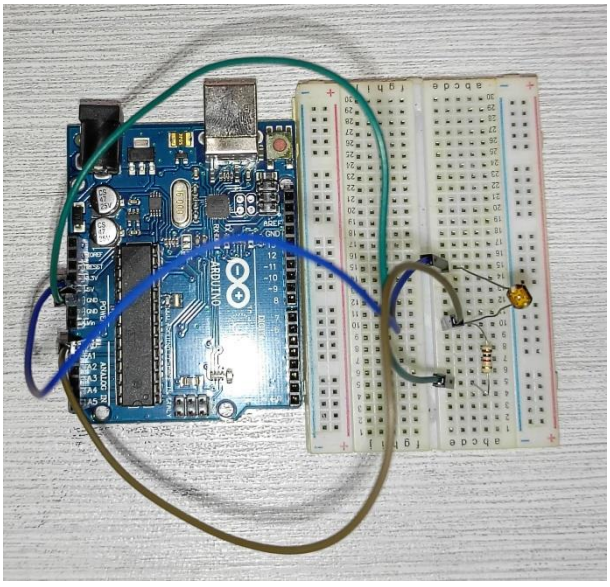
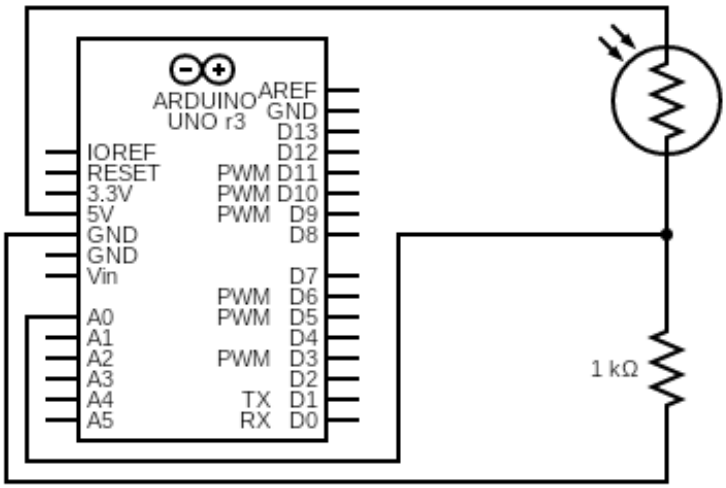
LAB TASKS

Q1: Develop the program to read values of LDR and display it on serial plotter. Now save this image and add in your lab report.

```
#define LDRPin A0 // Definning a global variable for LDR pin

int ldrValue; // Initializing a variable to store data
int brightness; // Initializing a variable to store data

void setup() {
  Serial.begin(9600); // Calling serial monitor for data display
}

void loop() {
  ldrValue = analogRead(LDRPin); // Reading data from LDR and storing in variable
  brightness = map(ldrValue, 0, 1023, 0, 255); // Mapping analog data to digital data and
storing in variable

  Serial.print("Analog Value = "); // Printing data to serial monitor
  Serial.print(ldrValue); // Printing value of ldrValue to serial monitor
  Serial.print(", Digital Value = "); // Printing data to serial monitor
  Serial.println(brightness); // Printing value of brightness to serial monitor

  delay(500); // Adding a delay
}
```

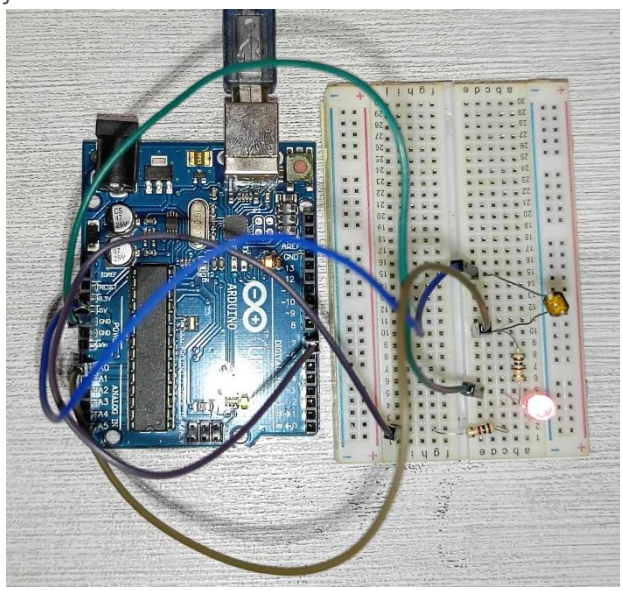


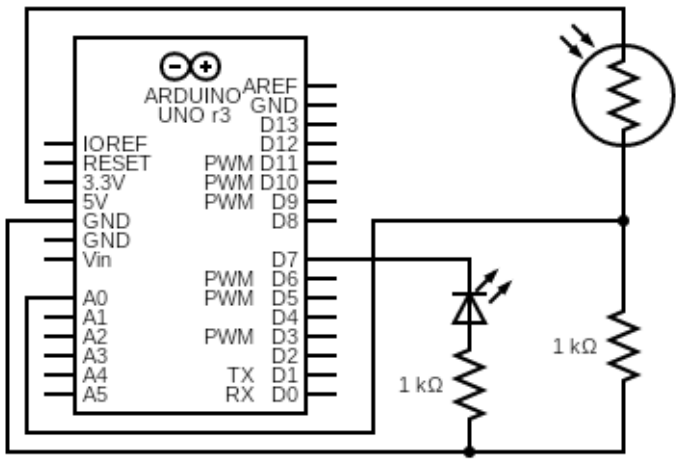


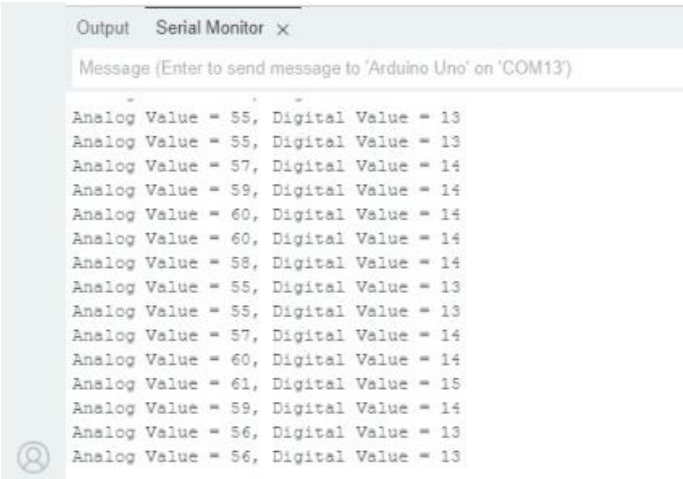
Q2: Develop a program which controls the brightness of LED with the help of an LDR and display its reading on serial monitor/serial plotter. Add this in your lab report. LED should turn ON only when it's dark as discussed in Case 1.

```
#define LDRPin A0 // Definning a global variable for LDR pin
#define LEDPin 7 // Definning a global variable for LED pin

int ldrValue; // Initializing a variable to store data
int brightness; // Initializing a variable to store data

void setup() {
```

```
  Serial.begin(9600); // Calling serial monitor for data display
  pinMode(LEDPin, OUTPUT); // Defining  port location for LED
}

void loop() {
  ldrValue = analogRead(LDRPin); // Reading data from LDR and storing in variable
  brightness = map(ldrValue, 0, 1023, 0, 255); // Mapping analog data to digital data and
storing in variable
  analogWrite(LEDPin, brightness); // Assigning LED with brightness value as input

  Serial.print("Analog Value = "); // Printing data to serial monitor
  Serial.print(ldrValue); // Printing value of ldrValue to serial monitor
  Serial.print(", Digital Value = "); // Printing data to serial monitor
  Serial.println(brightness); // Printing value of brightness to serial monitor

  delay(500); // Adding a delay
}
```



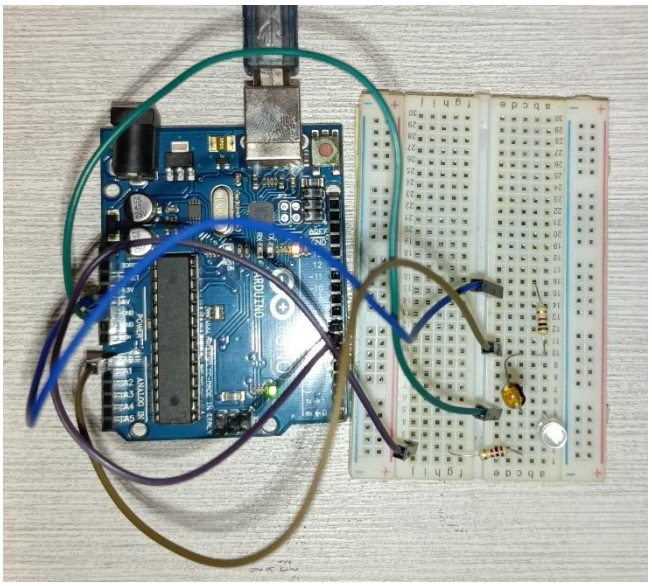


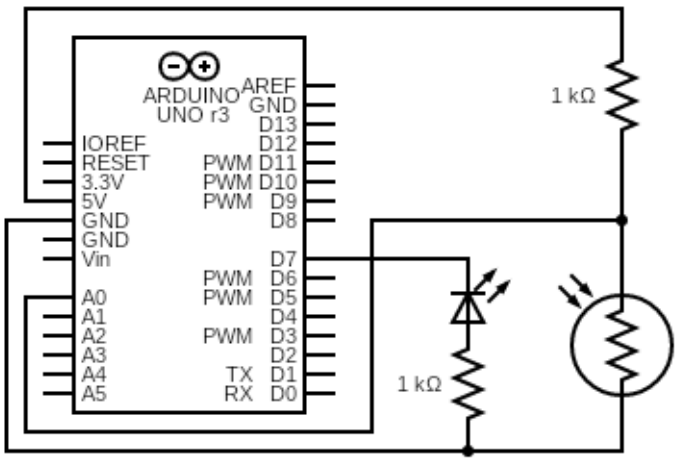


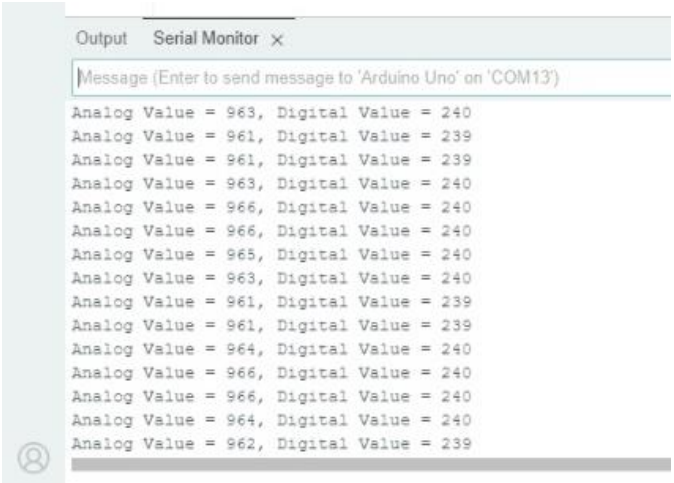
Q3: Develop a program which controls the brightness of LED with the help of an LDR and display its reading on serial monitor/serial plotter. For Case 2, LED should turn OFF when it's dark.

```
#define LDRPin A0 // Definning a global variable for LDR pin
#define LEDPin 7 // Definning a global variable for LED pin
```

```
int ldrValue; // Initializing a variable to store data
int brightness; // Initializing a variable to store data

void setup() {
  Serial.begin(9600); // Calling serial monitor for data display
  pinMode(LEDPin, OUTPUT); // Defining  port location for LED
}

void loop() {
  ldrValue = analogRead(LDRPin); // Reading data from LDR and storing in variable
  brightness = map(ldrValue, 0, 1023, 0, 255); // Mapping analog data to digital data and
storing in variable
  analogWrite(LEDPin, brightness); // Assigning LED with brightness value as input

  Serial.print("Analog Value = "); // Printing data to serial monitor
  Serial.print(ldrValue); // Printing value of ldrValue to serial monitor
  Serial.print(", Digital Value = "); // Printing data to serial monitor
  Serial.println(brightness); // Printing value of brightness to serial monitor

  delay(500); // Adding a delay
}
```







POST LAB TASKS

Q1: Suggest a circuit that can made using LED and LDR combination that can be implemented at your home. Give a brief summary of their working.

We face gas outages during night times, due to which a main problem that arises in the morning is the shutdown of gas water heaters (geysers). The weather being winter worsen this condition. Every day, in morning, we have to go outside and light up our water heaters and wait for the water to get warm before using it.

To solve this problem, I propose to design an automatic geyser lighter using Arduino. How it will work is by detecting the gas supply input and lighting fire on the stove of the geyser through a mechanism. Using LDR the circuit will detect if the fire has been lit because heat will start to generate and the temperature will increase around the stove. LED can be used to indicate on and off of the lit stove to the user inside the house, so he/she do not have to get out of the house to check.

This application can be further extended by adding a water temperature and displaying it as a level using 5 LEDs. 0 lit LEDs would mean the water is cold and 5 lit LEDs would mean the water is hot.

Q2: Explain Pulse Width Modulation with the help of graph.

PWM or Pulse Width Modulation generates a signal compromising a train of pulses in form of a square wave. At any given point this wave is either high or low.

For example, considering a 5V PWM signal, the PWM signal will either be 5V (high) or ground level 0V (low). The duration the signals stay high is the “on time” and the duration it stays low is the “off time”.

Two important parameters associated with PWM are the PWM duty cycle and PWM frequency.

The duty cycle represents the percentage of time where the PWM signal remain HIGH (on time). If the signal always ON, its 100% duty cycle. If the signal is always OFF, its 0% duty cycle. If the signal is ON for half time period and OFF for half time period, its 50% duty cycle. This can be observed in the below image.

The frequency of a PWM signal determines how fast a PWM completes its one cycle. One cycle is the complete ON and OFF time of a PWM signal as shown in the following figure.