Project -1 The PIR sensor detects motion

Objective

The PIR sensor detects motion and triggers an action, such as turning on an LED or sending a message to the Serial Monitor.

```
const int pirPin = 2;      // PIR sensor output pin

const int ledPin = 13;     // LED pin

int motionDetected = 0;    // Variable to store PIR state (LOW/HIGH)

void setup() {

  pinMode(pirPin, INPUT);   // Set PIR sensor pin as input

  pinMode(ledPin, OUTPUT);  // Set LED pin as output

  Serial.begin(9600);       // Start serial communication for debugging

  Serial.println("PIR Motion Sensor Test");

}

void loop() {

  motionDetected = digitalRead(pirPin); // Read PIR sensor state

  if (motionDetected == HIGH) { // Motion detected

    digitalWrite(ledPin, HIGH); // Turn LED on

    Serial.println("Motion detected!");

    delay(1000);                // Wait 1 second to avoid multiple triggers

  } else {                      // No motion

    digitalWrite(ledPin, LOW); // Turn LED off

    Serial.println("No motion.");

    delay(100);                 // Short delay for stability

  }}
```

Project-2 : An HC-SR04 Ultrasonic Distance Sensor with an Arduino.

This sensor measures the distance to an object by sending out ultrasonic pulses and calculating the time it takes for the echo to return.

Components Required:

1. Arduino board (e.g., Uno)
2. HC-SR04 Ultrasonic Distance Sensor
3. Breadboard
4. Jumper wires
5. (Optional) A 220-ohm resistor and an LED for visual indication.

Wiring:

- **HC-SR04**:
  - **VCC**: Connect to the 5V pin of Arduino
  - **GND**: Connect to GND of Arduino
  - **Trig**: Connect to Arduino digital pin 9
  - **Echo**: Connect to Arduino digital pin 10 (via a voltage divider if needed for 3.3V boards)

- **Code**

```
// Define pins

const int trigPin = 9;

const int echoPin = 10;

// Variables for distance calculation

long duration;

int distance;
```

```
void setup() {
  // Initialize serial communication
  Serial.begin(9600);
    // Set up pin modes
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
void loop() {
  // Send a 10-microsecond pulse to trigger pin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Read the echo pin
  duration = pulseIn(echoPin, HIGH);
  // Calculate the distance in centimeters
  distance = duration * 0.034 / 2; // Speed of sound: 343 m/s (0.034 cm/µs)
  // Print the distance to the Serial Monitor
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  // Small delay to stabilize
  delay(100);
}
```

lock system!

HW: Measuring Distance Accurately based on IR

You want to measure how far objects are from the Arduino with more accuracy than previous example.

Solution

Infrared (IR) sensors generally provide an analog output that can be measured using analogRead. They can have greater accuracy than ultrasonic sensors, with

a smaller range (a range of 10 cm to 1 m or 2 m is typical for IR sensors). This sketch provides similar functionality to previous one, but it uses an infrared sensor—the Sharp GP2Y0A02YK0F (Figure below shows the connections):