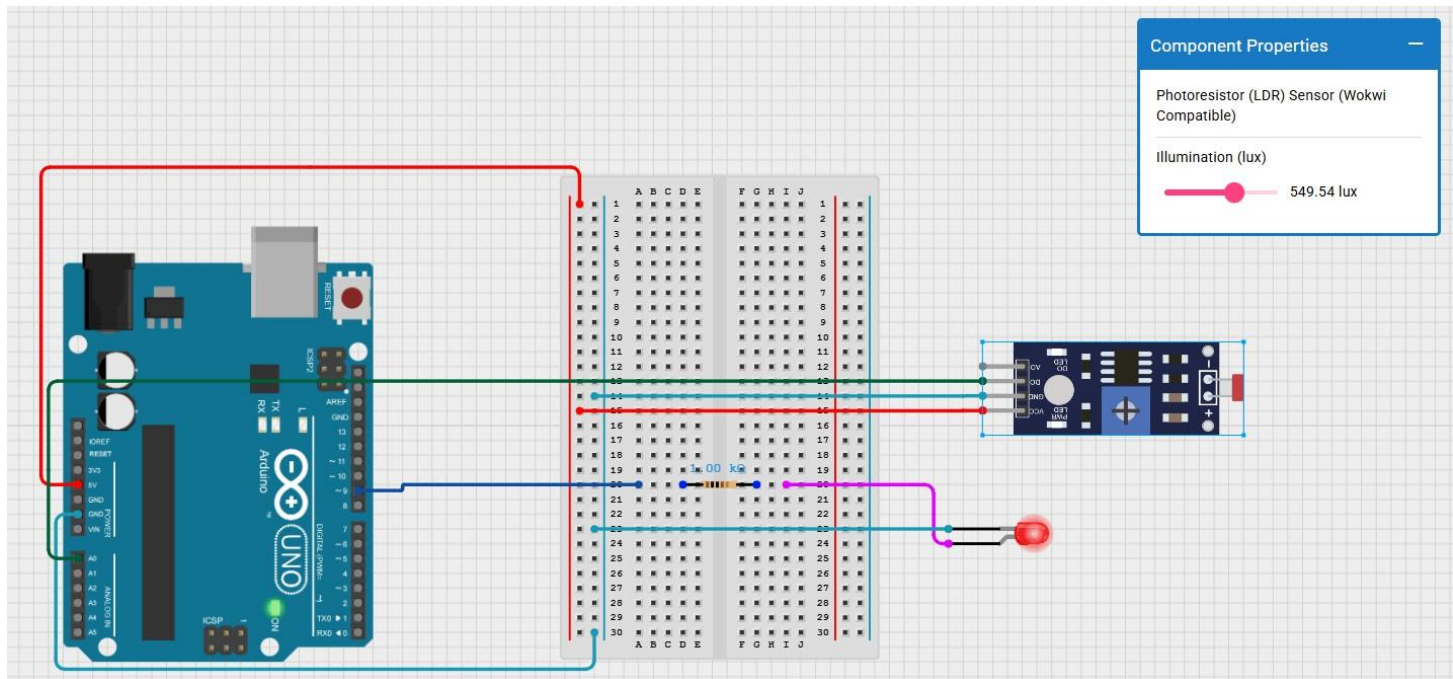# Practical No. 6

Aim: Use different types of sensors (LDR, Temperature) with Raspberry Pi/Uno.

1) Virtual


```
1    const int ledPin = 9; // Pin connected to the LED
2    const int ldrPin = A0; // Pin connected to the LDR
3
4    void setup() {
5      pinMode(ledPin, OUTPUT); // Set LED pin as output
6      pinMode(ldrPin, INPUT);  // Set LDR pin as input
7    }
8
9    void loop() {
10     int ldrValue = analogRead(ldrPin); // Read the value from the LDR
11
12     if (ldrValue < 500) { // Adjust the threshold as needed
13       digitalWrite(ledPin, HIGH); // Turn LED on if it's bright
14     } else {
15       digitalWrite(ledPin, LOW); // Turn LED off if it's dark
16     }
17
18     delay(100); // Small delay for stability
19   }
```

2) <u>Physical</u>

OVERVIEW OF LDR LIGHT SENSOR MODULE

The LDR light sensor module can be used to find out if there is light or measure how much light is present in the surroundings. It has two options: a digital output pin and an analog output pin.

PINOUT

The LDR light sensor module has four pins with different functions:

◆— **VCC pin:** Connect it to a power source (between 3.3V to 5V).

◆— **GND pin:** Connect it to the ground (0V).

◆— **DO pin:** This is a digital output pin. It goes **HIGH** when it's dark and **LOW** when it's light. You can adjust the threshold between darkness and lightness using a built-in potentiometer.

◆— **AO pin:** This is an analog output pin. The output value decreases as the light gets brighter and increases as the light gets darker.
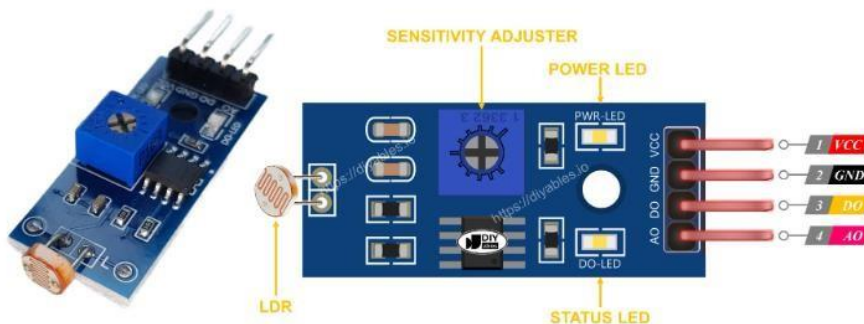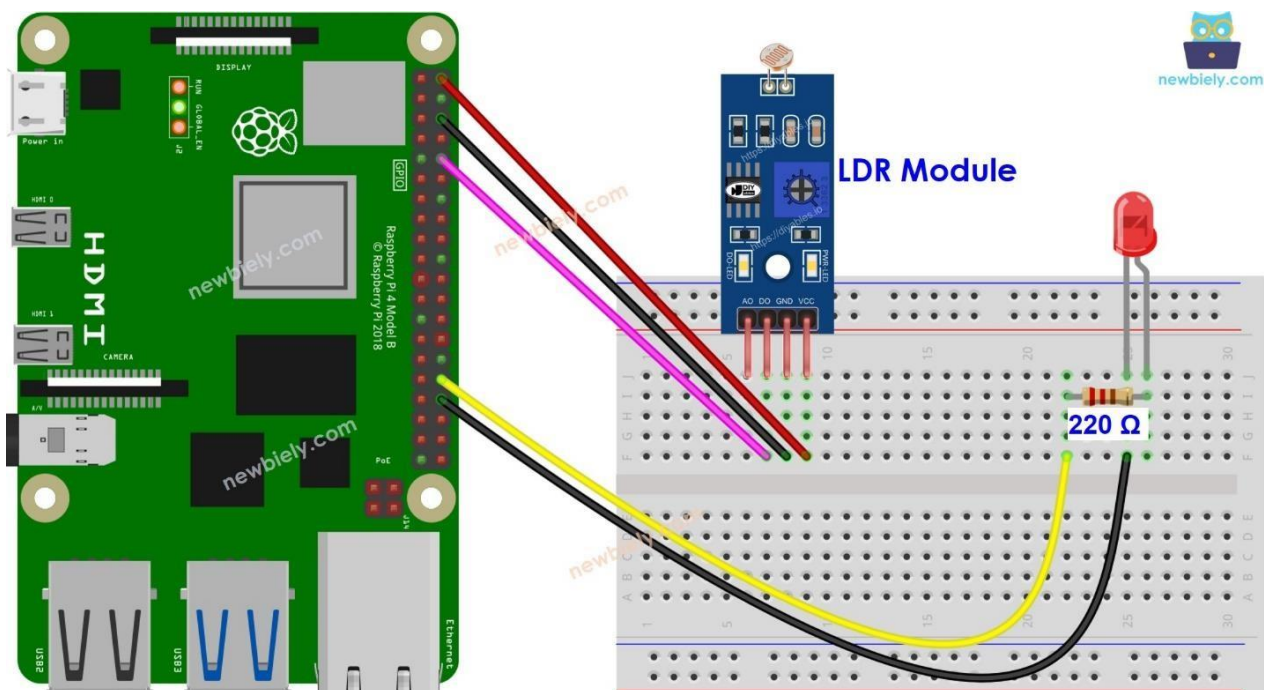

image source: diyables.io

Additionally, the LDR light sensor module has two LED indicators:

◆— The PWR-LED indicator shows the power status.

◆— The DO-LED indicator shows the state of light on the DO pin. It is illuminated when there is light and turned off when it is dark.



```
sudo apt-get update
sudo apt-get install python3-rpi.gpio
```

```python
1    import RPi.GPIO as GPIO
2    import time
3
4    # Set the GPIO mode to BCM
5    GPIO.setmode(GPIO.BCM)
6
7    # Define the PIN numbers for the light sensor and the LED
8    LIGHT_SENSOR_PIN = 15
9    LED_PIN = 12
10
11   # Set up the light sensor PIN as an input with a pull-up resistor
12   GPIO.setup(LIGHT_SENSOR_PIN, GPIO.IN, pull_up_down=GPIO.PUD_UP)
13
14   # Set up the LED PIN as an output
15   GPIO.setup(LED_PIN, GPIO.OUT)
16
17   try:
18       while True:
19           # Read the state of the light sensor
20           light_state = GPIO.input(LIGHT_SENSOR_PIN)
21
22           # Control the LED based on the light sensor state
23           if light_state == GPIO.LOW:
24               # Light is present, turn off the LED
25               GPIO.output(LED_PIN, GPIO.LOW)
26               print('Light is present, turn off the LED')
27           else:
28               # Light is not present, turn on the LED
29               GPIO.output(LED_PIN, GPIO.HIGH)
30               print('Light is not present, turn on the LED')
31
32           # Small delay to avoid rapid state changes
33           time.sleep(1)
34
35   except KeyboardInterrupt:
36       # Clean up GPIO settings on program exit
37       GPIO.cleanup()
```
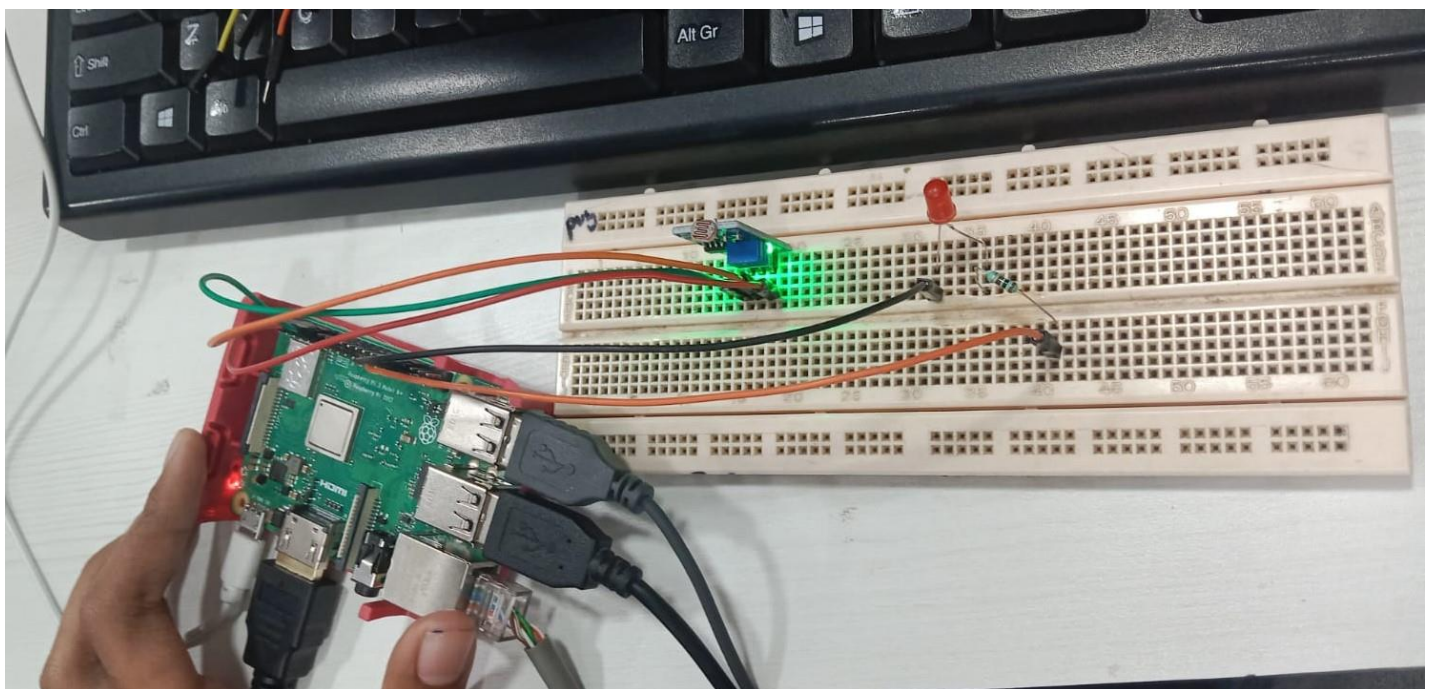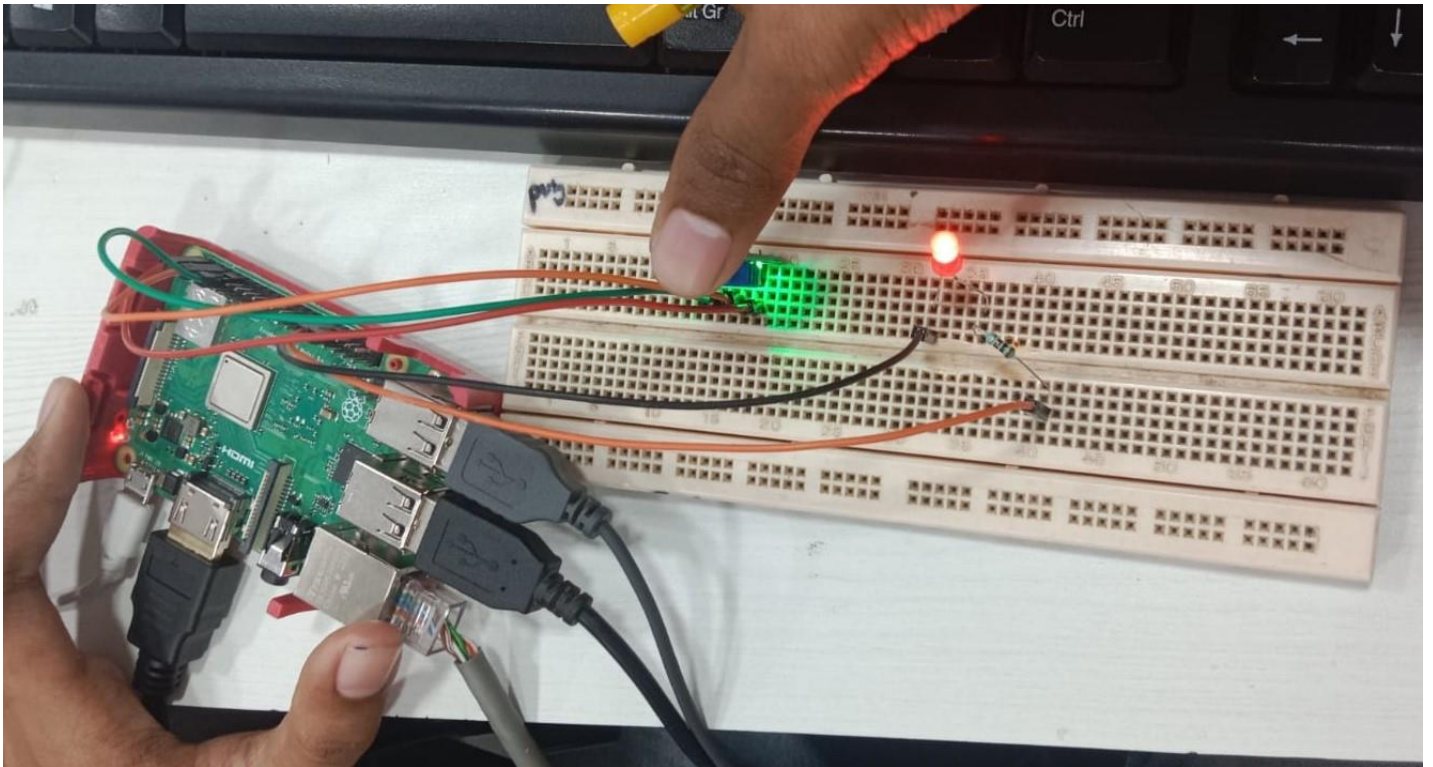
Conclusion: Hence we have successfully completed LDR Sensor with Raspberry using Uno/Raspberry Pi.