SI 100B

Sun-tracking system

Name	陈博克 Chen Boke
Student No.	2020533035
Email	chenbk@shanghaitech.edu.cn
Name	吴致豪 Wu Zhihao
Student No.	2020531051
Email	wuzhh@shanghaitech.edu.cn
Name	何欣灿 He Xincan
Student No.	2020531017
Email	hexc@shanghaitech.edu.cn
Date	January 5, 2021

上海科技大学
ShanghaiTech University

Abstract

At present, there are many methods to establish solar light tracking system. Some use GPS to collect information, use some algorithms to track the position of the sun, some people need a slightly complex structure to achieve tracking. This paper presents a novel, low-cost, simple, easy to install and implement solar self-tracking system. The algorithm only uses the light intensity data collected locally to track the direction of the sun, which is simple, convenient, and adaptable. The experimental results show that the system is simple, reliable, and practical.

Key words: sun-tracking Arduino solar power photosensitive resistance

1. Problem Statement

1.1 Requirements

Basic1: machine part
Basic2: circuit building
Basic3: sun-tracking
Bonus1: MPPT—find the best output
Bonus2: dark light reset & extreme reset
Bonus3: home-made DC converter
Bonus4: 3D printing transfer structure

2. Introduction

At the beginning, we were provided with an Arduino uno, some wires and photosensitive resistance which are used to connect the circuit, the main body of the bracket, and the core motive source: steering engine. We have considered loads of possible outcome of the sun-tracking system but compromised on one design, the one fits the components we had in hand. So, the specific mind set of this sun-tracking project is to use the input signal of Arduino to transfer the light signal to the electric signal. Because we cannot do the real time monitoring of the strength of light, so we use the voltage to replace it, which is also much easier to measure. And with the real time statistics, we can control the motion of the steering engine with our program.

3. Details

Coding

This device uses photosensitive resistance, dark resistance is almost 1m Ω, strong light condition is only a few hundred Ω, so we series 1K adjustable resistance for current limiting and voltage dividing, otherwise the photosensitive resistance is too low, the approximate conduction will burn things.

Ad data acquisition Arduino 5V corresponding to the integer value of 0-1023, about equal to 0.049v per unit. Through the test, under normal sunlight conditions, the adjustable resistance is 500 Ω, and the collected data is between 800-900. According to the different voltage of each photoresistor, we can judge the strength of the light source distributed around.

Originally, I wanted to directly use the photoresistor to judge, but the flashlight test was ok, but in the sun, because the sun is the same size, it can't work normally, so I added a light barrier, only the light is vertical can receive light at the same time, when the light is not vertical, it will produce shadows, causing some photoresistors can't receive light. The board used is Arduino nano and the supporting expansion board. Because the steering gear is used, it is convenient to use the expansion board.

So, the whole process can be concluded as:

Start——> reset to the origin place——> monitor the sunlight(voltage) ——> adjust the steering gear angle ——> over ——> Restart

Here is the program implementation:

Before we start, we should invoke the library:

```
#include <Servo.h>
Servo lr_servo;//定义左右旋转舵机名称
Servo ud_servo;//定义上下旋转舵机名称

int lr_angle = 90;//设置初始角度为90度
int ud_angle = 0;//设置初始角度为0度,让太阳能板水平朝上，检测太阳光最强点
int l_state = A0;//下
int r_state = A1;//上
int u_state = A2;//左
int d_state = A3;//右

int lr_servopin = 9;//定义舵机控制信号引脚
int ud_servopin = 10;
```

Then we can do the setup work:

```
void setup() {
    Serial.begin(9600); //定义串口波特率
    lr_servo.attach(lr_servopin);  // 设置舵机控制针脚
    ud_servo.attach(ud_servopin);  // 设置舵机控制针脚
    pinMode(l_state,INPUT);//设置引脚类型
    pinMode(r_state,INPUT);
    pinMode(u_state,INPUT);
    pinMode(d_state,INPUT);

    lr_servo.write(lr_angle);//恢复初始角度
    ud_servo.write(ud_angle);
    delay(1000);

}
```

The following code is important because it involve the correct direction of left and right when ud_angle < or > 180.

```
int L,R,U,D;
void loop() {
  if (lr_angle<=90){
     L = analogRead(l_state);//读取传感器模拟电压值,0-1200
     R = analogRead(r_state);
     U = analogRead(u_state);
     D = analogRead(d_state);
  }
  else {
     L = analogRead(l_state);
     R = analogRead(r_state);
     U = analogRead(d_state);
     D = analogRead(u_state);
  }
  int error = 10;//定义误差范围,防止抖动
  int m_speed = 20;//设置舵机速度
```

Then we can write the turn right/left part:

```
  if(abs(L-R)>error && L>R){ //判断误差是否在允许范围内，否者调整舵机
        lr_angle-=5;//调小角度
        if (lr_angle<0){  //极限复位
//          lr_angle = 0;
        ud_angle = 180-ud_angle;
      lr_angle = lr_angle+180;
        }

        lr_servo.write(lr_angle);  //输出舵机角度
        delay(m_speed);

  }
  else if(abs(L-R)>error && L<R){//判断误差是否在允许范围内，否者调整舵机
      lr_angle+=5;//调大角度
      if (lr_angle>180){   //极限复位
//        lr_angle = 180;
        ud_angle = 180-ud_angle;
        lr_angle = lr_angle-180;
        }
        lr_servo.write(lr_angle);  //输出舵机角度
        delay(m_speed);
```

and up/down part:

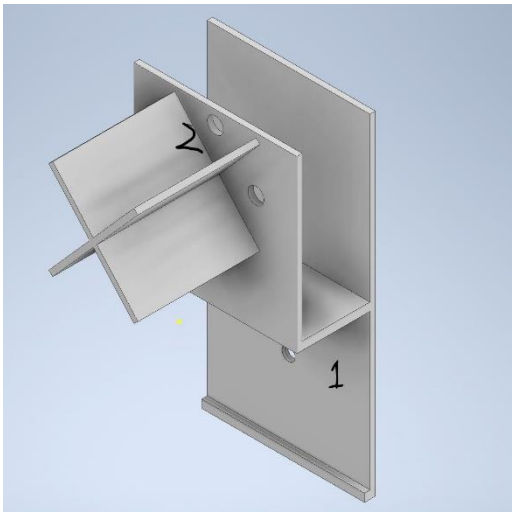
```
/**********************上下调节程序**********************/
if(abs(U-D)>error && U>=D){//判断误差是否在允许范围内，否者调整舵机
      ud_angle-=5;//调小角度
      if (ud_angle<0){  //限制舵机转动角度
      ud_angle=0;
      }
      ud_servo.write(ud_angle);  //输出舵机角度
      delay(m_speed);

}
   else if(abs(U-D)>error && U<D){//判断误差是否在允许范围内，否者调整舵机
    ud_angle+=5;//调大角度
    if (ud_angle>180){  //限制舵机转动角度
      ud_angle=180;
      }

      ud_servo.write(ud_angle);  //输出舵机角度
      delay(m_speed);
```

In the end, in order to monitor the statistics, we implement a monitor of the serial port, thus we can better adjust the error:

```
/**************************串口监视器**************************/
    //串口显示光敏电阻阻值和舵机角度
    Serial.print(" L ");
    Serial.print(L);
    Serial.print(" R ");
    Serial.print(R);
    Serial.print(" U ");
    Serial.print(U);
    Serial.print(" D ");
    Serial.print(D);
    Serial.print(" ud_angle ");
    Serial.print(ud_angle);
    Serial.print(" lr_angle ");
    Serial.println(lr_angle);
    delay(0);//在测试时,串口数据接收过快,可以适当加延时调整
}
```

3D printing

According to our design, a structure to fasten the solar panel and shadow the photoresistances is required. So we designed the structure below and made it by 3D



Our design

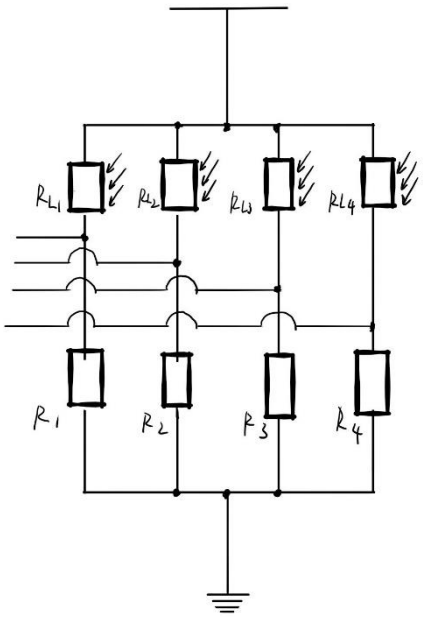
printing.

Part 1, which is marked in the picture, is used to fasten the solar panel. The structure prevents the solar panel from rotation. And the solar panel can be screwed on the motor through the screw hole in part 1.

The ‘x’ structure in part 2 ensures that the photoresistances, which are placed on the four directions through the four holes, only receive the light from their own direction.

Circruit

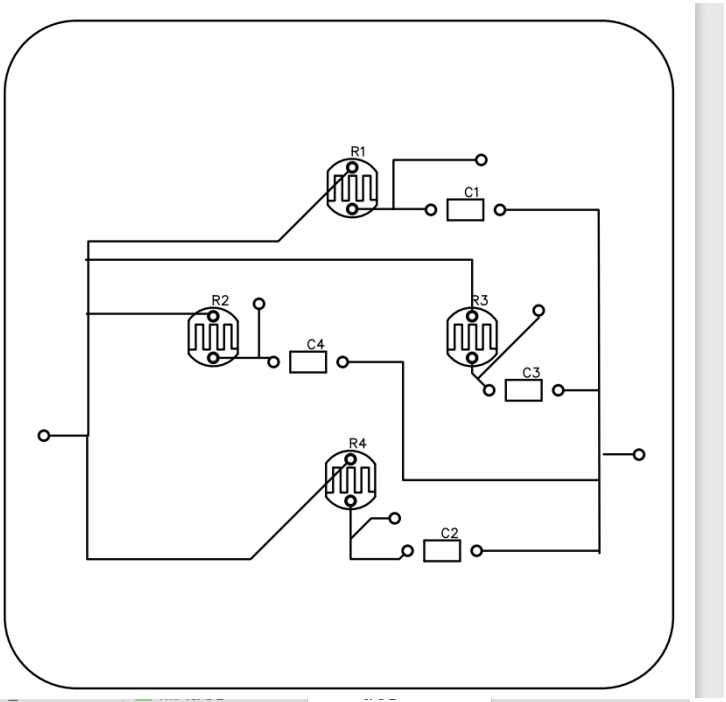
The circuit is designed to compare the light intensity of the four photo resistances. Using the property of photoresistance. We designed the circuit below.

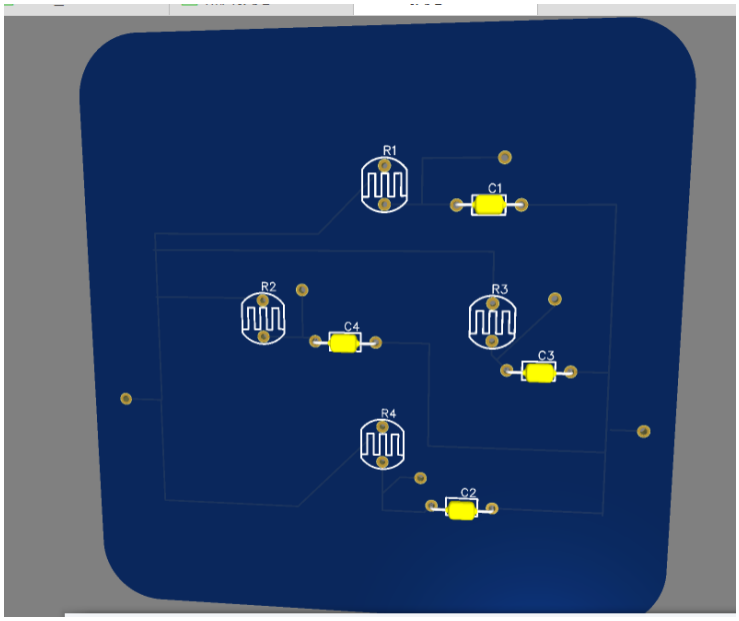


By comparing the output voltages form the opposite resistances, we can compare their light intensity.（if the voltage is higher, the light is stronger）

PCB

Following the circuit,using tools to create the map of PCB to replace Bread plate to make the circuit more direct and beautiful. And also fits the 3D-print part.





4. Conclusions

In the end, we have an understanding on Single chip microcomputer, circuit to create a common system to achieve sun catching function or other basic function. Besides have a Preliminary understanding on 3D-print and how to make PCB to replace Bread plate to make the circuit more direct and beautiful.In the process, we pull together to de-bug, we know a group how to solve a problem and how to correct the mistakes.

Appendix

```
#include <Servo.h>
Servo lr_servo;//定义左右旋转舵机名称
Servo ud_servo;//定义上下旋转舵机名称

int lr_angle = 90;//设置初始角度为 90 度
int ud_angle = 0;//设置初始角度为 0 度,让太阳能板水平朝上，检测太阳光最强点
int l_state = A0;//下
int r_state = A1;//上
int u_state = A2;//左
int d_state = A3;//右

int lr_servopin = 9;//定义舵机控制信号引脚
int ud_servopin = 10;

void setup() {
  Serial.begin(9600); //定义串口波特率
  lr_servo.attach(lr_servopin);  // 设置舵机控制针脚
  ud_servo.attach(ud_servopin);  // 设置舵机控制针脚
  pinMode(l_state,INPUT);//设置引脚类型
  pinMode(r_state,INPUT);
  pinMode(u_state,INPUT);
  pinMode(d_state,INPUT);

  lr_servo.write(lr_angle);//恢复初始角度
  ud_servo.write(ud_angle);
  delay(1000);
```

```
}
int L,R,U,D;
void loop() {
 if (lr_angle<=90){
   L = analogRead(l_state);//读取传感器模拟电压值,0-1200
   R = analogRead(r_state);
   U = analogRead(u_state);
   D = analogRead(d_state);
 }
 else {
   L = analogRead(l_state);
   R = analogRead(r_state);
   U = analogRead(d_state);
   D = analogRead(u_state);
 }
 int error = 10;//定义误差范围,防止抖动
 int m_speed = 20;//设置舵机速度
 /**********************左右调节程序**********************/
 if(abs(L-R)>error && L>R){ //判断误差是否在允许范围内，否者调整舵机
     lr_angle-=5;//调小角度
     if (lr_angle<0){  //极限复位
//       lr_angle = 0;
     ud_angle = 180-ud_angle;
    lr_angle = lr_angle+180;
     }

     lr_servo.write(lr_angle);  //输出舵机角度
     delay(m_speed);

 }
 else if(abs(L-R)>error && L<R){//判断误差是否在允许范围内，否者调整舵机
    lr_angle+=5;//调大角度
    if (lr_angle>180){   //极限复位
//      lr_angle = 180;
     ud_angle = 180-ud_angle;
     lr_angle = lr_angle-180;
     }
     lr_servo.write(lr_angle);  //输出舵机角度
     delay(m_speed);
```

```
    }
 else if(abs(L-R)<=error){ //判断误差是否在允许范围内,在范围内,舵机静止,角度不做调整
    lr_servo.write(lr_angle); //输出舵机角度
  }
 /**********************上下调节程序**********************/
 if(abs(U-D)>error && U>=D){//判断误差是否在允许范围内，否者调整舵机
    ud_angle-=5;//调小角度
    if (ud_angle<0){  //限制舵机转动角度
    ud_angle=0;
    }
    ud_servo.write(ud_angle);  //输出舵机角度
    delay(m_speed);

 }
   else if(abs(U-D)>error && U<D){//判断误差是否在允许范围内，否者调整舵机
    ud_angle+=5;//调大角度
    if (ud_angle>180){  //限制舵机转动角度
     ud_angle=180;
     }

    ud_servo.write(ud_angle);  //输出舵机角度
    delay(m_speed);

    }
  else if(abs(U-D)<=error){  //判断误差是否在允许范围内,在范围内,舵机静止,角度不做调整
   lr_servo.write(lr_angle);  //输出舵机角度
  }
/**********************暗光复位**********************/
 if(L>900 && R>900 && U>900 && D>900){
  lr_angle = 90;
  ud_angle = 0;
  lr_servo.write(lr_angle);
  ud_servo.write(ud_angle);
  delay(20);
 }
```

```
/**********************串口监视器**********************/
  //串口显示光敏电阻阻值和舵机角度
  Serial.print(" L ");
  Serial.print(L);
  Serial.print(" R ");
  Serial.print(R);
  Serial.print(" U ");
  Serial.print(U);
  Serial.print(" D ");
  Serial.print(D);
  Serial.print(" ud_angle ");
  Serial.print(ud_angle);
  Serial.print(" lr_angle ");
  Serial.println(lr_angle);
  delay(0);//在测试时,串口数据接收过快,可以适当加延时调整
}
```