



西安电子科技大学
XIDIAN UNIVERSITY

基于 Arduino 的全天候 太阳能自动跟踪控制器

姓名	学号	分工
杨博宇	15020110050	程序编写
杨魁	15020110054	结构设计
赵子铭	15020110056	论文撰写
秦韵韵	15020110057	电路搭建
刘康磊	15020110058	组装测试

2017.10.25

目录

摘要	1
1. 项目背景分析	2
2. 理论基础分析	4
2.1 电路基本原理	4
2.2 太阳跟踪方式的基本原理	4
2.3 跟踪器感光结构的基本原理	6
2.4 跟踪器传动结构的基本原理	6
2.4.1 跟踪器基本传动结构	6
2.4.2 舵机的基本工作原理	7
3. 系统的总体设计	9
3.1 系统的设计目标	9
3.2 设计的主要内容	9
3.3 分模块设计方案	10
3.3.1 光电检测模块	10
3.3.2 系统检测模块	10
3.3.2.1 Arduino UNO 简介	10
3.3.2.2 系统检测电路设计	11
3.3.3 伺服传动模块	14
3.3.4 电源电压转换模块	14
4. 测试结果与分析	16
附录一：参考文献	17
附录二：Arduino 程序源代码	18
附录三：元件清单	28
附录四：开发及测试环境	29

基于 Arduino 的全天候太阳能自动跟踪控制器

摘要：太阳能作为一种新型的清洁能源，受到了世界各国的广泛重视。现阶段影响太阳能普及的主要原因是太阳能电池的成本较高而光电转化效率却较低。因此，如何提高太阳能利用效率是太阳能行业发展的关键问题。在国内，大多数太阳能电池阵列都是固定安装的，无法保证太阳光实时垂直照射，导致太阳能资源不能得到充分利用。太阳能自动跟踪控制器在跟踪太阳旋转的情况下可接收到更多的太阳辐射能量，从而提高太阳能电池板的输出功率。

在本文的设计中，用 Arduino 做控制核心，采用光敏电阻和电位器构成光电检测电路再连接到两个电压比较器上，根据光照情况的不同，输出不同的控制信号，从而控制伺服传动机构在水平与竖直方向转过不同的角度，在太阳的不停移动过程中，伺服电机转-停交替，从而使太阳能接收装置始终面朝太阳。

关键词：太阳能；Arduino；自动跟踪；伺服电机

1. 项目背景分析

能源问题一直是人们关注的热点问题，太阳能作为一种极为丰富并且可再生的清洁能源更是受到人们的重视。因而，太阳能光伏发电是最具应用前景的太阳能利用方式。目前，太阳能光伏发电的成本太高，世界各国正在积极的改进电池制造工艺，采用新技术以提高转换效率，降低光伏发电的成本。全自动跟踪式太阳能发电设备从控制技术出发，采用新的光伏发电装置技术，在相同的条件下，要比固定朝向式太阳能发电设备的发电量提高 35%，而成本下降 25%[1]。现阶段我国能源短缺，环境保护与能源利用也是目前我们所面临的重大课题。地球每天接收到的太阳能约 4×10^{15} 千瓦时，相当于 2.5×10^8 万桶石油，并且太阳能清洁无污染、取之无尽、用之不竭。与此同时，我国每年接收的太阳能约 $335 \sim 873 \text{ kJ/cm}^2$ 。因此，更好的利用太阳能，对于我国的环境保护以及资源利用都具有重要的意义。

太阳能利用率不高的原因主要在于：分散性、方向性及不稳定性。为了提高太阳能的利用率，设计太阳能跟随系统，使太阳光永远垂直照射在接收面上并且提高太阳能的吸收率和转化率。

现有的太阳能自动跟踪控制器无外乎两种：一是使用一只光敏传感器与施密特触发器或单稳态触发器，构成光控施密特触发器或光控单稳态触发器来控制电机的停、转；二是使用两只光敏传感器与两只比较器分别构成两个光控比较器控制电机的正反转。由于一年四季、早晚和中午环境光和阳光的强弱变化范围都很大，所以上述两种控制器很难使太阳能接收装置四季全天候跟踪太阳。

本文所介绍的控制电路也包括两个电压比较器，但设在其输入端的光敏传感器则由四只光敏电阻并联组合而成。每一组两只光敏电阻中的一只为比较器的上偏置电阻，另一只为下偏置电阻；一只检测太阳光照，另一只则检测环境光照，送至比较器输入端的比较电平始终为两者光照之差。所以，本控制器能使太阳能接收装置四季全天候跟踪太阳，而且调试十分简单，成本

也比较低。[2]特别适合天气变化比较复杂和无人值守的情况，有效地提高了太阳能利用率，有较好的推广应用价值。

2. 理论基础分析

2.1 电路基本原理

电路基本原理图如图 2.1-1 所示。记从左到右的四个光敏电阻分别为 R1、R2、R3、R4，四个电阻分别与一个电位器串联后又并联在一起，每两个为一组，即 R1 与 R2、R3 与 R4 一组（包括电位器），每组分别接向一个电压比较器，对每组光敏电阻与电位器中间电压进行比较之后，输出相应的电信号到后续模块进行比较处理。

其中 R1 与 R2 比较输出的信号控制伺服电机（舵机）在 yaw 轴的转动，R3 与 R4 比较输出的信号控制伺服电机在 pitch 轴的转动。在感受到阳光照射后，四个光敏电阻的阻值即发生相应的变化，在经过电压比较器的比较后输出的高低电平即反映光照差值，从而后续的 Arduino 模块可以对其作出判断，从而输出控制信号。

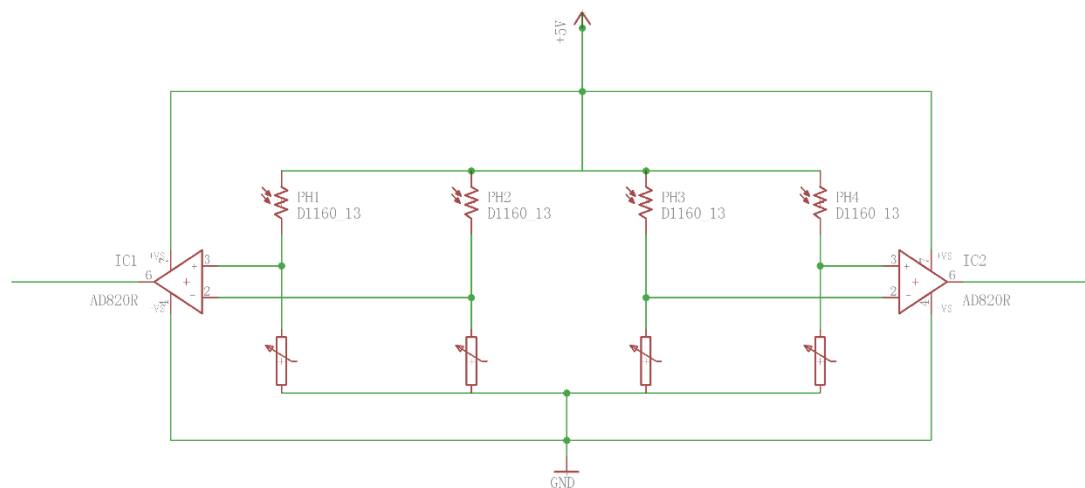


图 2.1-1 电路基本原理图

2.2 太阳跟踪方式的基本原理

以集热器跟踪太阳的方法为例：

高度角-方位角式太阳跟踪。高度角-方位角式太阳跟踪方法又称为地平

坐标系双轴跟踪，其原理如图 2.1-1 所示。集热器的方位轴垂直于地面，另一根轴与方位轴垂直，成为俯仰轴。工作时集热器根据太阳的视日运动绕方位轴转动改变方位角，绕俯仰轴作俯仰运动改变集热器的倾斜角，从而使反射镜面的主光轴始终与太阳光线平行。这跟踪系统的特点是跟踪精度高，而且集热器装置的重量保持在垂直轴所在的平面内，支承结构的设计比较容易。集热器采光面上直接日射入射角 θ 由太阳赤纬角 δ 、太阳时角 ω 、集热器倾角 β 、集热器方位角 γ 和实验地的纬度 ϕ 计算，计算公式如下：

$$\cos \theta = (\sin \theta \sin \phi \cos \beta) - (\sin \delta \cos \phi \sin \beta \cos \gamma) + (\cos \delta \cos \phi \cos \beta \cos \omega) + (\cos \delta \sin \phi \sin \beta \cos \gamma \cos \omega) + (\cos \delta \sin \beta \sin \gamma \sin \omega) \quad (2.2-1)$$

式(2.2-1)中一年内第 n 天的太阳赤纬角计算为：

$$\delta = 23.45 \sin \frac{360 (284+n)}{365} \quad (2.2-2)$$

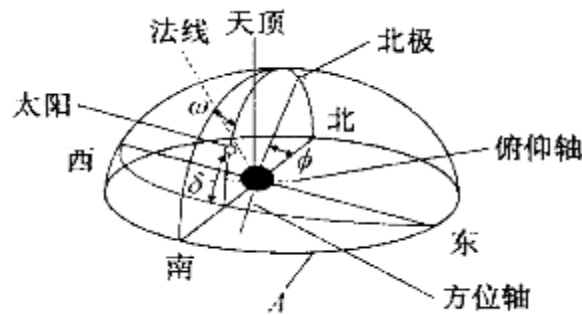


图 2.2-1 高度角-方位角式全跟踪

从式(2.2-1)、(2.2-2)可以看出，当 δ 、 ω 、 ϕ 确定以后，集热器倾角 β 和方位角 γ 的值决定了直接日射入射角 θ 。因此只要控制集热器的角度使其具有合适的倾角和方位角，就可以保证太阳光线入射角 θ 为 0，从而最大限度地收集太阳能。[3]

2.3 跟踪器感光结构的基本原理

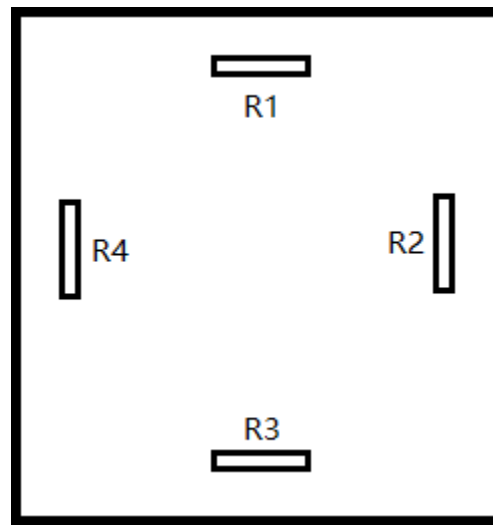


图 2.3-1 感光结构示意图

如图 2.3-1 所示。用一个方形平板，在其表面上、下、左、右四个位置布置 4 个相同的光敏电阻。其中 R1、R3 对称安装在图示平板的上下两侧，用来粗略感知太阳的视高度即高度角；R2、R4 对称安装在图示平板的左右两侧，用来粗略感知太阳由东往西运动的偏转角度即方位角。

在感受到太阳光照射时，由于每个光敏电阻接收到太阳光线角度存在差异，每个电阻的阻值会相应产生不同的变化，从而引起负载电压的变化，在 R1 与 R3、R2 与 R4 之间会产生电压差。利用电压比较器对每组光敏电阻的电压进行比较，从而粗略确定感光装置在水平与竖直方向移动的角度，从而实现跟踪器对太阳的跟踪，使太阳光线始终垂直照射板面。

2.4 跟踪器传动结构的基本原理

2.4.1 跟踪器基本传动结构

跟踪器的基本传动结构采用两个伺服电机相连接的方式，来控制感光装置在水平和竖直方向的移动。其基本结构如图 2.4-1 所示。

其中，舵机 A 控制感光装置在 pitch 轴即竖直方向的转动，舵机 B 控制感光装置在 yaw 轴即水平方向的转动。在输出控制信号的控制下，舵机 A 与 B 分别转过一定角度，使太阳光线垂直照在感光装置上。

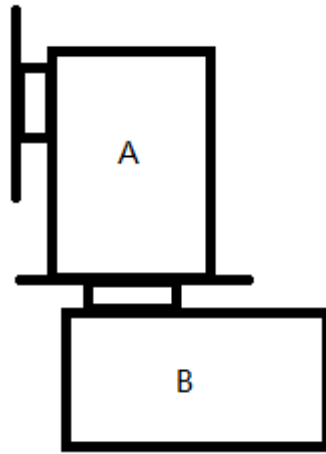


图 2.4-1 基本传动结构示意图

2.4.2 舵机的基本工作原理

控制电路板接受来自信号线的控制信号，控制电机转动，电机带动一系列齿轮组，减速后传动至输出舵盘。舵机的输出轴和位置反馈电位计是相连的，舵盘转动的同时，带动位置反馈电位计，电位计将输出一个电压信号到控制电路板，进行反馈，然后控制电路板根据所在位置决定电机转动的方向和速度，从而达到目标停止。其工作流程如下：

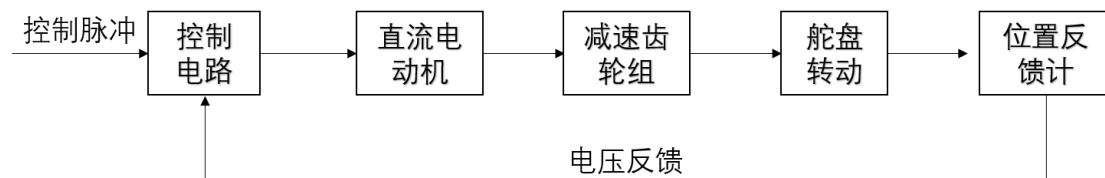


图 2.4-2 舵机工作流程图

舵机的控制信号周期为 20ms 的脉宽调制 (PWM) 信号, 其中脉冲宽度从 0.5-2.5ms, 相对应的舵盘位置为 0—180 度, 呈线性变化。也就是说, 给它提供一定的脉宽, 它的输出轴就会保持一定对应角度上, 无论外界转矩怎么改变, 直到给它提供一个另外宽度的脉冲信号, 它才会改变输出角度到新的对应位置上如图 2.4-3 所示。舵机内部有一个基准电路, 产生周期为 20ms, 宽度 1.5ms 的基准信号, 有一个比较器, 将外加信号与基准信号相比较, 判断出方向和大小, 从而生产电机的转动信号。由此可见, 舵机是一种位置伺服驱动器, 转动范围不能超过 180 度, 适用于那些需要不断变化并可以保持的驱动器中, 比如说机器人的关节、飞机的舵面等。

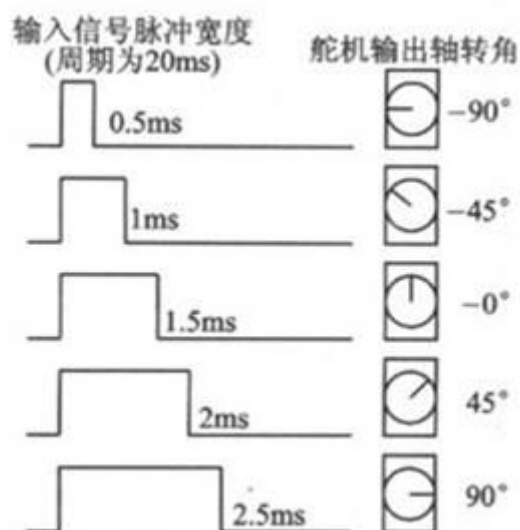


图 2.4-3 舵机输出转角与输入脉冲的关系

3. 系统的总体设计

3.1 系统的设计目标

根据目前太阳能技术发展特点以及国内外光伏发电系统的发展状况，针对现阶段太阳能光伏发电系统普遍存在的效率偏低的现实。本文研究并设计了基于 Arduino 的双轴自动太阳能跟踪控制系统。具体设计思路如下：

(1) 双轴：系统采用两个伺服电机来控制感光装置水平方向和竖直方向的移动；

(2) 自动跟踪：系统的运行过程是自动化的，采用光电跟踪方式实现对太阳能的跟踪；

(3) Arduino：系统以 Arduino 为控制核心输出控制信号，实现对传感及传动装置的控制；

3.2 设计的主要内容

本文所研究的太阳能自动跟踪装置是以 Arduino 为控制核心的自动控制系统，整个装置是通过软件控制系统控制硬件结构部分完成的。本文涉及的主要内容是控制系统，所以在这里主要讲系统的硬件和软件两部分。

(1) 系统的硬件部分主要包括芯片、光敏电阻及各种电子元器件、电路图的连接。电路主要包含四部分：

a. 光电检测电路：这部分内容主要作用是通过光敏电阻产生的电压差产生电信号，来判别太阳的方位；

b. 系统检测电路：Arduino 通过接受光电检测电路发出的电信号来控制伺服电机的转动，来达到跟踪太阳的目的。

c. 伺服传动电路：伺服电机通过接收 Arduino 传来的信号，然后在 yaw 轴与 pitch 轴分别转过一定角度，从而实现对太阳的跟踪。

d. 电源电压转换电路：将 XL6009 升压模块反用，使 12V 电压降为 5V，对电路进行供电。

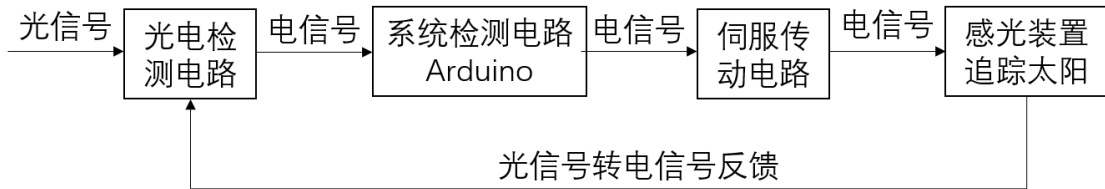


图 3.2-1 系统总体框图

(2) 软件部分：在硬件电路的基础上用 Arduino 单片机语言配写相应的算法控制程序，对软硬件进行联合调试，直到系统能够稳定运行。

3.3 分模块设计方案

3.3.1 光电检测模块

该模块主要由 2.1 中所描述的光电检测的基本电路构成。通过电压比较器，对两组光敏电阻产生的电压分别进行比较，产生相应的电信号传到 Arduino，由其进行识别比较，并输出控制信号。

3.3.2 系统检测模块

主要由 Arduino 构成，开机启动后，系统进行自检，然后通过接受光电检测电路产生的电信号，对电压进行判断然后输出控制信号，控制传动机构在水平及竖直方向进行转动，最终使感光平面与太阳光线垂直。

3.3.2.1 Arduino UNO 简介

Arduino 是源于意大利的一套开源硬件开发平台，Arduino UNO 是其推出的最为基础的开发板。配合外围器件（传感器、控制开关、LED 屏等），任何稍有基础的人都可以基于这个平台快速搭建出具有各种有趣功能的实用电子电路。其基本组成如下：

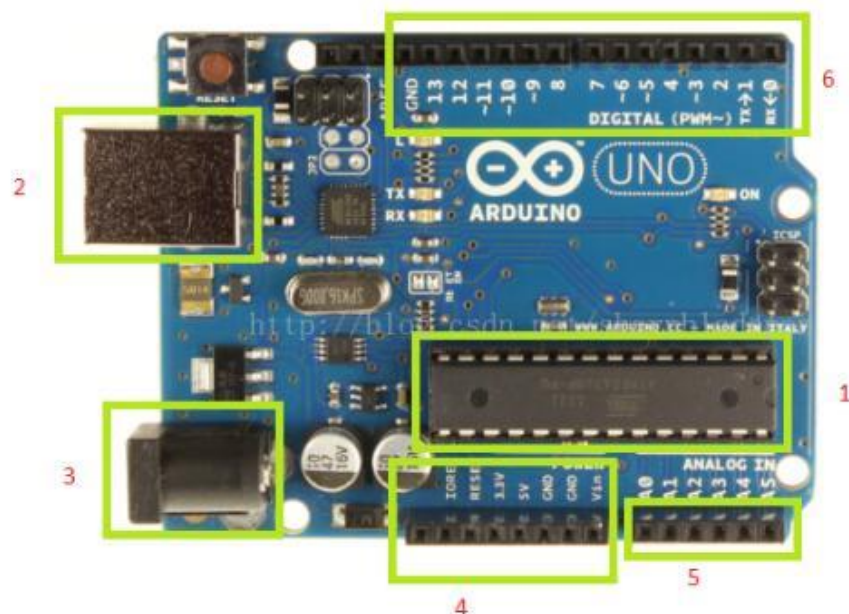


图 3.3-1 Arduino UNO 开发板

- (1) 微处理器：ATmega328, 8 位处理器，片内包含 32KB（0.5KB 由 BootLoader 使用），2KB SRAM, 1KB EEPROM，运行时钟频率为 16MHz。
- (2) USB 接口：与电脑连接，用于从电脑中下载程序，同时给 UNO 单板供电。
- (3) 外接电源：当 UNO 脱离电脑独立运行时，用来给装置供电，一般使用 5V 电源即可。
- (4) 电源管脚：地，可输出 5V、3.3V；也可以通过 Vin 给整个装置供电。
- (5) 模拟输入：6 个模拟输入，提供 10-bit 的解析度（0-1203）。
- (6) 数字输入/输出：14 个数字输入/输出管脚，可以由程序自己定义为输入或者输出；其中 0, 1 两个管脚是串口复用管脚；3, 5, 6, 9, 10, 11 六个管脚可提供 8 位（0-255）的 PWM 输出；13 管脚与板上 LED 相连，当输出高电平时，LED 点亮。

3.3.2.2 系统检测电路设计

系统连接图如图 3.3.2 所示。将光电检测电路的两个输出端分别接到 Arduino 的 A0、A1 端口，然后板子接外接电源构成整个的系统检测电路。当

Arduino 检测到 A0、A1 端口传输来的高低电平信号以后，通过烧录的程序算法，对该信号进行处理，判断太阳位置及角度，并输出相应的控制信号到伺服传动机构。同时该电路在移动过程中也可以接受光电检测电路传输回来的反馈信号，在追踪太阳的过程中输出控制信号，不断地调整伺服机构的移动，使感光板面能够与太阳光线垂直。

3.3.3 伺服传动模块

该模块的主要设计采用两枚 SG90 舵机连接构成,电路外接 Arduino 的端口,接收 Arduino 发出的控制信号。其机械结构如图 2.4-1 所示,电路原理图如图 3.3-3 所示:

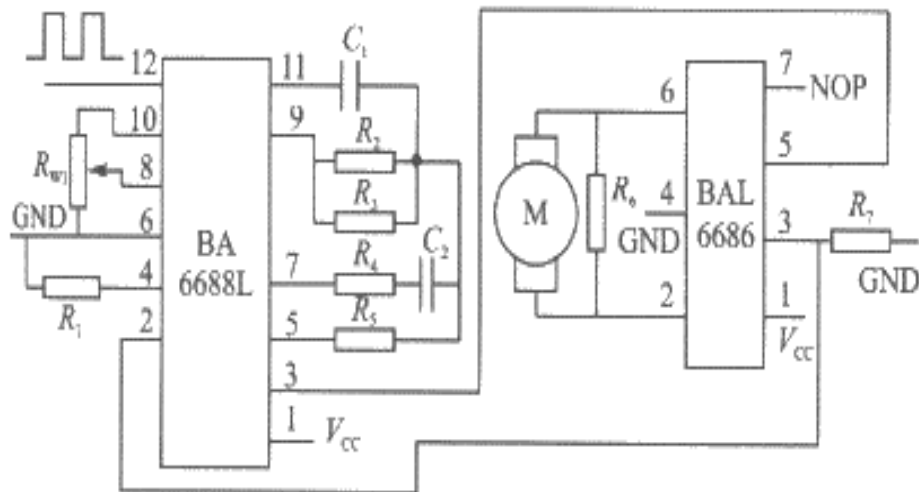


图 3.3-3 舵机电路图

PWM 信号由接收通道进入信号解调电路 BA6688 的 12 脚进行解调,获得一个直流偏置电压,该直流偏置电压与电位器的电压比较,获得电压差由 BA6688 的 3 脚输出,该输出送入电动机驱动集成电路 BAL6686,以驱动电动机正反转,当电动机转动时,通过级联减速齿轮带动电位器 RW1 旋转,直到电压差为 0,电机停止转动。借此,利用两个舵机,连接感光装置,即可接收 Arduino 的控制信号后进行转动,使感光的板面追踪太阳。

3.3.4 电源电压转换模块

因为所选用电源为 12V 直流电源进行供电,而 Arduino 的供电电压为 5V,因此需要设计电源电压转换模块进行电源电压转换,使系统整体正常工作。本设计中采用 XL6009 调压模块进行电压转化。

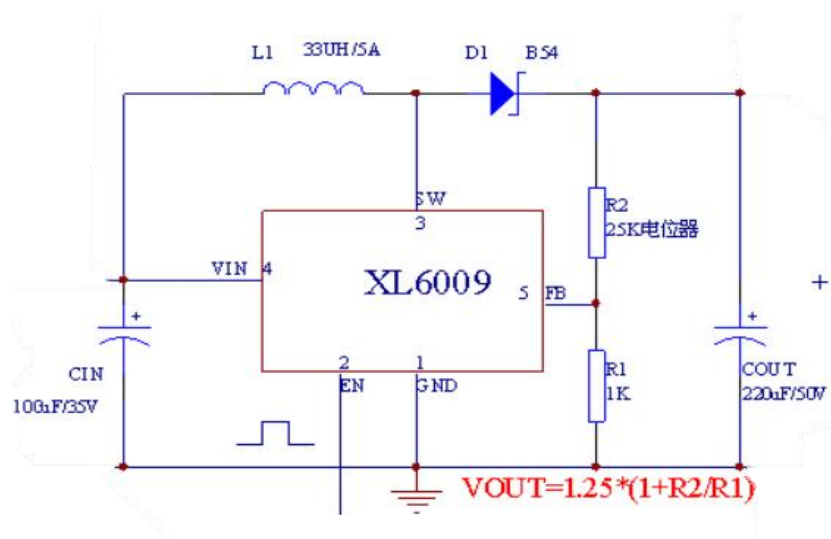


图 3.3-4 XL6009 电路图

XL6009 调节器是一宽输入范围，电流模式，能够产生正或负输出电压的直流/直流转换器。它可以被配置为一升压反激式 SEPIC 或反相转换器。芯片 XL6009 由功率 N 沟道 MOSFET 管和固定频率振荡器构成，电流模式结构在宽范围输入和输出电压下稳定工作。

其电路基本原理图如图 3.3-4 所示，该模块通过外接 12V 直流电压，调节电位器 R2，然后使输出电压达到 5V。

4. 测试结果与分析

实物最终制作完成后，对整个作品进行测试，光敏电阻感受光照后，正常产生电压差；伺服电机可以正常转动追踪太阳。整个作品基本功能已经达成。但在制作过程中也出现了一个比较典型的问题：

伺服电机最开始转动并未按照设置的 180° 最大角度转动，而是 360° 转动。

经问题排查以后发现，该现象是由于供电不足导致 Arduino 输出的方波控制信号严重失真。在提高供电电压后问题得到解决，测试到正常的方波输出信号如下：



图 4.1-1 控制信号输出波形图

附录一：

参考文献：

- [1] 陈维,李戡洪.太阳能利用中的跟踪控制方式的研究 [J].能源工程,2003(3):24-28.
- [2] 胡济元.全天候太阳能自动跟踪控制器 [J].无线电,1997,(3):40.
- [3] 郑晓年,黄巧燕.太阳跟踪方法及应用 [J].能源技术,2003,24(4):150.

附录二：

Arduino 程序源代码：

```
#include<Servo.h>

//定义 pitch 轴舵机和 yaw 轴舵机

Servo pitch_ser;

Servo yaw_ser;

//初始化及自检过程

void setup()

{

    pitch_ser.attach(5);

    yaw_ser.attach(6);

    pinMode(A0, INPUT);

    pinMode(A1, INPUT);

    pinMode(A2, INPUT);

    pinMode(A3, INPUT);

    Serial.begin(9600);

    //伺服电机自检过程

    int pitch_angle;

    int yaw_angle;

    //pitch 轴伺服电机自检

    for (pitch_angle = 90; pitch_angle < 180; pitch_angle++)

    {
```

```
    pitch_ser.write(pitch_angle);

    delay(5);

}

delay(100);

for (pitch_angle = 179; pitch_angle > 0; pitch_angle--)

{

    pitch_ser.write(pitch_angle);

    delay(5);

}

delay(100);

for (pitch_angle = 1; pitch_angle < 91; pitch_angle++)

{

    pitch_ser.write(pitch_angle);

    delay(5);

}

delay(100);

//yaw 轴伺服电机自检

for (yaw_angle = 90; yaw_angle < 180; yaw_angle++)

{

    yaw_ser.write(yaw_angle);

    delay(5);

}
```

```
delay(100);

for (yaw_angle = 179; yaw_angle > 0; yaw_angle--)

{

    yaw_ser.write(yaw_angle);

    delay(5);

}

delay(100);

for (yaw_angle = 1; yaw_angle < 91; yaw_angle++)

{

    yaw_ser.write(yaw_angle);

    delay(5);

}

delay(100);

//双轴联动自检

//正向转动自检

for (pitch_angle = 90, yaw_angle = 90; pitch_angle < 180 && yaw_angle
< 180; pitch_angle++, yaw_angle++)

{

    pitch_ser.write(pitch_angle);

    yaw_ser.write(yaw_angle);

    delay(5);

}
```

```
//逆向转动自检

for (pitch_angle = 179, yaw_angle = 179; pitch_angle > 0  &&
yaw_angle > 0; pitch_angle--, yaw_angle --)

{

    pitch_ser.write(pitch_angle);

    yaw_ser.write(yaw_angle);

    delay(5);

}

for (pitch_angle = 1, yaw_angle = 1; pitch_angle < 91  && yaw_angle
< 91; pitch_angle++, yaw_angle ++)

{

    pitch_ser.write(pitch_angle);

    yaw_ser.write(yaw_angle);

    delay(5);

}

//伺服电机角度归位

pitch_ser.write(90);

yaw_ser.write(90);

}

int sensor_pitch_a, sensor_pitch_b;

int sensor_yaw_a, sensor_yaw_b;
```

```
int pitch_angle = 90;

int yaw_angle = 90;

//校准 pitch 轴和 yaw 轴光敏传感器的初始差值

const static int pitch_set = 0;

const static int yaw_set = 0;

//舵机转动角度控制向量

int pitch_motor_victor;

int yaw_motor_victor;

int step = 2;

//程序标准执行过程

void loop()

{

    //伺服电机角度归位

    pitch_ser.write(pitch_angle);

    yaw_ser.write(yaw_angle);

    sensor_pitch_a = analogRead(A0);

    sensor_pitch_b = analogRead(A1);

    sensor_yaw_a = analogRead(A2);

    sensor_yaw_b = analogRead(A3);

    //数据滤波及标准化

    //均值滤波

    int i, j, k;
```



```
for (i = 0; i < 10; i++)  
{  
    sensor_pitch_a  += analogRead(A0);  
    sensor_pitch_b  += analogRead(A1);  
    sensor_yaw_a    += analogRead(A2);  
    sensor_yaw_b    += analogRead(A3);  
    delay(5);  
}  
  
sensor_pitch_a /= 10;  
sensor_pitch_b /= 10;  
sensor_yaw_a   /= 10;  
sensor_yaw_b   /= 10;  
  
const static int threshold = 30;  
  
const static int step = 1;  
  
const static int speed = 2;  
  
if(sensor_pitch_a - sensor_pitch_b > threshold)  
{  
    if(sensor_yaw_a - sensor_yaw_b > threshold)  
    {  
        pitch_angle += step;  
        yaw_angle -= step;  
    }  
}
```

```
    pitch_ser.write(pitch_angle);

    yaw_ser.write(yaw_angle);

    delay(speed);
}

else if(sensor_yaw_a - sensor_yaw_b < 0 - threshold)
{
    pitch_angle += step;

    yaw_angle += step;

    pitch_ser.write(pitch_angle);

    yaw_ser.write(yaw_angle);

    delay(speed);
}

else
{
    pitch_angle += step;

    pitch_ser.write(pitch_angle);

    delay(speed);
}
}

else if(sensor_pitch_a - sensor_pitch_b < 0 - threshold)
{
    if(sensor_yaw_a - sensor_yaw_b > threshold)
```

```
{

    pitch_angle -= step;

    yaw_angle += step;
    pitch_ser.write(pitch_angle);

    yaw_ser.write(yaw_angle);

    delay(speed);

}

else if(sensor_yaw_a - sensor_yaw_b < 0 - threshold)

{

    pitch_angle -= step;

    yaw_angle -= step;

    pitch_ser.write(pitch_angle);

    yaw_ser.write(yaw_angle);

    delay(speed);

}

else

{

    pitch_angle -= step;

    pitch_ser.write(pitch_angle);

    delay(speed);

}

}
```

```
else
{
    if(sensor_yaw_a - sensor_yaw_b > threshold)
    {
        yaw_angle += step;
        yaw_ser.write(yaw_angle);
        delay(speed);
    }
    else if(sensor_yaw_a - sensor_yaw_b < 0 - threshold)
    {
        yaw_angle -= step;
        yaw_ser.write(yaw_angle);
        delay(speed);
    }
}

//舵机转动限位
if(pitch_angle > 175)
    pitch_angle = 175;
else if(pitch_angle < 5)
    pitch_angle = 5;
if(yaw_angle > 175)
```

```
    yaw_angle = 175;

else if(yaw_angle < 5)

    yaw_angle = 5;

Serial.print("sensor_pitch_a:");

Serial.println(sensor_pitch_a);

Serial.print("sensor_pitch_b:");

Serial.println(sensor_pitch_b);

Serial.print("sensor_yaw_a:");

Serial.println(sensor_yaw_a);

Serial.print("sensor_yaw_b:");

Serial.println(sensor_yaw_b);

Serial.println("  ");

Serial.println("  ");

}
```

附录三：

元件清单：

Arduino UNO Rev3	1 片
3296W 电位器	4 枚
5506 光敏电阻	4 枚
XL6009 调压模块	1 片
SG90 舵机	2 枚

附录四：

开发及测试环境

软件设计环境

电路设计软件：Autodesk Eagle v2016

单片机烧录软件：Arduino Developers Kits v1.7.8

电路仿真软件：NI Multisim v14.0

硬件测试环境

示波器：Rigol DS1102E