

CS 362: Homework 4

Due on March 8, 2024 at 11:59pm

Professor Troy 11:00am

Ryan Magdaleno

rmagd2@uic.edu

1 Report Information

Title: Lab Report for Lab 4: Photoresistor (LDR – Light Dependent Resistor)

Name/NetID: Ryan Magdaleno, rmagd2

Lab Description: The lab report outlines the creation of a circuit and program utilizing a photoresistor and a 16x2 display. The display indicates the relative amount of light in the room using five predefined text values: dark, partially dark, medium, fully lit, and brightly lit. These values are shown on the top line of the LCD display. The bottom line displays the number of milliseconds since the Arduino was last reset, continuously updating without using the delay() function. Determine the 5 light range values via testing.

2 Hardware Required

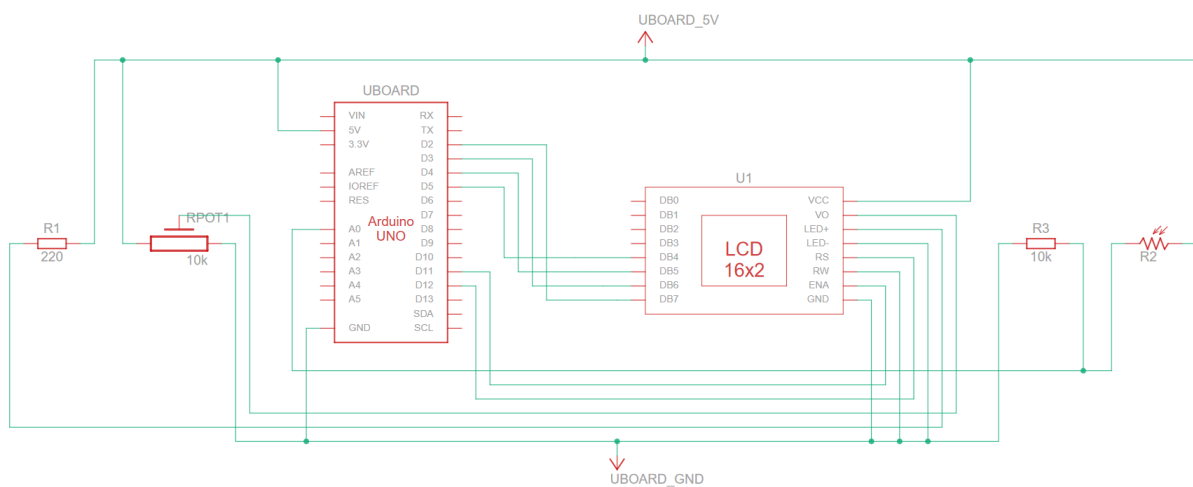
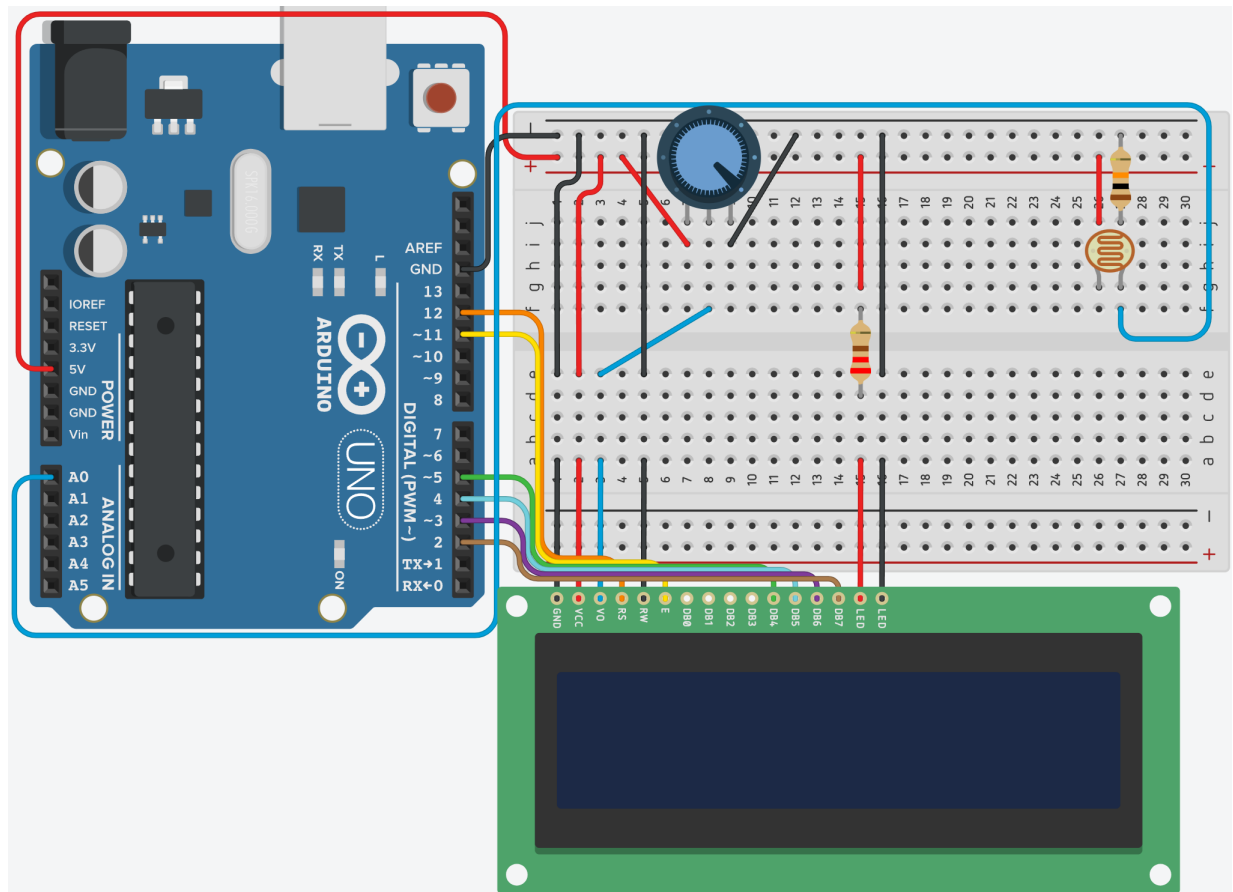
1. 1x Arduino Uno R3 Board
 2. 1x Breadboard
 3. 1x 16x2 LCD
 4. 1x Photoresistor
 5. 1x 10K Ω Potentiometer
 6. 1x 10K Ω Resistor
 7. 1x 220 Ω Resistor
 8. Many wires
-

3 Circuit Instructions

Here is a step by step guide on how to wire the circuit on the next page.

1. To get started with wiring, get your Arduino board and breadboard situated next to each other.
2. Connect a Arduino GND pin to the ground rail on your breadboard.
3. Next connect your Arduino 5V power pin to the source rail on your breadboard.
4. On your breadboard, place your photoresistor on two rails, connect one lead to the source rail. Connect the other lead via the 10K Ω resistor to the ground rail. On the grounded lead rail, connect a wire from there to your Arduino's analog in A0 pin. This part sends a resistance value based on the light level.
5. Place your 10K Ω Potentiometer on 3 rails, wire the outer rails to the source and ground rails. The middle rail will be wired in a later step, this part controls the LCD's contrast.
6. Place your 16x2 LCD on your breadboard and wire the connections like so:

16x2 LCD	Wire	Arduino	Potentiometer	Note
VSS (GND)	Ground	GND	–	–
VDD (VCC)	Source	5V	–	–
V0 (V0)	Blue	–	Middle rail	–
RS (RS)	Orange	12	–	–
RW (RW)	Ground	GND	–	–
E (E)	Yellow	11	–	–
D4 (DB4)	Green	5	–	–
D5 (DB5)	Turquoise	4	–	–
D6 (DB6)	Purple	3	–	–
D7 (DB7)	Brown	2	–	–
A (LED)	Source	5V	–	via 220 Ω
K (LED)	Ground	GND	–	–



6 Sample Code

Please install the LiquidCrystal library in your Arduino IDE to run the following code. The goal of the code is to update the LCD while reading the photoresistor's analog values.

```
/* rmagd2-Lab-04.ino */
/* -----
>> Assignment details and provided code are created and owned by Patrick Troy.
>> University of Illinois Chicago - CS 362, Spring 2024
>> -----
>> File    : rmagd2-Lab-04.ino :: Lab 4 - Photoresistor (LDR { Light Dependent Resistor)
>> Author  : Ryan Magdaleno
>> UIN/nID: 668523658 (rmagd2)
>> System  : Windows 10 w/ Mingw-w64
>> TA      : David Levit
- - - - -
>> References used ::
>> https://docs.wokwi.com/parts/wokwi-photoresistor-sensor
- - - - -
>> File overview ::
>> This program makes use of a 16x2 LCD along with a photoresistor. The photoresistor
>> gets the light level and sends its analog output to the arduino to be displayed
>> on the 16x2 with predefined light level messages. The 16x2 LCD will also be
>> displaying the time in ms since the start of the program.
----- */

// Preprocessor directives ::
#include <LiquidCrystal.h>

// Global variables ::
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
byte prevLux, curLux;
String luxText[] = {
    "brightly lit",
    "fully lit",
    "medium",
    "partially dark",
    "dark"
};
int ranges[][2] = {
    {501, 2000},
    {201, 500},
    {51, 200},
    {16, 50},
    {0, 15}
};
```

```
void setup()
{
    // Set up devices ::
    lcd.begin(16, 2);

    // Set initial values on lcd ::
    prevLux = getLuxIndex();
    lcd.print(luxText[prevLux]);
    lcd.setCursor(0, 1);
    lcd.print("          ");
    lcd.print(millis());
}

// Retrieve corresponding text index ::
byte getLuxIndex()
{
    int lux = analogRead(A0);
    for (byte i = 0; i < 5; ++i) {
        if (lux >= ranges[i][0] && lux <= ranges[i][1]) {
            return i;
        }
    }
    return -1;
}

void loop()
{
    curLux = getLuxIndex();
    if (prevLux != curLux) {
        lcd.setCursor(0, 0);
        lcd.print(luxText[curLux]);
        prevLux = curLux;
    }
    lcd.print(millis());
    lcd.setCursor(0, 1);
}
```

7 References

1. [Photoresistor tutorial](#)
 2. [16x2 LCD tutorial](#)
 3. [LiquidCrystal library documentation](#)
 4. [Potentiometer tutorial](#)
-

8 Experimenting

1. I determined the ranges by covering the photoresistor with my hands and read the lux value via the serial monitor, this was my "dark" values, $\approx 0 - 15$ lux.
 2. I then used a flashlight and checked documentation that the max lux value is 1023, so I used 501 – 2000 for my "brightly lit" values.
 3. Everything in between was done by moving my flashlight away and letting the photoresistor enter darkness, that's how I got my "partially dark", "medium", and "fully lit" ranges.
-

9 Conclusion

This was a fun lab, I learned how to use a photoresistor. My final project for this class might rely on resistors in case a part of my group's final project cannot be achieved. My final project uses solenoids to press keys on a keyboard, currently my groupmate is having trouble with the solenoid circuit. The solenoid circuit caught on fire recently and we're looking for alternatives if the circuit can't be completed in time for our presentation. I thought of using photoresistors to detect the notes falling and sending the data via serial to a computer side program for virtual key presses. Writing this report in a how to manner also helped me fully understand the materials used in this lab.
