

The Solar Tracker project will use two light sensitive resistors to track the general intensity of light and the balance of light between the two detectors. The circuit used is shown below.

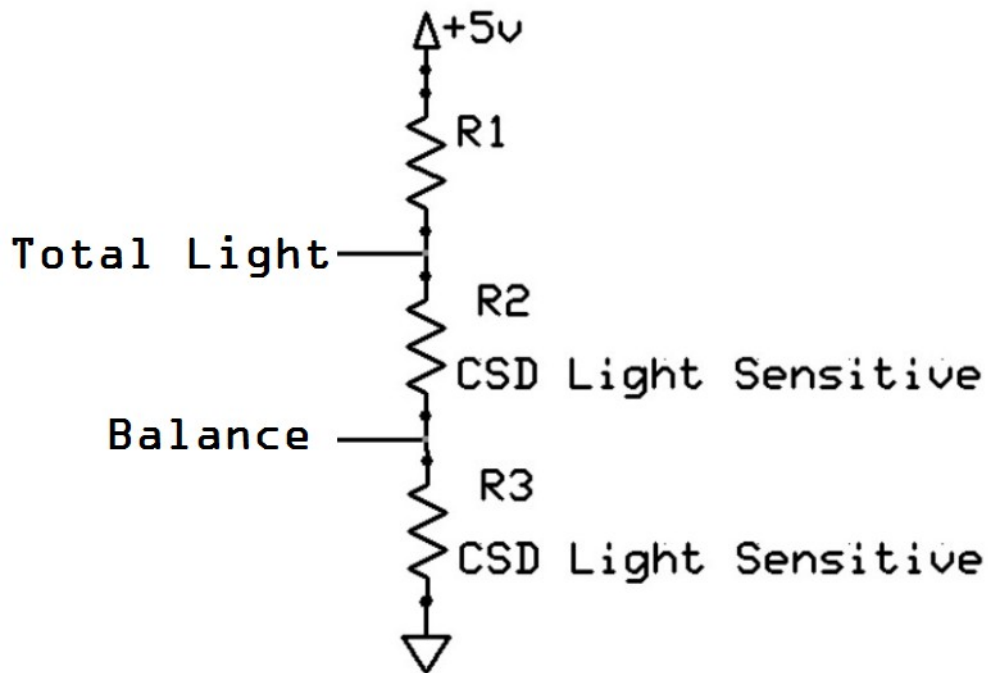


Figure 1. Light Detection Circuit, Yielding Total Light and Balance.

It should be noted that the components R2 and R3 are made of a material that is light sensitive, and whose resistance to electricity is reduced in the presence of light. From an analysis standpoint this is a voltage divider, and as light hits R2 and R3 (their resistance will drop) the voltage at "Total Light" will drop. This voltage will be used to determine if sufficient light is present.

Now the two light sensitive components will be pointed in different directions in the application. Thus if the light is directed at one sensor more than the other, that component's resistance will be less. This effect will cause the "Balance" voltage to be either above or below half of the "Total Light" voltage. This will be demonstrated in class.

Appendix A: Code

```
#include <LiquidCrystal.h>
#include <Servo.h>

// LCD Display
LiquidCrystal lcd(11, 9, 5, 6, 7, 8);

// Software Timer for control 2 Hz
unsigned long ControlTimer;
#define CONTROL_INTERVAL 500

// put your setup code here, to run once:
void setup()
{
    // display setup
    lcd.begin(16, 2);
    lcd.clear();

    // Software timer setup.
    ControlTimer = millis();

    // Set ADC for EXTERNAL reference voltage
    analogReference( EXTERNAL );
} // end of setup.

// Parameters of search algorithm
#define LightOff 500
#define LightTolerance 25

void loop()
{
    int Total, MidPoint;
    // put your main code here, to run repeatedly:
    if (millis() - ControlTimer >= CONTROL_INTERVAL)
    {
        // Read in light levels.
        Total = analogRead(0);
        MidPoint = analogRead(1);

        // Send results to display.
        lcd.clear();
        lcd.print("Tot. ");
        lcd.print(Total);
        lcd.print("->");
        lcd.print(3.3*(float)Total / 1024.0, 3);
        lcd.setCursor(0, 1);
        lcd.print("Bal. ");
        lcd.print(MidPoint);
        lcd.print("->");
        lcd.print(3.3*(float)MidPoint / 1024.0, 3);

        // Update timer.
        ControlTimer += CONTROL_INTERVAL;
    } // end of timer if
} // end of loop.
```

