**FORMAN CHRISTIAN COLLEGE**

**(A CHARTERED UNIVERSITY)**



**Embedded Systems (CSCS 306)**

**FALL-2019**

**LAB-06**

**Controlling LED with Fuzzy Logic**

**Group Members:**

* Mohammad Usman 20-10558
* Haris Naseer 20-10612
* Abeer Butt 20-10596

**Introduction:**

The aim of the lab was to test our skills in understanding the sample code of FuzzyLogic that is provided with the library. It also tested our skills in making linguistic variables for the input (Brightness) as well as the output (Intensity of LED). Moreover, we also made the membership functions and the rule base for the given problem. This lab consists of two LabTasks.

**LabTask1:** In this LabTask, we had to download the embedded fuzzy logic library (eFLL), open the sample code of FuzzyLogic, understand it and answer various questions that are mentioned on the Lab handout.

**LabTask2:** In this LabTask, we had to set up a circuit that consisted of an LDR and an LED interfaced with Arduino. We had to change the sample code provided with the library so that the LED should glow with a very low (or zero) intensity, if light falling on LDR is very bright. The LED should glow with high intensity if there is no light falling on LDR that is complete dark. So, we had to design the fuzzy controller with appropriate linguistic variables and define the membership functions.

**Working Code:**

**LabTask:**

#include <Fuzzy.h>

// Instantiating a Fuzzy object

Fuzzy \*fuzzy = new Fuzzy();

const int led = 3; // the pin that the LED is attached to

const int ldr = 0; // the pin that ldr is attached to

int ldrVal = 0; // variable to store the ldr value

void setup()

{

// Set the Serial output

Serial.begin(9600);

pinMode(led, OUTPUT);

pinMode(ldr, INPUT); // initialize ldr as an input

// Instantiating a FuzzyInput object

FuzzyInput \*brightness = new FuzzyInput(1);

// Instantiating a FuzzySet object

FuzzySet \*veryDim = new FuzzySet(0, 30, 30, 40);

// Including the FuzzySet into FuzzyInput

brightness->addFuzzySet(veryDim);

// Instantiating a FuzzySet object

FuzzySet \*dim = new FuzzySet(30, 50, 50, 60);

// Including the FuzzySet into FuzzyInput

brightness->addFuzzySet(dim);

// Instantiating a FuzzySet object

FuzzySet \*bright = new FuzzySet(50, 70, 70, 80);

// Including the FuzzySet into FuzzyInput

brightness->addFuzzySet(bright);

// Instantiating a FuzzySet object

FuzzySet \*veryBright = new FuzzySet(70, 120, 120, 130);

// Including the FuzzySet into FuzzyInput

brightness->addFuzzySet(veryBright);

// Including the FuzzyInput into Fuzzy

fuzzy->addFuzzyInput(brightness);

// Instantiating a FuzzyOutput objects

FuzzyOutput \*intensity = new FuzzyOutput(1);

// Instantiating a FuzzySet object

FuzzySet \*low = new FuzzySet(0, 30, 30, 40);

// Including the FuzzySet into FuzzyOutput

intensity->addFuzzySet(low);

// Instantiating a FuzzySet object

FuzzySet \*average = new FuzzySet(30, 80, 80, 120);

// Including the FuzzySet into FuzzyOutput

intensity->addFuzzySet(average);

// Instantiating a FuzzySet object

FuzzySet \*high = new FuzzySet(100, 170, 170, 210);

// Including the FuzzySet into FuzzyOutput

intensity->addFuzzySet(high);

// Instantiating a FuzzySet object

FuzzySet \*veryHigh = new FuzzySet(190, 230, 230, 255);

// Including the FuzzySet into FuzzyOutput

intensity->addFuzzySet(veryHigh);

// Including the FuzzyOutput into Fuzzy

fuzzy->addFuzzyOutput(intensity);

// Building FuzzyRule "IF brightness = veryDim THEN intensity = veryHigh"

// Instantiating a FuzzyRule Antecedent objects

FuzzyRuleAntecedent \*ifBrightnessVeryDim = new FuzzyRuleAntecedent();

// Creating a FuzzyRuleAntecedent with just a single FuzzySet

ifBrightnessVeryDim->joinSingle(veryDim);

// Instantiating a FuzzyRuleConsequent objects

FuzzyRuleConsequent \*thenIntensityVeryHigh = new FuzzyRuleConsequent();

// Including a FuzzySet to this FuzzyRuleConsequent

thenIntensityVeryHigh->addOutput(veryHigh);

// Instantiating a FuzzyRule objects

FuzzyRule \*fuzzyRule01 = new FuzzyRule(1, ifBrightnessVeryDim, thenIntensityVeryHigh);

// Including the FuzzyRule into Fuzzy

fuzzy->addFuzzyRule(fuzzyRule01);

// Building FuzzyRule "IF brightness = dim THEN intensity = high"

// Instantiating a FuzzyRule Antecedent objects

FuzzyRuleAntecedent \*ifBrightnessDim = new FuzzyRuleAntecedent();

// Creating a FuzzyRuleAntecedent with just a single FuzzySet

ifBrightnessDim->joinSingle(dim);

// Instantiating a FuzzyRuleConsequent objects

FuzzyRuleConsequent \*thenIntensityHigh = new FuzzyRuleConsequent();

// Including a FuzzySet to this FuzzyRuleConsequent

thenIntensityHigh->addOutput(high);

// Instantiating a FuzzyRule objects

FuzzyRule \*fuzzyRule02 = new FuzzyRule(2, ifBrightnessDim, thenIntensityHigh);

// Including the FuzzyRule into Fuzzy

fuzzy->addFuzzyRule(fuzzyRule02);

// Building FuzzyRule "IF brightness = bright THEN intensity = average"

// Instantiating a FuzzyRule Antecedent objects

FuzzyRuleAntecedent \*ifBrightnessBright = new FuzzyRuleAntecedent();

// Creating a FuzzyRuleAntecedent with just a single FuzzySet

ifBrightnessBright->joinSingle(bright);

// Instantiating a FuzzyRuleConsequent objects

FuzzyRuleConsequent \*thenIntensityAverage = new FuzzyRuleConsequent();

// Including a FuzzySet to this FuzzyRuleConsequent

thenIntensityAverage->addOutput(average);

// Instantiating a FuzzyRule objects

FuzzyRule \*fuzzyRule03 = new FuzzyRule(3, ifBrightnessBright, thenIntensityAverage);

// Including the FuzzyRule into Fuzzy

fuzzy->addFuzzyRule(fuzzyRule03);

// Building FuzzyRule "IF brightness = veryBright THEN intensity = low"

// Instantiating a FuzzyRule Antecedent objects

FuzzyRuleAntecedent \*ifBrightnessVeryBright = new FuzzyRuleAntecedent();

// Creating a FuzzyRuleAntecedent with just a single FuzzySet

ifBrightnessVeryBright->joinSingle(veryBright);

// Instantiating a FuzzyRuleConsequent objects

FuzzyRuleConsequent \*thenIntensityLow = new FuzzyRuleConsequent();

// Including a FuzzySet to this FuzzyRuleConsequent

thenIntensityLow->addOutput(low);

// Instantiating a FuzzyRule objects

FuzzyRule \*fuzzyRule04 = new FuzzyRule(4, ifBrightnessVeryBright, thenIntensityLow);

// Including the FuzzyRule into Fuzzy

fuzzy->addFuzzyRule(fuzzyRule04);

}

void loop()

{

ldrVal = analogRead(ldr);

Serial.print("Brightness: ");

Serial.println(ldrVal);

// Set ldr value as an input

fuzzy->setInput(1, ldrVal);

// Running the Fuzzification

fuzzy->fuzzify();

// Running the Defuzzification

int output = fuzzy->defuzzify(1);

// Printing something

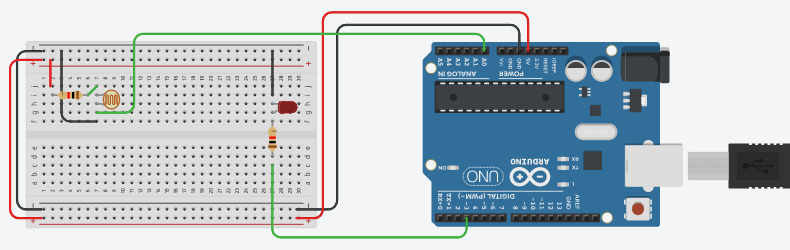
analogWrite(led, output);

Serial.print("Intensity: ");

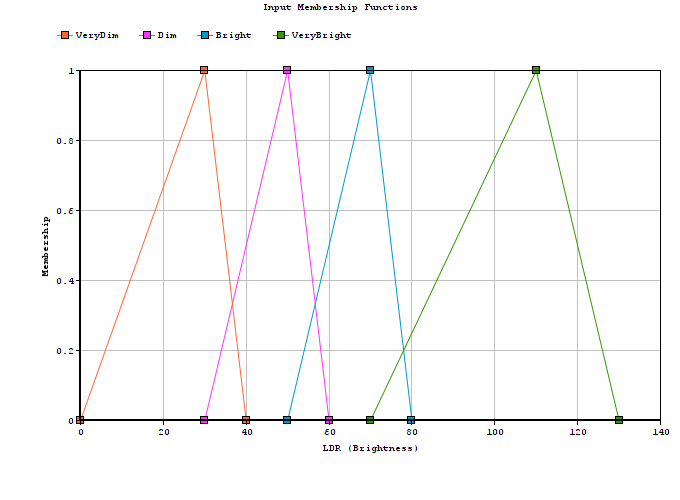
Serial.println(output);

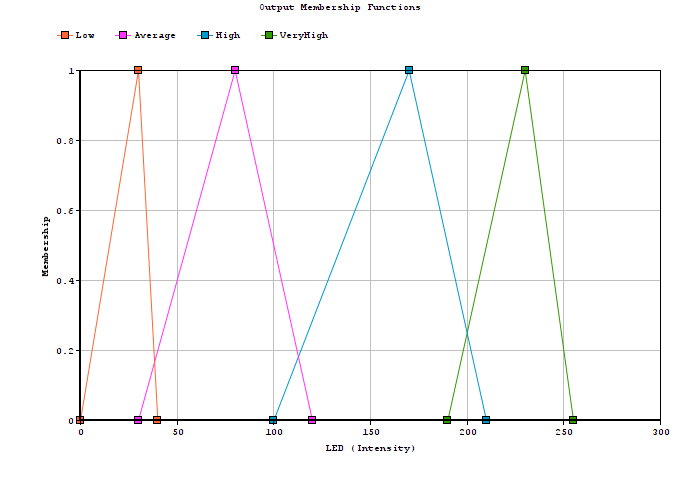
}

**Circuit Diagram:**

****

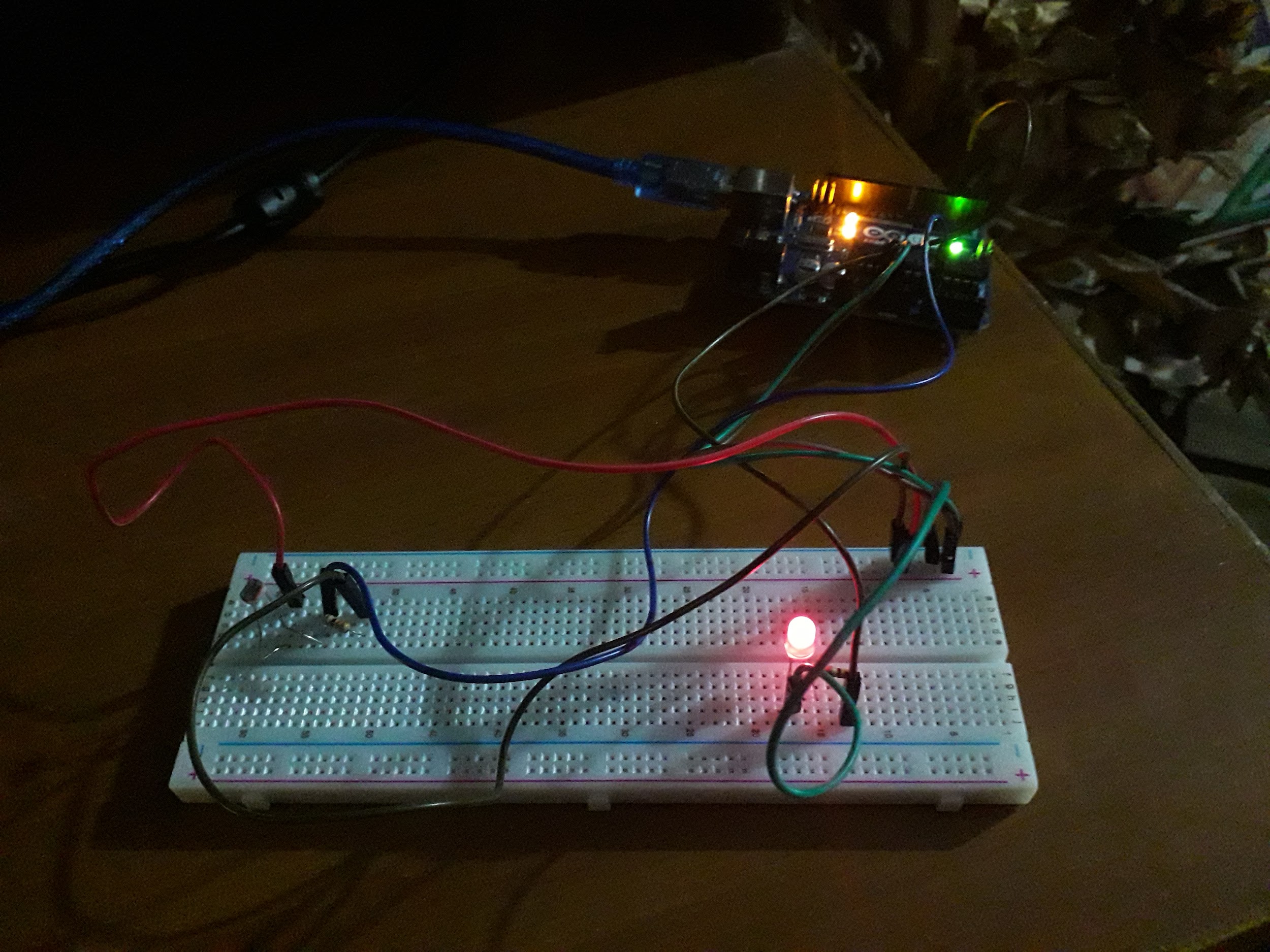
**Membership Functions:**

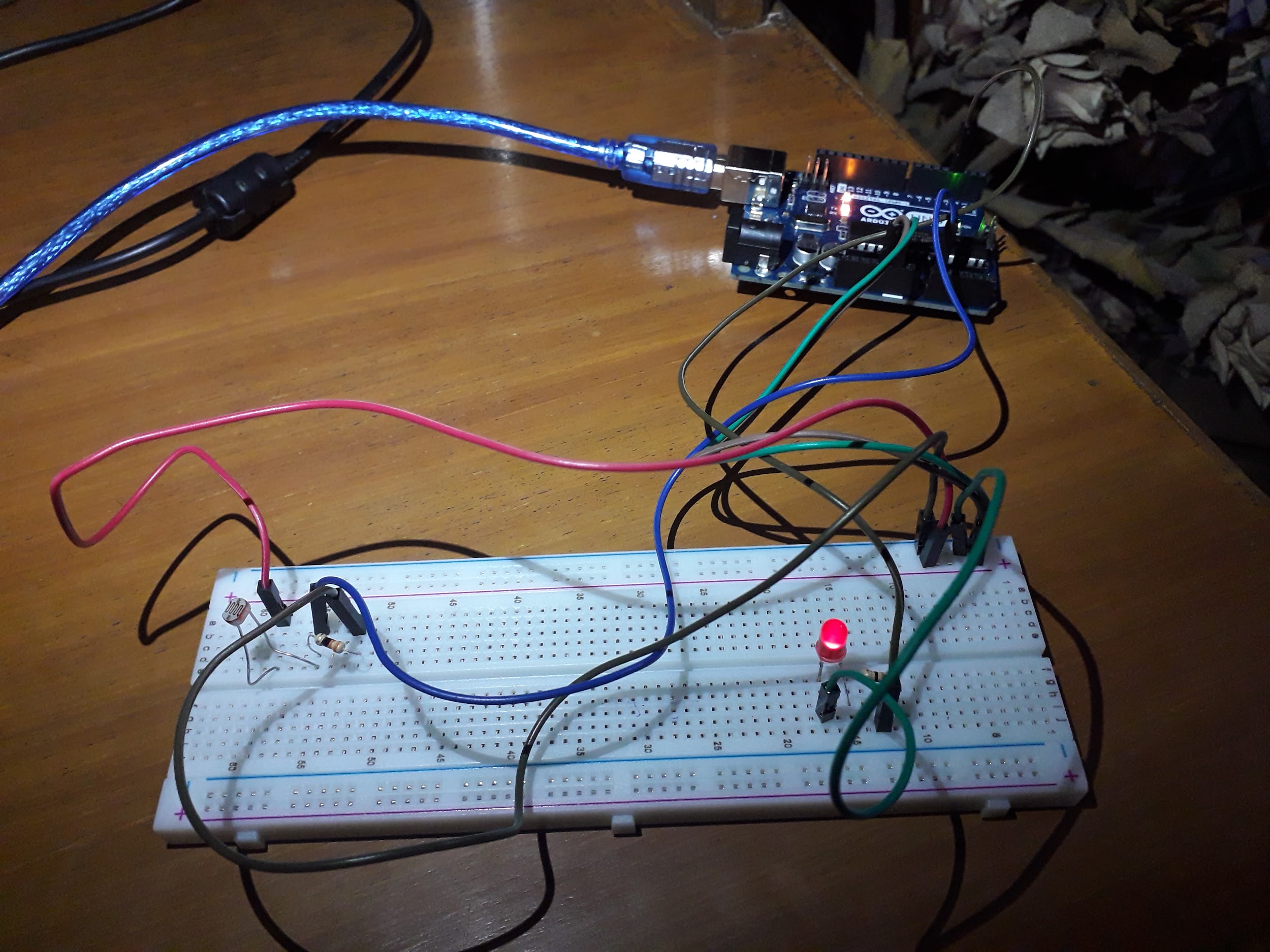
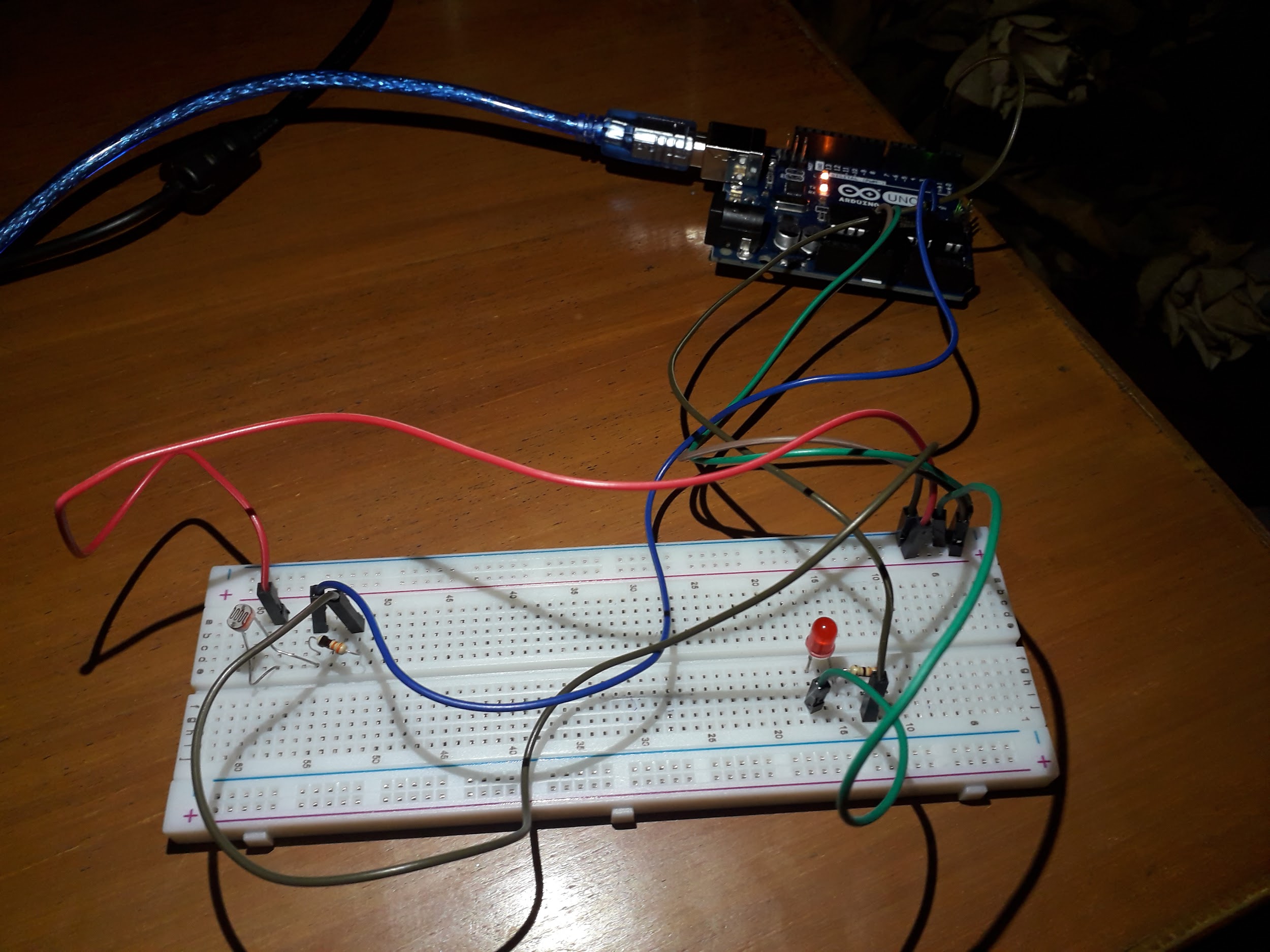
****

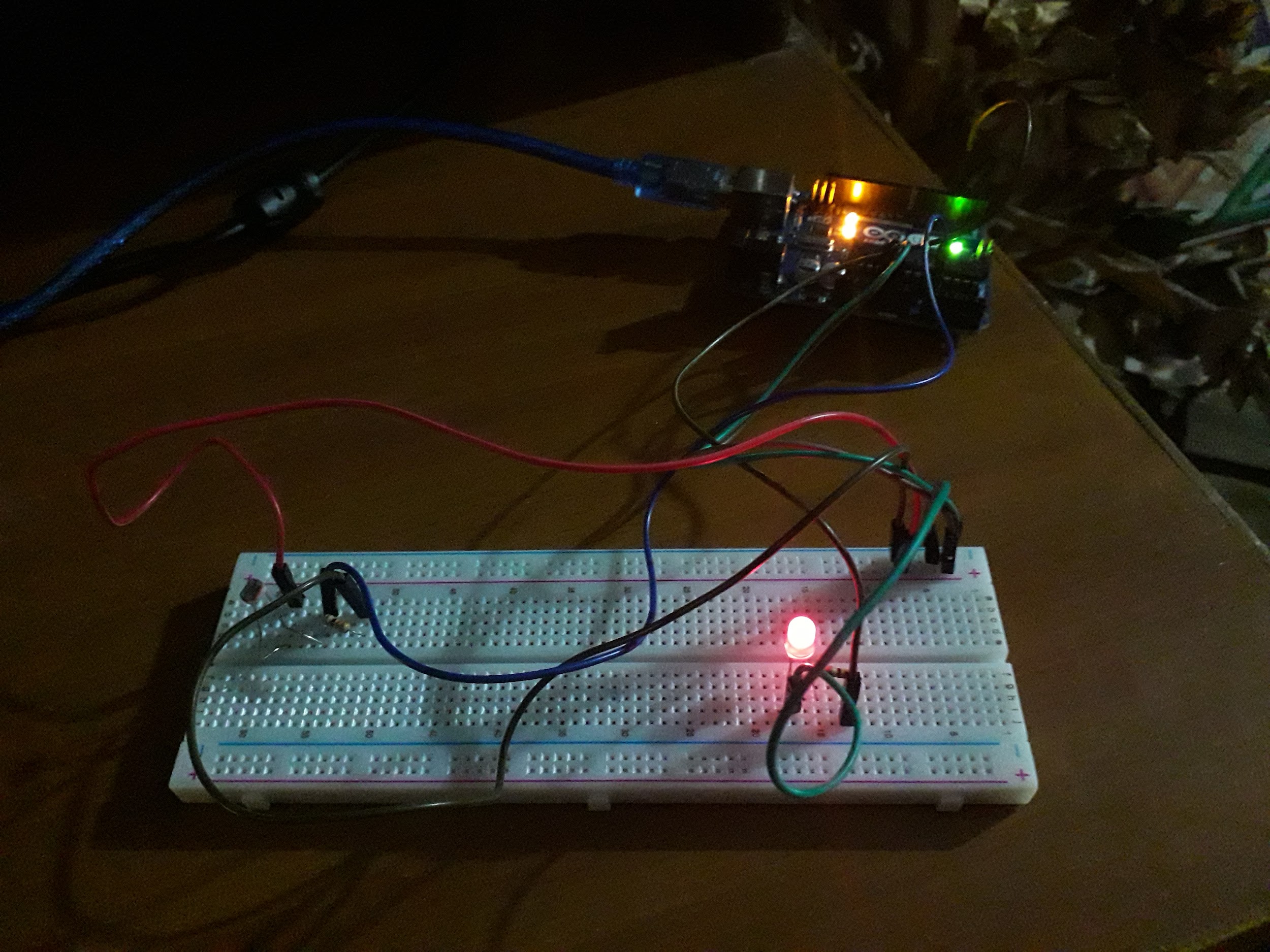
****

**Rule Base:**

1. if Brightness == VeryDim then Intensity = VeryHigh
2. if Brightness == Dim then Intensity = High
3. if Brightness == Bright then Intensity = Average
4. if Brightness == VeryBright then Intensity = Low

**Running Project Image:**

**Running Program Screenshots:**

****

**References:**

* <https://www.arduino.cc/reference/en/language/functions/communication/serial/println/>
* <https://blog.zerokol.com/2012/09/arduinofuzzy-fuzzy-library-for-arduino.html>
* <https://create.arduino.cc/projecthub/yqmt8708/how-to-control-led-brightness-539d5b>
* <https://www.tinkercad.com/>
* <https://www.onlinecharttool.com/>