DATE OF CONDUCTION -: 25/1/24

DATE OF SUBMISSION -: 8/2/24

EXPERIMENT 3

NAME -: ARJUNSINGH GAUTAM

PRN -: 22070123043

ENTC A1 2022-26

Study and working of PIR or Passive Infrared Sensors

AIM

To understand the functionality of a PIR Sensor, design a signal conditioning circuit on a breadboard and implement software.

HARDWARE USED

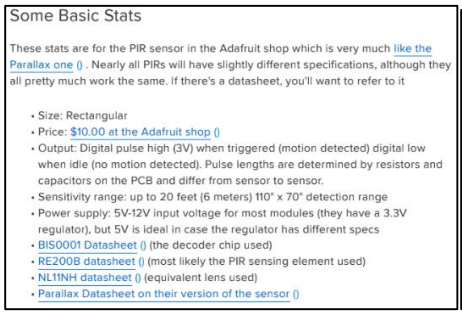
1. Arduino Board
2. One LED
3. One resistor (220-330 ohms)
4. Breadboard
5. Jumper wires (male-to-female, male-to-male)
6. USB Cable
7. Computer with Arduino IDE
8. Arduino Code
9. Power Source (optional, e.g., battery pack)
10. Passive Infrared (PIR) Sensor

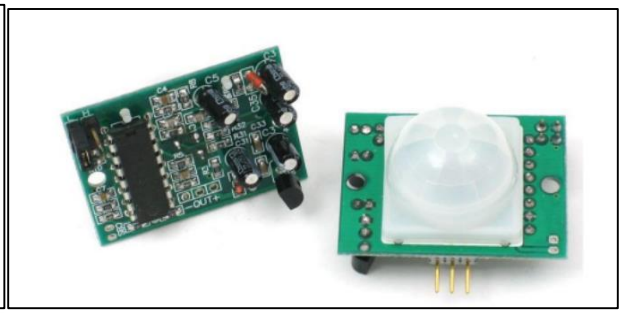
SOFTWARE USED

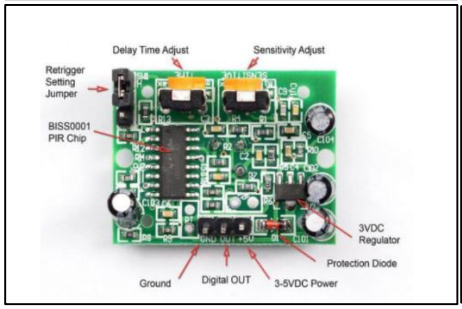
1. Arduino IDE for programming and microcontroller
2. Serial monitor for data visualization software for data analysis.

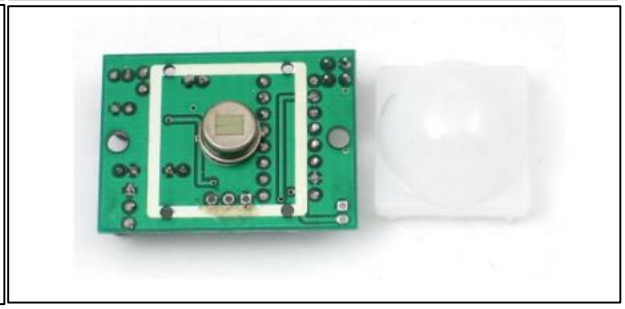
SYESETM INFORMATION

A Passive Infrared (PIR) sensor is a type of electronic sensor that measures infrared (IR) light radiating from objects in its field of view. PIR sensors are commonly used for motion detection in various applications, such as security systems, automatic lighting, and occupancy sensing. They are basically made of pyroelectric sensor, which can detect levels of infrared radiations









Sensor Construction

The pyroelectric sensor is typically made of crystalline materials like lithium tantalate or polyvinyl fluorides' sensors may have multiple elements arranged in a specific pattern to improve sensitivity and reduce false alarms. A protective housing with a Fresnel lens or other optical components helps focus and direct the incoming IR radiation.

APPLICATIONS

- PIR sensors are widely used in security alarms to detect intruders.
- Lighting Control: PIR sensors are used in automatic lighting systems to turn lights on when motion is detected and off when the area is unoccupied.
- HVAC Systems: PIR sensors are used in occupancy sensors to control heating, ventilation, and air conditioning systems based on room occupancy.

FUNCTIONS USED IN CODE

pinMode: Configures a specified pin as INPUT to receive signals from the PIR sensor in Arduino code.

Serial.begin : is a function used to initialize communication with the computer via the serial port.

digitalRead: is a function used to read the state (high or low) of a digital input pin. It is commonly used to check the status of a digital pin.

digitalWrite: is a function used to set the state (high or low) of a digital output pin.

pirState: the Arduino prints a message to the serial monitor based on whether motion is detected (HIGH state) or not (LOW state). The specific pin number (pirPin) would be adjusted based on your actual wiring configuration.

Serial.println(sensor value): prints the sensor value obtained from analog pin A0 to the serial monitor. The println() function is particularly useful when you want each set of data to appear on a new line in the serial monitor, making it easier to read.

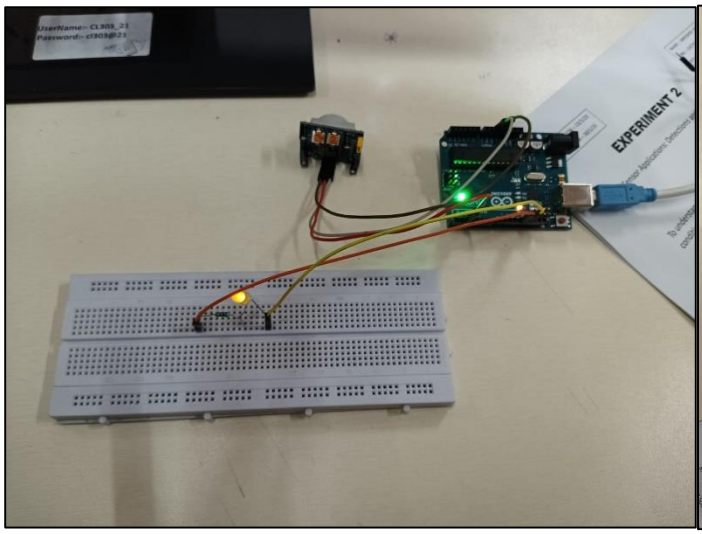
CODE: TYPED CODE

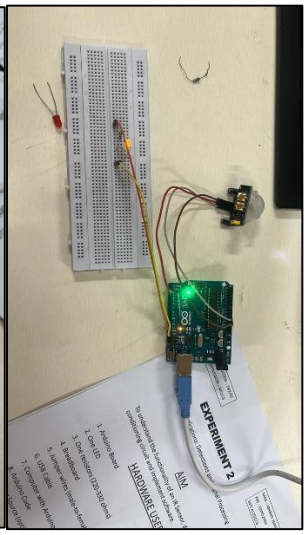
```
int LED = 13;
int Sensor = 2;
int PIRstate = LOW;
int val = 0;
void setup()
{
 pinMode(LED, OUTPUT);
 pinMode(Sensor, INPUT);
 Serial.begin(9600);
}
void loop()
{
  val = digitalRead(Sensor);
 if (val == HIGH)
 {
   digitalWrite(LED, HIGH);
   if (PIRstate == LOW)
   {
     Serial.println("Motion detected");
     PIRstate = HIGH;
   }
 }
 else
 {
  digitalWrite(LED, LOW);
  if (PIRstate == HIGH)
  {
   Serial.println("Motion ended");
   PIRstate = LOW;
  }
 }
}
```

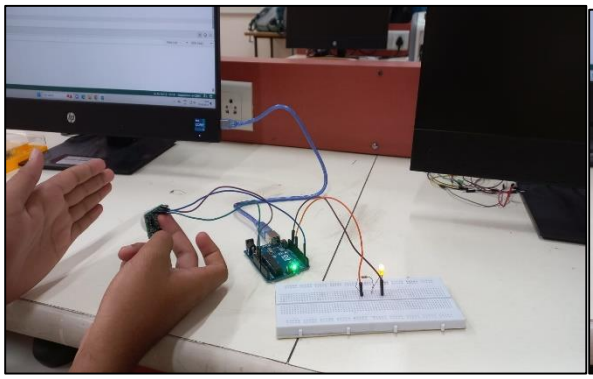
CODE: SCREENSHOT

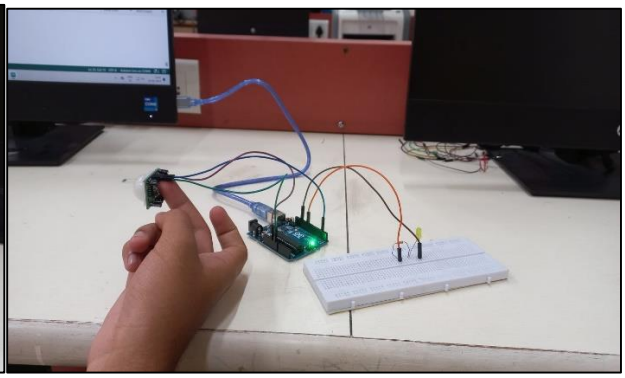
```
5   int ledPin = 13;
6   int inputPin = 2;
7   int pirState = LOW;
8   int val = 0;
9   void setup()
10  {
11    pinMode(ledPin, OUTPUT);
12    pinMode(inputPin, INPUT);
13    Serial.begin(9600);
14  }
15
16  void loop()
17  {
18    val = digitalRead(inputPin);
19    if (val == HIGH)
20    {
21      digitalWrite(ledPin, HIGH);
22      if (pirState == LOW)
23      {
24        Serial.println("Motion detected!");
25        pirState = HIGH;
26      }
27    } else {
28      digitalWrite(ledPin, LOW);
29      if (pirState == HIGH)
30      {
31        Serial.println("Motion stopped!");
32        pirState = LOW;
33      }
34    }
35  }
```

PICTURES OF THE EXPERIMENT









CONCLUSION

In this experiment, we successfully interfaced a PIR sensor with a microcontroller, designed a signal conditioning circuit, and implemented a simple program to read and display sensor values. The LED provided a visual indication when the sensor was active.