Practical 7

Aim: - To interface **PIR Sensor** with Arduino.

Components Required :

Arduino Board, Bread Board, PIR (Passive Infrared Sensor), Resistors, Connecting wires.

Theory:

The Passive Infra-Red sensors or PIR sensors detect motion or movement of an object that detect infrared radiations, such as the human body. Hence, the use of sensors is very common.

The advantages of using a PIR sensor are listed below:

- Inexpensive
- Adjustable module
- Efficient
- Small in size
- Less power consumption
- It can detect motion in the dark as well as light.

The PIR sensor has three terminals, which are listed below:

- VCC
- Digital Output
- GND (Ground)

We will connect the Vcc terminal of the sensor to the 5V on the Arduino board. The PIR's sensor output can be connected to any of the digital pins on the Arduino board.

The applications of the PIR sensor are automation, security systems, etc. Such sensors work great in detecting the entrance of a person in an area and leaving it.

The detection range of PIR sensors is from 5m to 12m.

Working of PIR Sensors

The working of the PIR sensor is entirely based on detecting the IR (Infra-Red) radiations, which are either emitted or reflected by the objects.

The infrared radiations are detected by the crystalline material present at the center of the sensor.

Consider a person passing in front of the background like a wall, etc. The temperature changes from room to body temperature and vice-versa within the sensor field. Changes arising in the arrival infrared radiations are converted by the sensor to the output voltage. It later detects the human body or object.

Principle

The movement of jumper present on the sensor on the L side will cause a change in the state of the sensor whenever the motion is detected. Such a condition is defined as a single trigger mode. When the sensor resets the timer after every detection of motion, it is defined as repeated trigger mode.

The two potentiometers present on the sensor are called as **Sensitivity** Potentiometer and **Time** Potentiometer. We can adjust both the parameters (time and sensitivity) accordingly. It should be restricted for atleast 15 seconds in front of the PIR sensor for proper calibration in the output. After 15 seconds, the sensor can easily detect movements.

If any movement is detected, the LED will be HIGH. If there is no such movement, the output will be LOW.

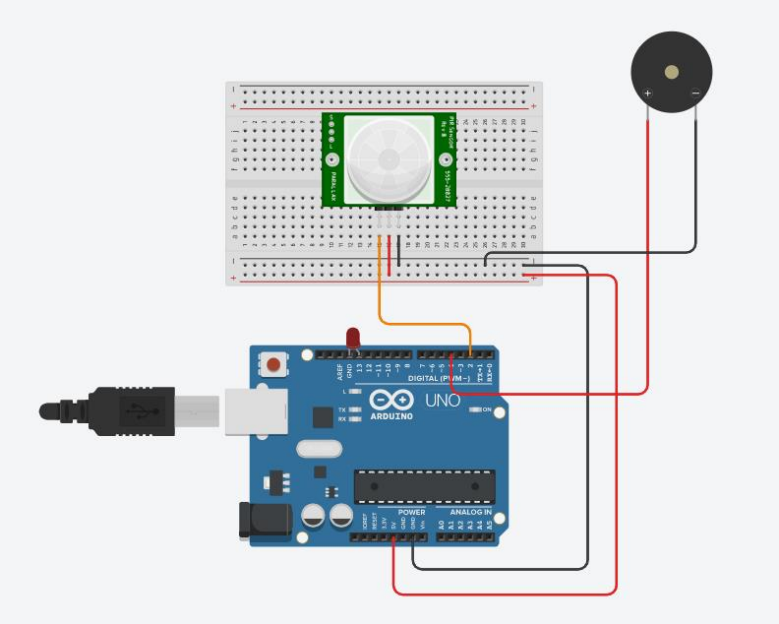
Implementation :

The connection of the practical is as below:

The steps to set up the connection are listed below:

- Connect the Vcc terminal of the PIR sensor to the 5V pin of the Arduino board.
- Connect the Output terminal of the PIR sensor to pin 8 of the Arduino board.
- Connect the GND terminal of the PIR sensor to the Ground pin of the Arduino board.
- Connect the positive leg of the LED in series with 220 Ohm resistor to pin 13 of the Arduino board.
- Connect the negative terminal of the LED to the Ground pin of the Arduino board.

Circuit Diagram:



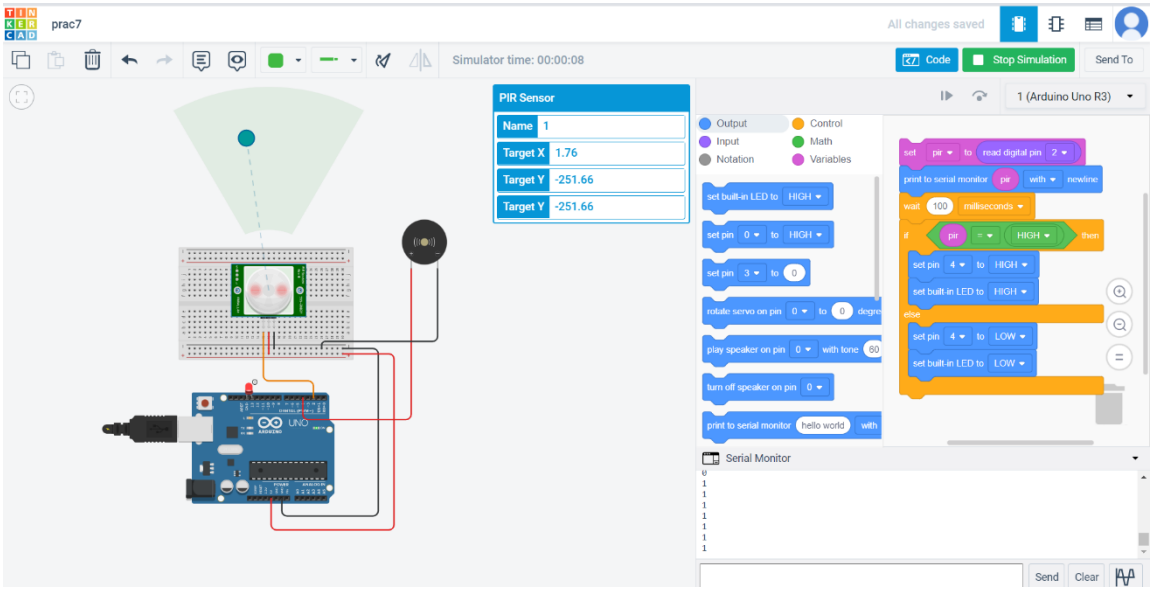
Code:

```
// C++ code
//
int pir = 0;

void setup()
{
```

```
  pinMode(2, INPUT);
  Serial.begin(9600);
  pinMode(4, OUTPUT);
  pinMode(LED_BUILTIN, OUTPUT);
}
void loop()
{
  pir = digitalRead(2);
  Serial.println(pir);
  delay(100); // Wait for 100 millisecond(s)
  if (pir == HIGH) {
    digitalWrite(4, HIGH);
    digitalWrite(LED_BUILTIN, HIGH);
  } else {
    digitalWrite(4, LOW);
    digitalWrite(LED_BUILTIN, LOW);
  }
}
```

Output :



Conclusion: Thus we studied the interfacing of Light Dependent resistor and how the resistance changes depending on the light.