Project 13
Brightness Led Control

Objective

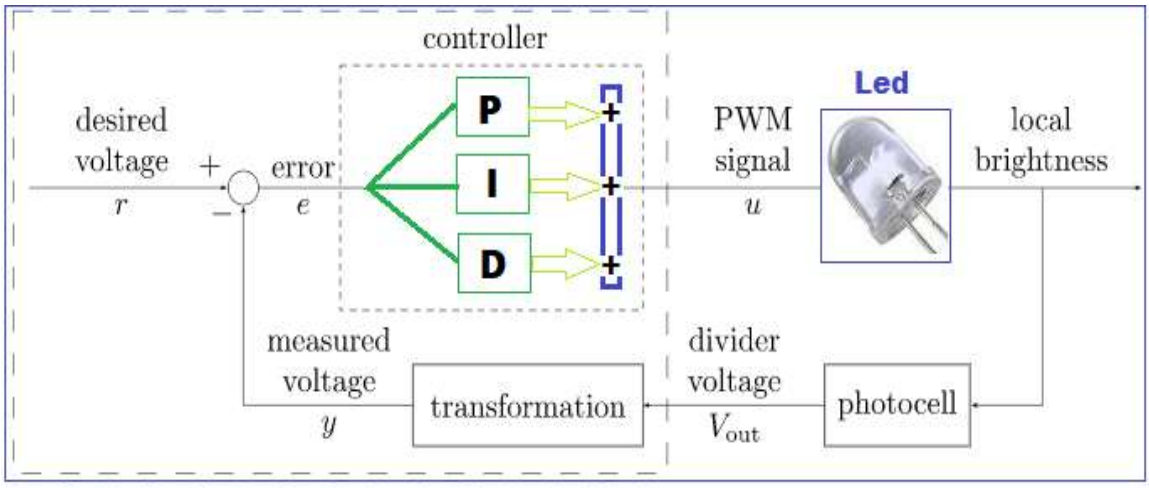
Control of Brightness Led using PID and Arduino UNO.

Required Equipment's

Arduino Uno, LED, Photocell (LDR), jumper Cables, Resistors (1kΩ, and 220Ω).

Control System

Now we'll put the LED and photocell together in order to obtain a desired brightness level. Here is a block diagram of the control system we will implement to achieve



this:

In this controller project, we will use voltage as a representation of brightness. Because of the voltage divider configuration, the voltage read by the Arduino's input pin will vary proportionally to the brightness sensed by the photocell.

The measured voltage is compared to a voltage representing the desired brightness, resulting in some error. This error is then fed into a controller, which transforms the error into a PWM signal to change the LED brightness. For example, if the measured brightness is lower than desired, the error will be positive, and the controller coefficients will produce a positive PWM signal to drive the LED to become brighter. This has the effect of increasing the measured voltage, hence decreasing the error. This kind of controller configuration is called a regulator, and its job is to achieve and maintain zero error between the measured output and the desired output.

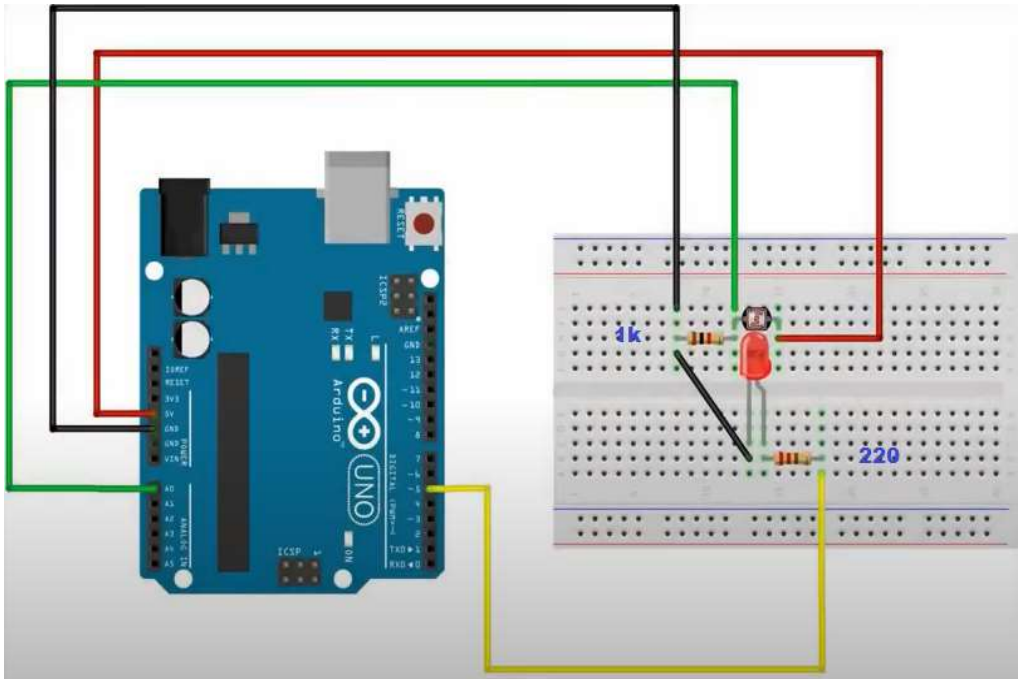
Circuit Construction

Now let's put together the hardware of our control system. The photocell will be the sensor and the led will be the actuator that we control. Our goal is to control the local brightness of small volume surrounding our sensor.

The circuit consists of a light emitting diode (LED) circuit, driven by one of the Arduino's digital I/O pins capable of producing a pulse width modulation (PWM) signal. This will allow the LED's brightness to change. A photocell facing the LED senses the ambient lighting. The objective of the circuit is to demonstrate an automatic feedback control system that drives the LED to a desired brightness level near the sensor. You will be able to cast shadows on the photocell and watch as the LED brightness increases to compensate for the dimmed lighting.

Both components (the LED and the photocell) should be connected from the previous two sections. The most important part of the control circuit construction (aside from making the correct electrical connections) is that the LED and photocell are close to and facing one another. This will ensure that the LED is able to influence the reading of the sensor as much as possible.

Once constructed, the circuit should look like the image below:



Arduino code

```
#include <PID_v1.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x27, 16, 2);
double Setpoint;
double  photocellReading;
double  ledBrightness;

double kp=0.1, ki=10, kd=0.0012;

#define photocellPin  A0
#define ledPin 5

PID myPID(&photocellReading, &ledBrightness, &Setpoint, kp, ki, kd, DIRECT);

void setup() {
Serial.begin(9600);
lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("ldr_lcd");
  lcd.setCursor(0,1);
  lcd.print("salam 3alaykom!");

Setpoint=180;
myPID.SetMode(AUTOMATIC);
myPID.SetTunings(kp, ki, kd);
delay(2000);
}

void loop() {
photocellReading = analogRead(photocellPin);
myPID.Compute() ;
analogWrite(ledPin, ledBrightness);
Serial.print( Setpoint);
Serial.print(" ");
Serial.print(photocellReading);
Serial.print(" ");
Serial.println(Setpoint-photocellReading);
  lcd.setCursor(0,0);
  lcd.print("Setpoint= ");
  lcd.print(Setpoint);
  lcd.setCursor(0,1);
  lcd.print("Real LDR= ");
  lcd.print(photocellReading);
}
```