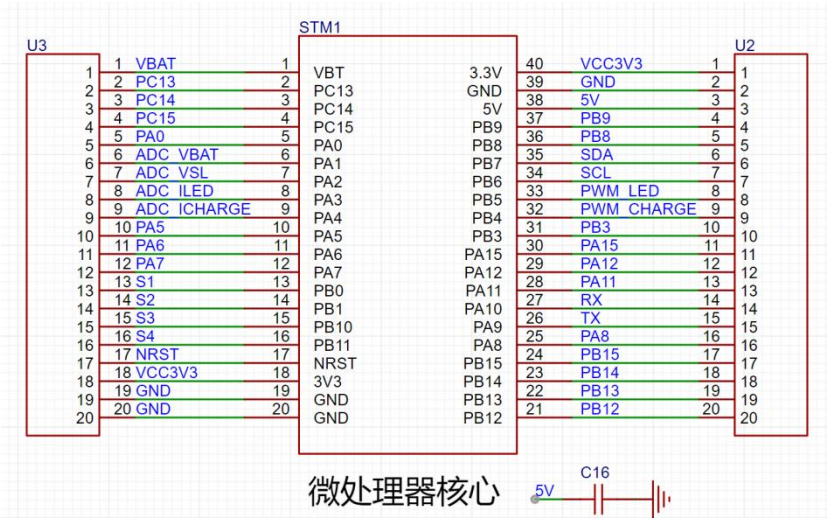

图 14 微处理器核心

在 STM32F103C8T6 中，PB6, PB7 属于 I2C 接口，可用于配置 ILED 模块；PA1~PA4 可进行 ADC 采样，用于接收 ADC 信号；PB4, PB5 连接定时器 3，可以输出 PWM 波；PA9, PA10 为串口通信部分，可用于蓝牙模块的连接；PB0, PB1, PB10, PB11 接口接按键，有正常 GPIO 功能即可。

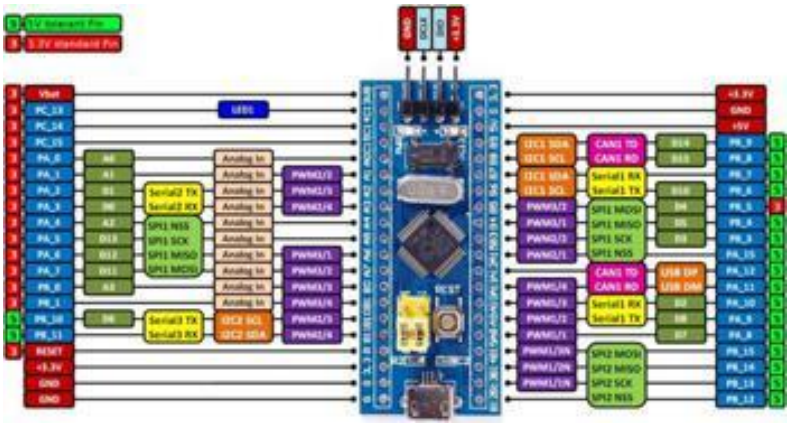

图 15 引脚配置图

4.1.2 PCB 设计

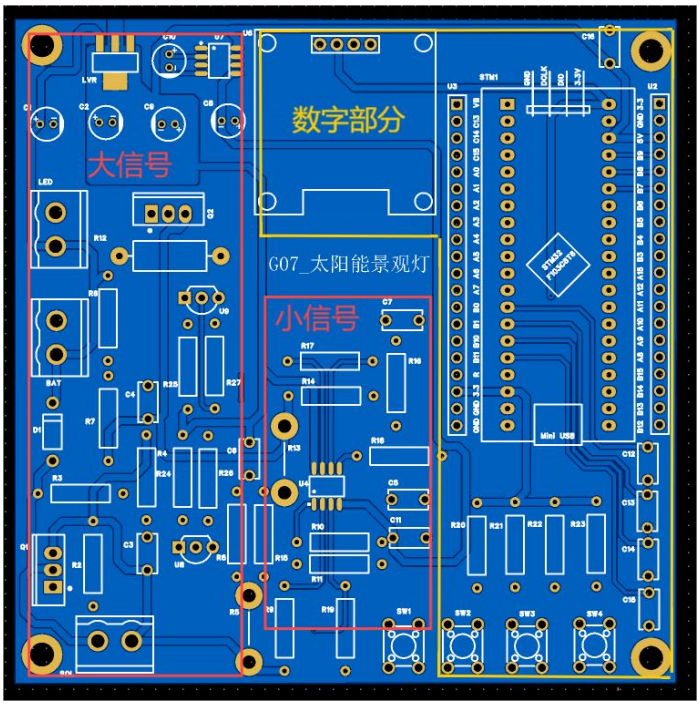

图 16 PCB 平面图

在 PCB 设计的过程中，器件摆放遵循了模拟部分与数字部分分开，大信号与小信号分开的原则，以确保电路之间尽可能地不受干扰。

在布线的过程中，充分考虑电流大小，电源相关的电线与恒流 LED 部分采用 40mil 线宽，其余部分采用 20mil 线宽。与此同时，电路走线尽可能走水平线与垂直线，布线较为规整，转弯处采用 45 度斜线连接，避免电流冲击问题。

在铺铜方面，绝大部分区域采用正反面铺铜接地，利于优化走线方式；稳压模块附近采用+5V 铺铜，利于稳压模块散热。

在人机交互方面，电路所有器件的丝印标签都是正的，OLED 显示屏在上方居中，按键在右下角四个位置，方便使用者操作。

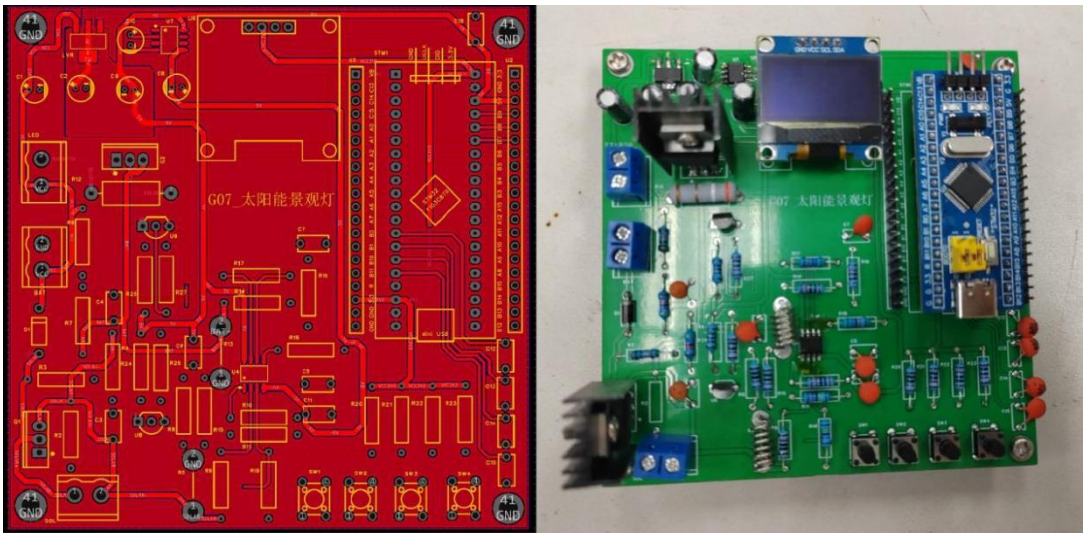

图 17 PCB 与实物安装图

4.2 软件设计

4.2.1 流程图

软件部分程序实现了对电池充电和放电的智能控制，通过 ADC 采样、PWM 控制、蓝牙通信、按键输入和定时器等功能，系统能够根据不同的输入和条件，自动调整充电和放电的状态，并通过 OLED 屏幕实时显示相关信息。整个系统具有高度的灵活性和可配置性，能够满足不同应用场景的需求。流程图如下：

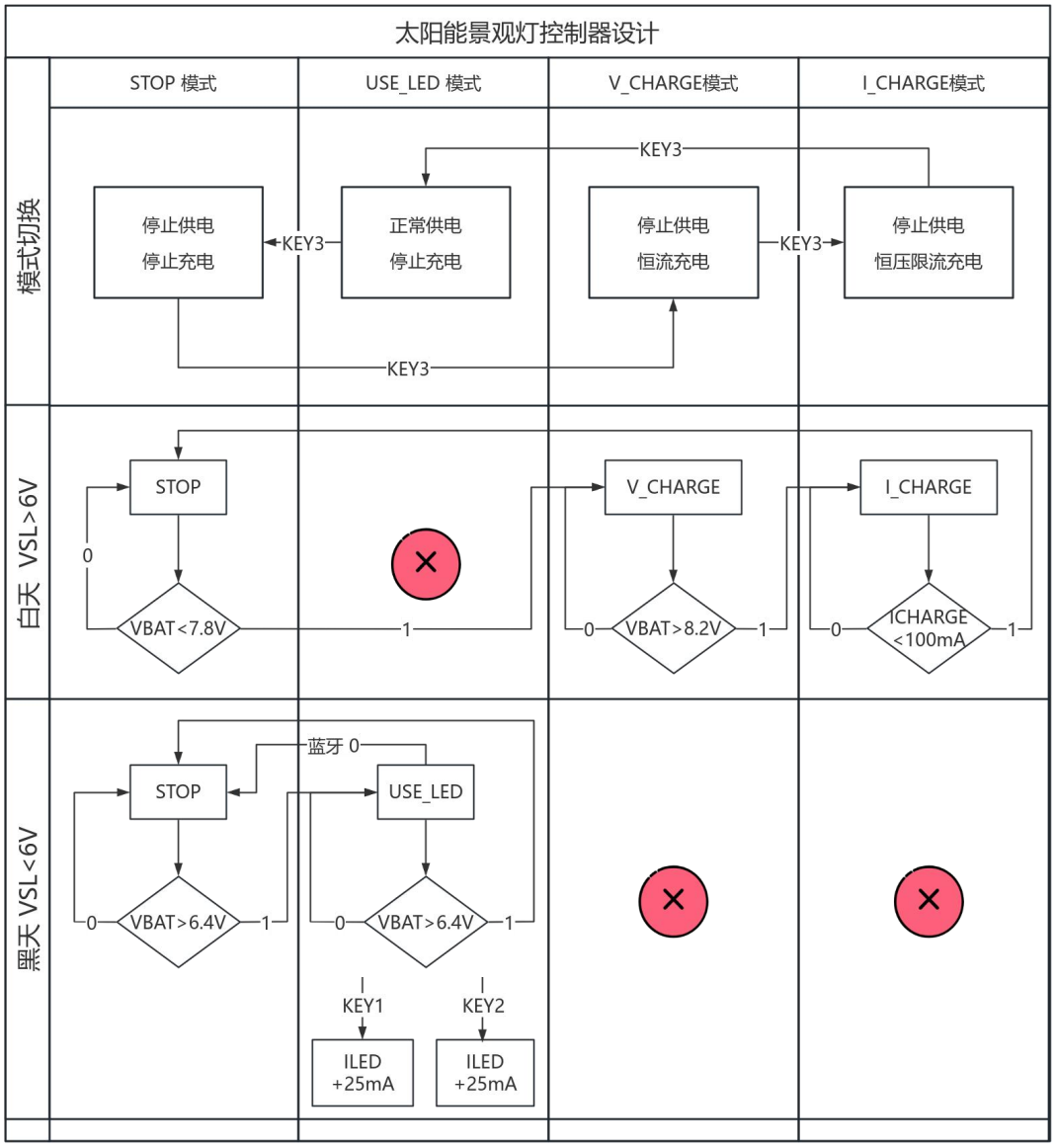

图 18 软件设计流程图

对流程图各部分的功能解释如下：

• OLED 显示模块：负责显示电池电压、太阳能电压、LED 电流、充电电流以及当前工作状态等信息。

• ADC 采样模块：对电池电压、太阳能电压、LED 电流和充电电流进行采样，并将采样值用于后续的逻辑判断和控制。

• PWM 控制模块：通过 PWM 信号控制 LED 和充电器的亮度/电流。
• .蓝牙通信模块：接收蓝牙信号，并根据信号内容调整系统状态。
• 按键输入模块：通过按键输入改变 LED 电流、切换工作模式等。
• 定时器模块：用于控制 ADC 采样频率、OLED 刷新频率、按键扫描频率等。

4.2.2 模块核心代码

• 初始化系统

对系统的初始化主要包括以下几个方面：

a. 初始化 OLED、ADC、PWM、Timer 和 Serial 等模块的功能，主要包括引脚配置、功能选择。
b. 设置引脚 PC13 为开漏输出，方便使用板载 LED 用于调试
c. 预先在 OLED 中显示静态字符串。
d. 在无限循环中，根据系统状态进行不同的操作。

```
int main(void)
{
 /*模块初始化*/
 OLED_Init();     //OLED 初始化
 AD_Init();      //AD 初始化
 PWM_Init();      //PWM 初始化
 Timer_Init();     //TIME 初始化
 Serial_Init();    //Serial 初始化

 RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOC, ENABLE);  //开启 GPIOB 的时钟
 GPIO_InitTypeDef GPIO_InitStructure;
 GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_OD;
 GPIO_InitStructure.GPIO_Pin = GPIO_Pin_13;
 GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
 GPIO_Init(GPIOC, &GPIO_InitStructure);      //将 PC13 引脚初始化为开漏输出
 /*显示静态字符串*/
 OLED_ShowString(1, 1, "VBAT:xxxxmV");
 OLED_ShowString(2, 1, "VSL:xxxxmV");
 OLED_ShowString(3, 1, "MODE=");

 while (1)
 {
     ...
 }
}
```

在无限循环中，代码将根据系统**当前状态**与**系统输入**做出一系列的响应，如调整 PWM、刷新 OLED、采集 ADC 数据等。

为了提升可读性，我们使用四个宏定义分别表示系统的四种工作状态：

```
// main.h
#define state_stop       0 //待机模式
#define state_V_charge   1 //恒压充电模式
#define state_I_charge   2 //恒流充电模式
#define state_use_charge  3 //供电模式
```

并在主函数中变量 state 来表示当前系统状态。下面根据系统所处的状态，分别介绍核心代码的功能。

• 充放电逻辑设计

a. 状态设计

当处于充电模式下时，单片机需要将放电开关关断；当处于恒流放电模式下，单片机需要将放电开关打开，并将电流限制在一定范围内（10mA）。同时，太阳能天池板为电池的充电包括三个阶段：预充电、恒流充电、恒压充电，也需要单片机进行控制。

对充电、放电的控制使用脉宽调制信号（PWM）实现。我们使用变量 pwm_LED 和 pwm_CHARGE 分别表示施加在放电、充电开关上的 PWM 占空比（单位%）。在硬件电路的设计约束下，约定 pwm_LED 为 100 时表示放电开关完全打开，pwm_CHARGE=0 是表示充电开关完全打开。

b. 状态初始化

下面的代码根据当前状态的不同，对充电、放电开关设置不同的约束。

```
//-------------------------------------------------------不同状态初始化
   switch (state)
   {
   case state_stop:     // 停止充电，停止放电
    pwm_CHARGE = 0;
    pwm_LED = 100;
    break;
   case state_V_charge:   // 停止放电，按电压充电
    pwm_CHARGE = 100;
    pwm_LED = 100;
    break;
   case state_I_charge:   // 停止放电，恒压限流充电
    pwm_LED = 100;
```

```
14.       break;
15.      case state_use_charge: // 停止充电，恒流放电
16.       pwm_CHARGE = 0;
17.       break;
18.      default:
19.       break;
20.      }
21.      pwm_LED = (pwm_LED > 60) ? pwm_LED : 60;
22.      PWM_LED_SetCompare2((uint16_t)pwm_LED);
23.      PWM_CHARGE_SetCompare1((uint16_t)pwm_CHARGE);
```

c. 状态转换

根据要求，太阳能电池对蓄电池的充电分为恒压充电、恒流充电两部分。黑夜时，太阳能电池电压较小（小于 SL_limit，一般设置为电池板最大电压的一半，即 7V 左右），若蓄电池电压大于 6.4V 即可为 LED 供电；白天时，太阳能电池板可为蓄电池充电，当蓄电池电压小于 8.2V 时恒流快充，否则进行恒压充电以保护电池。

以下是状态转换部分的代码：

```
1.      //-------------------------------------------------------状态转换处理
2.       if (ADC_VSL <= SL_limit) // 采样到太阳能电压 <= 为电池供电的最低电压，应停止充电
3.       {
4.        if (state != state_stop && state != state_use_charge) // 如果不是停止充电或放电状态
5.         state = state_use_charge;         // 切换到放电状态。
6.        if (ADC_VBAT >= 6400)           // 电池电压大于6.4V
7.         state = state_use_charge;         // 切换到放电状态
8.        else if (ADC_VBAT <= 6000)          // 电池电压小于6V
9.         state = state_stop;           // 切换到停止状态
10.      }
11.      else // 采样到太阳能电压 > 为电池供电的最低电压，应充电
12.      {
13.       ` if (state == state_use_charge)             // 如果是放电状态
14.        state = state_stop;                // 停止放电
15.       if (ADC_ICHARGE <= 100 && ADC_VBAT >= 8200 && state == state_I_charge) // 充电电流小于100mA，电池电压大于8.2V，且为恒流充电状态时，应停止充电
16.        state = state_stop;
```

```
17.       else if (ADC_VBAT >= 8200 && state == state_V_charge) // 电池电压>=8.2V，且当前在恒压充电时，应转换为恒流充电
18.        state = state_I_charge;
19.       else if (ADC_VBAT <= 7800 && state == state_stop) // 电池电压小于7.8V，且为停止充电状态时，应恒压充电
20.        state = state_V_charge;
21.      }
```

为了更加贴合实际功能，我们还添加了系统总开关，使用变量 sys_flag 表示，便于启动、关停整个系统。

```
1.       if (sys_flag == 0) // 系统总开关
2.       {
3.        state = state_stop;
4.       }
```

• 定时器中断回调函数 TIM2_IRQHandler：

每当 TIM2 定时器计数溢出，会发起更新，调用此函数。定时器设置中断间隔时间 1ms，每 10ms 触发一次按键扫描，每 500ms 刷新一次 OLED，每 10s 扫描一次太阳能电池板电压。

```
1.     //------------------------------------------------------------TIM2 中断回调函数
2.     void TIM2_IRQHandler(void)
3.     {
4.      if (TIM_GetITStatus(TIM2, TIM_IT_Update) == SET)
5.      {
6.       CountTimer++;
7.       CountTimer10ms++;
8.       CountTimer500ms++;
9.       CountTimer30s++;
10.
11.      if (CountTimer > Delayms_V) //"Delayms_V", ADC 采样标志
12.      {
13.       CountTimer = 0;
14.       ADC_flag = 1;
15.      }
16.
17.      if (CountTimer10ms > 9) // 10ms,  按键采样标志
18.      {
19.       CountTimer10ms = 0;
20.       flag_KS = 1;
```

```
    }

    if (CountTimer500ms > 499) // 500ms，OLED 刷新标志
    {
     CountTimer500ms = 0;
     OLEDFlash_flag = 1;
     // 恒流充电即将结束时，逐渐减小充电电流
     if (state == state_I_charge)
     {
      if (ADC_ICHARGE > 200 && ADC_VBAT > 8220)
       pwm_CHARGE -= 0.1;
     }
    }

    if (CountTimer30s > 29999) // 30s，  检测 solar 电压
    {
     CountTimer30s = 0;
     flag_test = 1;
    }

    TIM_ClearITPendingBit(TIM2, TIM_IT_Update);
   }
  }
```

• 太阳能电池电压显示

我们通过间接测量太阳能电池板电压的方法来监测外界光照情况。一般来说，光照强时，太阳能电池板开路电压大；光照弱时，太阳能电池板开路电压小。由于需要监测开路电压，又不能影响系统正常工作，我们设置每 30s 将太阳能电池板短暂开路，使用 ADC 监测电压。

```
    if (flag_test)
    {
     flag_test = 0;

     PWM_CHARGE_SetCompare1(100); // 将太阳能电池板开路
     Delay_ms(50);
     // 太阳能电池板电压采样
     ADC_VSL = (double)AD_GetValue(ADC_Channel_2) * 3300 / 4095 * 147 / 47;

```

```
        VSL[0] = ((ADC_VSL / 1000) % 10) + 0x30;
        VSL[1] = '.';
        VSL[2] = ((ADC_VSL / 100) % 10) + 0x30;
        VSL[3] = ((ADC_VSL / 10) % 10) + 0x30;
        VSL[4] = ((ADC_VSL) % 10) + 0x30;
      }
```

根据硬件电路的设计，太阳能板实际电压与采样电压呈比例关系，因此附带了 `147 / 47` 的系数。

- **ADC 采样**

在系统中，我们需要实时监测太阳能电池板、蓄电池的电压以及充电、放电电流的大小。实时性由定时器的定时中断实现，而采样则由硬件电路与单片机板载 ADC 实现。我们使用多通道 ADC 分别对四个值进行采样，并将采样值显示在 OLED 屏上。

```
        //------------------------------------------------------------AD 转换处理
        if (ADC_flag)
        {
         ADC_flag = 0; // 启动ADC 转换

         // VBAT 采样
         VBAT_sum += (double)AD_GetValue(ADC_Channel_1) * 3300 / 4095; // 取一秒钟内平均值
         VBAT_num++;                // 保存采样次数

         // 电池电压校准
         ADC_VBAT = (double)VBAT_sum / VBAT_num * 8.2 / 8.12 * 147 / 47;
         if (ADC_VBAT > 7700)
          ADC_VBAT = (double)ADC_VBAT * 1.0169 - 67.588;
         else if (ADC_VBAT < 6800)
          ADC_VBAT = (double)ADC_VBAT * 1.0111 - 29.543;
         else
          ADC_VBAT += 55;
         if (state == state_V_charge || state == state_I_charge)
          ADC_VBAT -= 30;

         if (VBAT_num == 20)
         {
          VBAT_sum = 0;
```

```
24.         VBAT_num = 0;
25.       }
26.       // OLED 显示电池电压
27.       VBAT[0] = ((ADC_VBAT / 1000) % 10) + 0x30;
28.       VBAT[1] = '.';
29.       VBAT[2] = ((ADC_VBAT / 100) % 10) + 0x30;
30.       VBAT[3] = ((ADC_VBAT / 10) % 10) + 0x30;
31.       VBAT[4] = ((ADC_VBAT) % 10) + 0x30;
32.
33.       // ILED 采样
34.       ILED_sum += (double)AD_GetValue(ADC_Channel_3) * 3300 / 4095;
35.       ILED_num++;
36.       ADC_ILED = ((((double)ILED_sum / ILED_num / 5.6) + 63) * 1.15954 - 33.815)
    > 0 ? ((((double)ILED_sum / ILED_num / 5.6) + 63) * 1.15954 - 33.815) : 0; //
    线性校准
37.       if (ILED_num == 20)
38.       {
39.         ILED_sum = 0;
40.         ILED_num = 0;
41.       }
42.       ILED[0] = ((ADC_ILED / 1000) % 10) + 0x30;
43.       ILED[1] = ((ADC_ILED / 100) % 10) + 0x30;
44.       ILED[2] = ((ADC_ILED / 10) % 10) + 0x30;
45.       ILED[3] = ((ADC_ILED) % 10) + 0x30;
46.
47.       // ICHARGE 采样
48.       ICHARGE_sum += (double)AD_GetValue(ADC_Channel_4) * 3300 / 4095;
49.       ICHARGE_num++;
50.       ADC_ICHARGE = (((double)ICHARGE_sum / ICHARGE_num / 12) * 0.9281 + 23.96)
    * 1.1362 - 2.6482; // 线性校准
51.       if (ICHARGE_num == 20)
52.       {
53.         ICHARGE_sum = 0;
54.         ICHARGE_num = 0;
55.       }
56.       ICHARGE[0] = ((ADC_ICHARGE / 1000) % 10) + 0x30;
57.       ICHARGE[1] = ((ADC_ICHARGE / 100) % 10) + 0x30;
```

```
58.         ICHARGE[2] = ((ADC_ICHARGE / 10) % 10) + 0x30;
59.         ICHARGE[3] = ((ADC_ICHARGE) % 10) + 0x30;
60.         // 实现 300mA 恒流控制
61.         if (state == state_use_charge)
62.         {
63.          if (ADC_ILED > chargeI + 15)
64.           pwm_LED += 0.3;
65.          else if (ADC_ILED > chargeI)
66.           pwm_LED += 0.03;
67.          else if (ADC_ILED > chargeI - 15)
68.           pwm_LED -= 0.03;
69.          else
70.           pwm_LED -= 0.3;
71.         }
72.       }
```

为了保证采样的精度，减少随机误差的干扰，我们对 1s 之内的采样值取平均处理。对充放电的限流控制与充电恒压的控制在与 ADC 采样同时进行，保证了系统的及时反馈。由于硬件电路的设计，ADC 采样电压与实际电压呈线性关系，需要进行线性校准。

• 蓝牙控制模块

单片机通过 UART1 串口接收蓝牙信息，从而控制整个系统的工作状态。

```
1.       // ---------------------------------------- 蓝牙接收
2.       if (Serial_GetRxFlag() == 1)
3.       {
4.        RxByte = Serial_GetRxData(); // 获取串口接收的数据
5.        sys_flag = RxByte - '0';
6.        LED_State ^= 1;
7.       }
```

• OLED 显示处理模块

根据 OLEDFlash_flag 和 OLEDflag 标志，决定是否在 OLED 上显示信息、是否刷新显示。定时器会定时置位两个标志。

```
1.       //------------------------------------------------------------OLED 显示模块处理
2.       if (OLEDFlash_flag && !OLEDflag)
3.       {
4.        OLEDFlash_flag = 0;
```

```
        OLED_ShowNum(1, 6, ADC_VBAT, 4); // 显示通道1的转换结果ADC_VBAT
        OLED_ShowNum(2, 5, ADC_VSL, 4);  // 显示通道2的转换结果ADC_VSL
        OLED_ShowString(4, 1, MODE[state]);
       }
       if (OLEDFlash_flag && OLEDflag)
       {
        OLEDFlash_flag = 0;
        if (state == state_use_charge)
         OLED_ShowNum(2, 1, ADC_ILED, 3);
        else
         OLED_ShowNum(2, 1, 0, 3);
        OLED_ShowNum(4, 1, ADC_ICHARGE, 3);
       }
```

• 按键处理：

当 flag_KS 标志在定时器的中断函数中被设置时，进入扫描按键状态。根据按键输入，执行相应的操作，按钮 S1 能够在一定范围内增加亮度、S2 减小亮度、S3 可以手动切换模式、S4 可以切换显示内容。

```
       //-------------------------------------------------------按键处理
       if (flag_KS)
       {
        KeyV = Key_GetNum();
        switch (KeyV)
        {
        case S1_PRES: // S1 增加亮度
         if (state == state_use_charge)
          chargeI = (chargeI > 300) ? 325 : chargeI + 25;
         break;

        case S2_PRES: // S2 降低亮度
         if (state == state_use_charge)
          chargeI = (chargeI < 25) ? 0 : chargeI - 25;
         break;

        case S3_PRES: // S3 手动切换模式
         state++;
         state %= 4;
```

```
20.         break;
21.
22.       case S4_PRES: // S4 切换显示内容
23.         OLEDflag++;
24.         OLEDflag %= 2;
25.         OLED_Clear();
26.         if (!OLEDflag) // 显示电压
27.         {
28.           OLED_ShowString(1, 1, "VBAT:xxxxmV");
29.           OLED_ShowString(2, 1, "VSL:xxxxmV");
30.           OLED_ShowString(3, 1, "MODE=");
31.         }
32.         else // 显示电流
33.         {
34.           OLED_ShowString(1, 1, "ILED:");
35.           OLED_ShowString(2, 1, "xxxmA");
36.           OLED_ShowString(3, 1, "ICHARGE:");
37.           OLED_ShowString(4, 1, "xxxmA");
38.         }
39.         break;
40.       default:
41.         break;
42.       }
43.     }
```

5 测试方案与测试结果

5.1 调试步骤：

- 电源调试
- 静态工作点调试
- 动态调试
- 软硬件联调

5.2 硬件调试具体方案及其结果

5.2.1 电源调试：

- ±5V 电压供电测试

SOLAR+与 SOLAR-接电源发生器 12V（CC 恒流模式），Battery 接电源发生器 7.2V 直流，使用万用表直接测量引脚输出电压。

在稳压芯片与电压转换芯片的作用下，供电电压稳定在 4.93V 左右。

5.2.2 静态工作点调试：

- 运放偏置电压测试

测试时，将采样电阻 R5 和 R13 的引脚保留，用来进行运放输入接地，输出测得的电压值即为偏置电压

- 运放增益测试

测量采样电阻两端电压，将其值与对应的单片机 ADC 采样电压值进行比较，其放大倍数即为运放的增益。

经测量，对于充电部分，运放增益为 118 倍左右；对于供电部分，运放增益为 56 倍左右。

- 三极管与 MOS 管工作情况测试

三极管驱动电压配置测试：

排针 PWM_CHARGE 与 PWM_LED 分别接 GND，万用表测量 PWM_CHARGE_OUT 与 PWM_LED_OUT 的输出电压值如下：

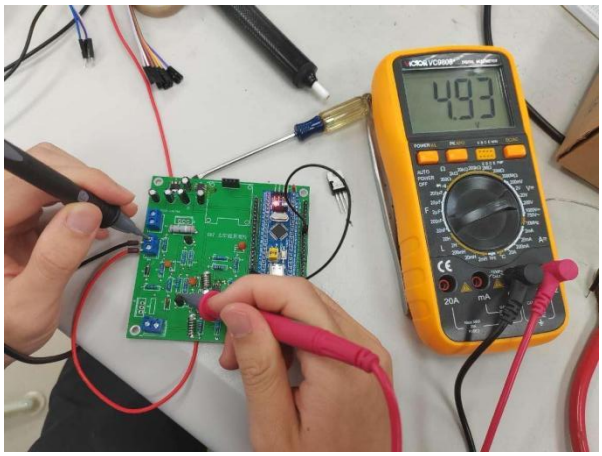

图 19 三极管驱动电压转换测试

MOS 管工作情况测试：

充电模式下，SOLAR+与 SOLAR-正常接太阳能电池板，在 PWM_CHARGE_OUT 高电平的情况下，用万用表测量 SOLAR+与 SOLAR-之间的电流，即可判断 Q1 是否导通；供电模式下，PWM_LED_OUT 输出高电平的情况下，使用外接限流电阻与 LED 灯，3.3V 驱动下，LED 灯灭表示 Q2 工作正常。

5.2.3 动态调试

• 滤波特性测试

对于小信号输出模块，通过配置低通滤波器，我们可以确保 ADC_ICHARGE 与 ADC_ILED 信号输入给单片机时，维持在一个几乎稳定的值附近

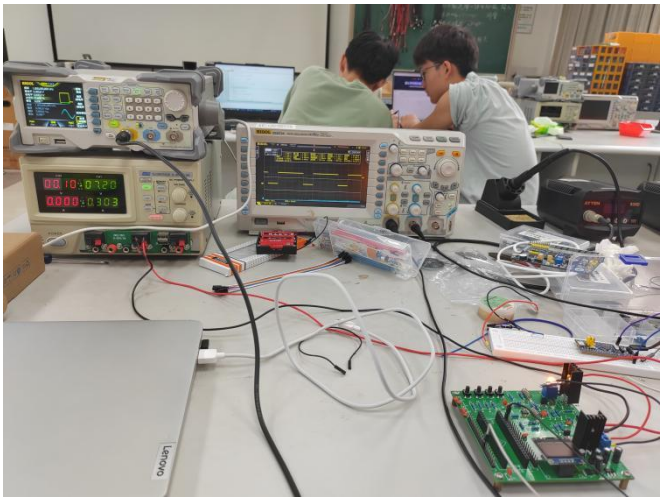

图 20 PWM 波特性测试

调试时，使用信号发生器输入 1000Hz 的方波信号，观察供电电源测量的电流值。调整不同的占空比，我们可以测得流过 LED 的电流，为后续软件控制景观灯亮度提供数据基础。

5.3 软硬件联合调试

5.3.1 PWM 波生成测试

下发代码，检验 PB4 或 PB5 引脚，可以输出正常的 PWM 方波波形

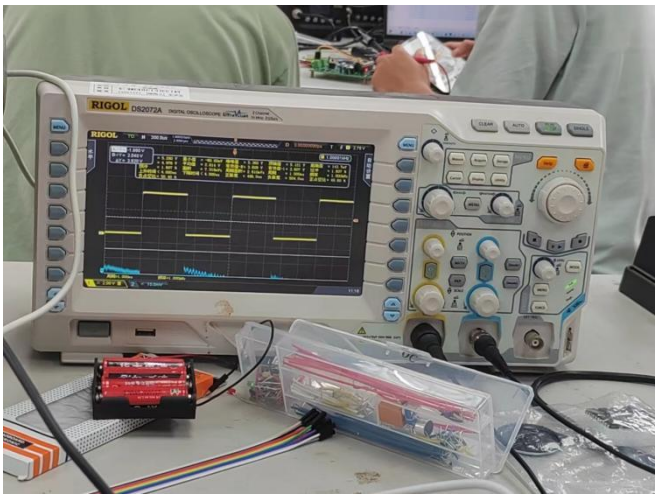

图 21 PWM 波产生

5.3.2 测试基本要求 2：LED 用电控制测试

• 测试目的：确保 LED 在电池电压低于 6V 时自动断电，而在电压高于 6.4V 时恢复供电。

• 调试方案：

使用 QJ3003SIII 直流稳压电源模拟电池电压。当电压设定为 6.5V 时，OLED 屏幕应正确显示电源电压和 useLED 状态，并保持 LED 点亮。随着电压降至 5.9V，LED 应自动熄灭，OLED 屏幕显示 stop 工作状态。当电压回升至 6.5V 时，LED 应重新点亮，OLED 屏幕恢复显示 useLED 状态。

5.3.3 测试基本要求 3：LED 恒流驱动测试

• 测试目的：实现 LED 的恒流驱动，确保电流维持在 300mA，控制精度不超过 15mA。

• 调试方案：

在本实验电路中，PWM_LED 为 0 意味着 LED 为电流最大状态，为了避免 LED 损坏，首先给 LED 设置 PWM 下限值为 70。然后，通过调整至 useLED 模式，监测 OLED 显示屏上的 ILED 电流值，并与万用表测量的限流电阻 R12 两端电压所换算的电流值进行比较，调整程序参数，确保电流控制的精确性。

5.3.4 测试发挥部分 2：恒压限流充电测试

• 测试目的：在电池电压达到 8.2V 时，通过 PWM 控制实现恒压限流充电，并在电流降至 100mA 以下时自动停止充电。

• 调试方案：

首先校准 Icharge 数值，确保其能准确显示充电电流。然后，切换至 Icharge 恒压限流充电模式，观察 OLED 屏上 ILED 电流值的变化，验证系统是否能在电流低于 100mA 时自动切换至停止模式。

5.3.5 测试发挥部分 3：LED 亮度调节测试

• 测试目的：通过设定实现 LED 亮度的精细调节，调节步进不大于 50mA。

• 调试方案：

通过按压按钮 SW1 和 SW2，监测 OLED 屏上 ILED 电流值的变化，确保其符合设定的步进值 25mA，并观察 LED 灯的亮度变化是否符合预期。

5.3.6 测试发挥部分 4：景观灯光控功能测试

• 测试目的：实现景观灯的光控功能，使其在白天熄灭，夜间点亮。

• 调试方案：

使用 QJ3003SIII 直流稳压电源模拟太阳能电池板。当电压值低于 6V 时，系统应自动切换至 useLED 或 stop 模式，LED 从熄灭变为点亮。当电压值高于 6V 时，系统应切换至 Vcharge、Icharge 或 stop 模式，LED 从点亮变为熄灭。

5.3.7 测试发挥部分 5：其他功能测试

• 测试目的：实现其他功能，如时空控制等。

• 调试方案：

使用 QJ3003SIII 直流稳压电源模拟太阳能电池板，将电压值调整为 1V 以模拟黑夜模式。通过手机 APP 发送熄灯指令，观察 LED 是否能从点亮状态变为熄灭。

5.4 数据处理及数据分析

5.4.1 OLED 显示屏数据校准

由于小信号部分运放放大系数的误差，最终显示在 OLED 显示屏上的数据需要在理论计算结果的基础上，用最小二乘法对 Icharge 和 ILED 的显示值进行了线性回归处理，使其能够以更小的误差来显示充电电流和用电电流的数值，进行软件层面的拟合校准，具体校准结果如下：

• 充电电流校准

Icharge实际值	188	179	169	160	151	142	134	125
Icharge显示值	168	160	151	143	135	127	120	113

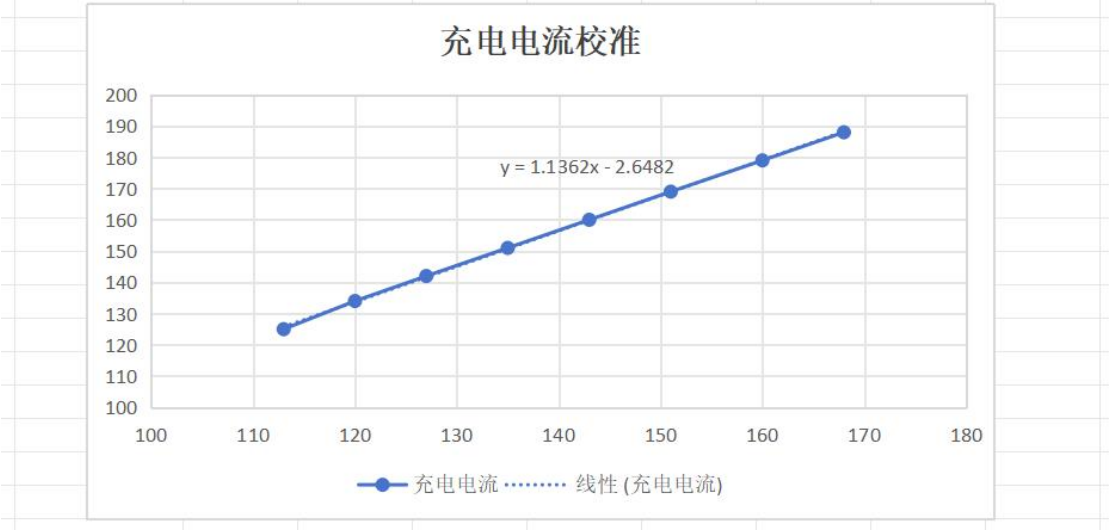

图 22 充电电流校准

• 供电电流校准

ILED实际值	290	269	243	220	198	175	148	121	93
ILED显示值	300	275	250	225	200	175	150	125	100

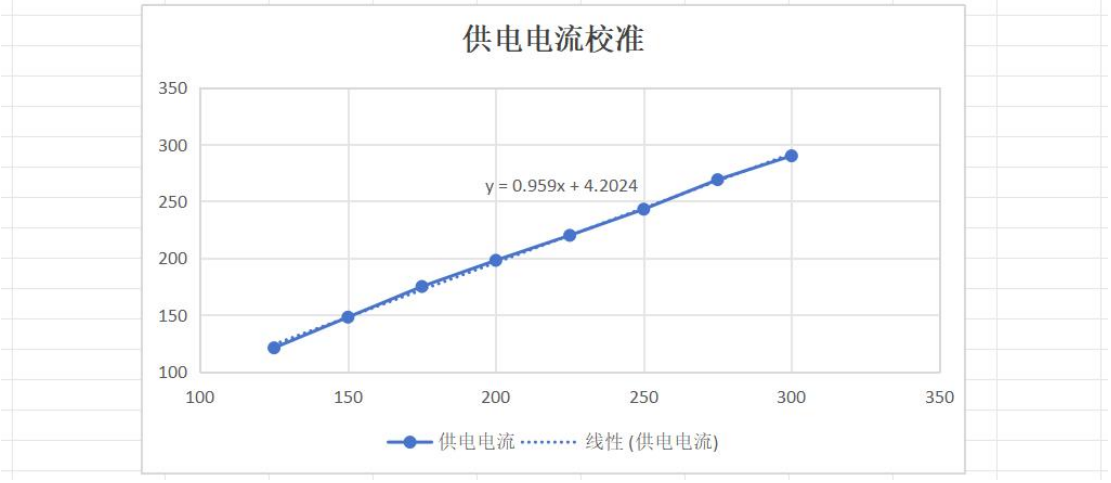

图 23 供电电流校准

5.4.2 数据误差分析

• 恒流供电电流误差：

300mA 要求下，OLED 显示屏电流工作范围再 296—303mA 左右波动，误差为：

$$\mathrm{m} = \frac{4}{300} \times 100\% = 1.33\%$$

• 电流 OLED 显示值与实际间接测量值误差：

当 OLED 显示屏为 300mA 时，3.4Ω限流电阻两端电压为 1.013V，误差计算如下：

$$I_m = \frac{1.013}{3.4} = 298mA$$

$$\mathrm{m} = \frac{|\mathrm{I} - \mathrm{I_m}|}{\mathrm{I}} \times 100\% = 0.67\%$$

6 小组分工及成绩分配

• **小组成员成绩分配方式：**成绩**平均**分配

• **小组成员具体分工：**

姓名	工作内容
	负责编程作业完成、软件程序设计、软硬件联调、报告撰写
	负责原理图设计、焊接、硬件调试、软件程序设计、报告撰写
	负责原理图设计、PCB 绘制、硬件调试、软硬件联调、报告撰写

7 总结与收获