```
#include <Servo.h>
Servo HORIZONTALservo, VERTICALservo;
int Hangle=90;
int Vangle=30;
int avrHOR, cal0,cal1,cal2;
int H1photocellPin = 1, H2photocellPin = 2, VphotocellPin = 0;
int photocellReading1,photocellReading2, photocellReading0;
long x,y;
int LEDPin = 19;
float PERcal0,PERcal1,PERcal2;

void setup()
{
Serial.begin(9600);
pinMode(LEDPin,OUTPUT);
HORIZONTALservo.attach(10);//the servo that turns the solar cells from 0 to 180 degrees(east to west!)
HORIZONTALservo.write(Hangle);
VERTICALservo.attach(9);   //the servo that gives the cells the right inclination (say 20 to 80 degrees)
VERTICALservo.write(Vangle);
initialization();                    // the routine that calibrates the photocells
}

void loop()
{
find_best_HORangle();  // the routine that turns the solar cells on an horizontal plane so as to find the
                       //vertical plane that includes the light source (say the sun!)
find_best_VERangle();  // the routine that turns the solar cells on that vertical plane, so as to be
                       //perpendicular to light beams
delay(100);
}

void find_best_HORangle()
{
 do
 {
   photocellReading1 = analogRead(H1photocellPin);
   photocellReading2 = analogRead(H2photocellPin);
   photocellReading1=photocellReading1+photocellReading1*PERcal1;//the PERcal1 is a correction
                                                                 //factor calculated in "initialization"
   photocellReading2=photocellReading2+photocellReading2*PERcal2;
   if ((photocellReading1<0)||(photocellReading2<0))  //the negative numbers shouldn't be there, if
                                                      //there are, something went very wrong!!
            {Serial.println("out of HOR range  ");initialization();break;}
   x=(photocellReading1-photocellReading2);
   y=0.1*photocellReading1;  //I allow a 10% divergence,  maybe is more than it should be!

   if ((photocellReading1) < (photocellReading2))
        {if((Hangle < 180) && (Hangle > 0))
          {Hangle = Hangle + 2;}      //if right is lighter than left then go to that direction with steps of 2degs
        HORIZONTALservo.write(Hangle);}
   if ((photocellReading1) > photocellReading2) //...and if the opposite is the case, then do the opposite!
```

```
        {if((Hangle < 180) && (Hangle > 0))
          {Hangle = Hangle - 2;}
        HORIZONTALservo.write(Hangle);}
   Serial.print("HOR photocellReading1= ");
   Serial.println(photocellReading1);
   Serial.print("HOR photocellReading2= ");
   Serial.println(photocellReading2);
   Serial.print("Hangle= ");
   Serial.println(Hangle);
   Serial.print("x= ");
   Serial.println(x);
   Serial.print("y= ");
   Serial.println(y);
   if ((Hangle>158) || (Hangle<12)) {break;}     //overturning is not allowed!
   delay(500);
 }
    while( abs(x)>y);       //if the right to left luminosity difference is less than 10% of the left reading, I'm
                            //satisfied, so "stop"
    blinkk(50);
}


void find_best_VERangle()
{
   do
 {
  photocellReading1 = analogRead(H1photocellPin);  //after finding the best horizontal orientation, let's
                                                   //find the best vertical!
  photocellReading2 = analogRead(H2photocellPin);
  photocellReading0 = analogRead(VphotocellPin);
  photocellReading1=photocellReading1+photocellReading1*PERcal1;
  photocellReading2=photocellReading2+photocellReading2*PERcal2;
  photocellReading0=photocellReading0+photocellReading0*PERcal0;
  if ((photocellReading1<0)||(photocellReading2<0)||(photocellReading0<0))
            {Serial.println("out of V range  ");initialization();break;}
  avrHOR = (photocellReading1 + photocellReading2)/2;
  if (photocellReading0 > avrHOR)             //the idea is to compare the upper photocell reading with the
                                              //average of the two lower ones
    {if (Vangle < 80)  {Vangle = Vangle + 2;} }     //since the initial position is the lower possible, it turns up
                                                    //with steps of 2 degs
          else
              {if (Vangle > 30) {Vangle = Vangle - 2;}} //you can increase inclination only if it is less than
                                                        //maximum
  VERTICALservo.write(Vangle);
  Serial.print("V photocellReading1= ");
   Serial.println(photocellReading1);
   Serial.print("V photocellReading2= ");
   Serial.println(photocellReading2);
   Serial.print("photocellReading0= ");
   Serial.println(photocellReading0);
   Serial.print("avrHOR= ");
```

```
   Serial.println(avrHOR);
   Serial.print("Vangle= ");
   Serial.println(Vangle);
   if ((Vangle>88) || (Hangle<22)) {break;}
   delay(500);
 }
  while(abs(photocellReading0-avrHOR)>0.1*photocellReading0);
  blinkk(100);
}

void blinkk(int j)
{
 for(int k=0;k<10;k++)
{digitalWrite(LEDPin,HIGH);
delay(j);
digitalWrite(LEDPin,LOW);
delay(j);
}
}

void initialization()
{
    long MIDcal;

    Serial.println("++++++++++++++++++++++++++++++++++++");
    blinkk(200);                          // get ready for calibration!!
    digitalWrite(LEDPin,HIGH);
    delay(5000);                          // light the down left photocell with a light source
    cal0= analogRead(VphotocellPin);      // takes the "value"
    digitalWrite(LEDPin,LOW);
    Serial.print("cal0 init= ");
    Serial.println(cal0);
    delay(500);
    //*********************************
    digitalWrite(LEDPin,HIGH);            //get ready for down right photocell!!
    delay(5000);                          // SAME light source
    cal1=analogRead(H1photocellPin);      // takes "value"
    digitalWrite(LEDPin,LOW);
    Serial.print("cal1 inti= ");
    Serial.println(cal1);
    delay(500);
    //*********************************
    digitalWrite(LEDPin,HIGH);            //get ready for the upper photocell1
    delay(5000);
    cal2=analogRead(H2photocellPin);      // takes "value"
    digitalWrite(LEDPin,LOW);
    Serial.print("cal2 inti= ");
    Serial.println(cal2);
    delay(500);
    blinkk(200);
    MIDcal = (cal0+cal1+cal2)/3;          //average of all photocells
```

```
    Serial.print("MIDcal init=  ");
    Serial.println(MIDcal);
    PERcal0=float((cal0-MIDcal)/MIDcal);    //divergence of each measurament from the average
    PERcal1=float((cal1-MIDcal)/MIDcal);   //the decimal number would make a correction to every new
                                           //measurement in every possible environment
    PERcal2=float((cal2-MIDcal)/MIDcal);
    Serial.print("PERcal0=  ");
    Serial.print(PERcal0,3);
    Serial.print("    PERcal1=  ");             //all the printing helps me checking the row of values
    Serial.print(PERcal1,3);
    Serial.print("    PERcal2=  ");
    Serial.print(PERcal2,3);
    Serial.println("$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$");
}
```