



STRIVICTA

STRIVICTA'S

Elevate Handbook



STRIVICTA

About us

Strivicta is an educational initiative focused on helping students learn technology through hands-on experience. We design practical electronics kits and conduct interactive workshops where students build real projects using sensors and development boards. Our goal is to make learning engaging, practical, and innovation-driven, encouraging students to explore engineering and develop problem-solving skills.



Table of Contents

Page No

- 01** Software Requirements
- 02** Components
- 03** Smart LED Alert System
- 04** Precision Motion Control System
- 05** Intelligent Obstacle Avoidance System
- 06** Intrusion Detection Door Security System
- 07** Automated Toll Collection Barrier System
- 08** Fire Hazard Detection & Alert System
- 09** IoT-Based Smart Parking Management System
- 10** Room Climate Monitor
- 11** Rain Detection & Alert System
- 12** Laser-Based Perimeter Security System

Software Requirements

Arduino IDE



Before running the experiments in this manual, the required software and libraries must be installed on your computer. The Arduino IDE is used to write, compile, and upload programs to the Arduino Uno board.

1. Install Arduino IDE

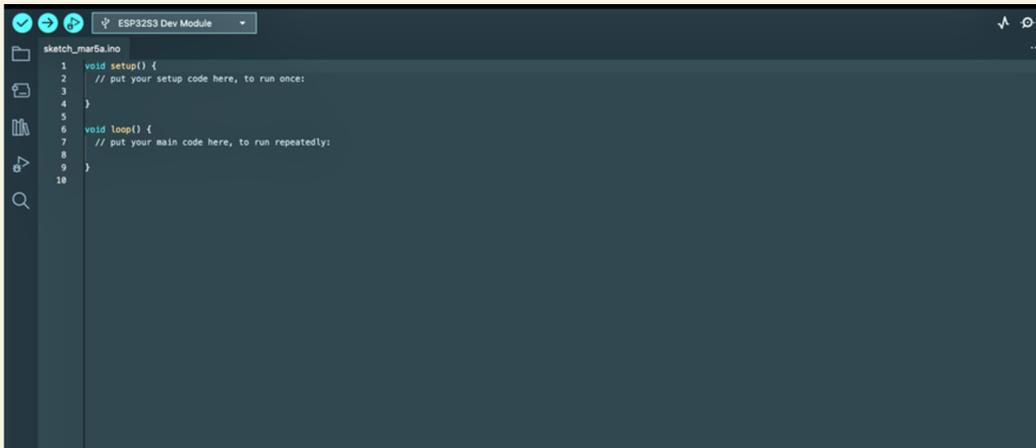
The Arduino IDE (Integrated Development Environment) is the software used to program the Arduino Uno.

Step 1: Download Arduino IDE

1. Open a web browser.
2. Go to the official Arduino website:
3. <https://www.arduino.cc/en/software>
4. Select the version suitable for your operating system:
 - Windows
 - macOS
 - Linux
5. Download the installer.

Step 2: Install Arduino IDE

1. Open the downloaded installer file.
2. Click I Agree to accept the license agreement.
3. Select all installation components.
4. Click Install.
5. After installation is complete, click Finish to launch the Arduino IDE.



2. Install Required Libraries

Some experiments in this manual require additional libraries. Install them before running the programs.

Steps to Install Libraries

1. Open Arduino IDE.
2. Click Sketch → Include Library → Manage Libraries.
3. The Library Manager window will open.
4. Search for the following libraries and install them.

Required Libraries

1. DHT Sensor Library (by Adafruit)

Used for the DHT11 Temperature and Humidity Sensor project.

Steps:

- In Library Manager, search for DHT sensor library.
- Install DHT sensor library by Adafruit.

2. Adafruit Unified Sensor Library

This library is required for the DHT sensor library.

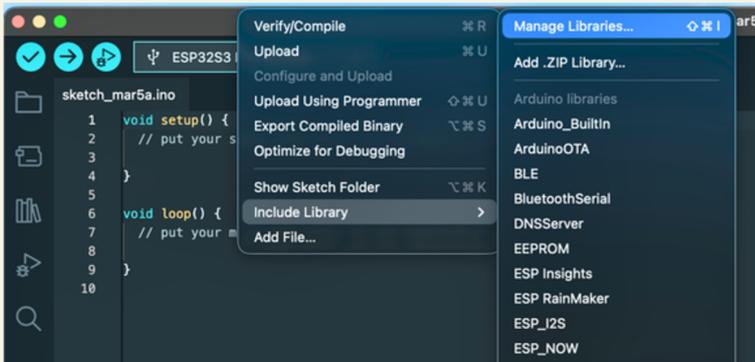
Steps:

- In Library Manager, search for Adafruit Unified Sensor.
- Click Install.

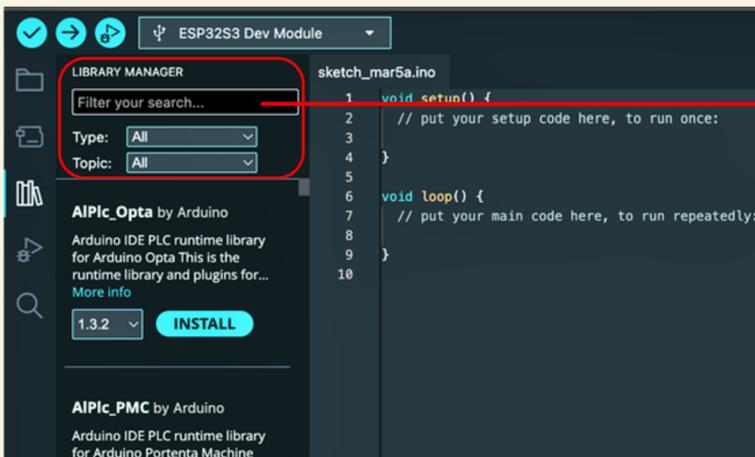
3. Servo Library

Used for the servo motor automatic gate project.

Note: The Servo library is already included in the Arduino IDE, so no additional installation is required.



Click Sketch → Include Library → Manage Libraries.



Search for libraries required as per programs

3. Connect Arduino Board

1. Connect the Arduino Uno to your computer using a USB cable.
2. Open the Arduino IDE.
3. Go to Tools → Board → Arduino Uno.
4. Go to Tools → Port and select the correct COM port connected to the Arduino board.

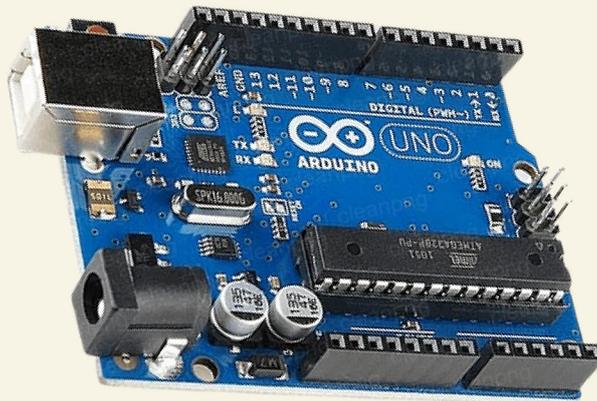
4. Uploading the Program

1. Open the Arduino IDE.
2. Copy the experiment code into the editor.
3. Select the correct Board and Port.
4. Click the Upload (→) button.
5. Wait until the message “Done Uploading” appears.

The program will start running on the Arduino board.

Components

i. Arduino Uno



The Arduino Uno is a popular microcontroller board based on the ATmega328P, widely used for learning electronics and embedded systems. It is easy to program using the Arduino IDE and can be connected to various sensors, motors, and display modules. The board can be powered either through a USB cable or an external power supply, making it suitable for both small projects and advanced applications.

The Arduino Uno consists of 14 digital input/output pins (0–13), which can be used to read digital signals or control devices like LEDs, buzzers, and relays. Among them, six pins (3, 5, 6, 9, 10, and 11) support PWM for analog-like output. It also has 6 analog input pins (A0–A5) that read analog signals from sensors with a 10-bit resolution (0–1023). The board includes power pins such as 5V, 3.3V, GND, and VIN, along with communication interfaces like UART (RX, TX), SPI, and I2C, allowing it to connect with various external modules and devices.

ii Buzzer



A buzzer module is a small electronic device used to produce sound signals such as beeps or alarms in electronic circuits. It works by converting electrical energy into sound energy and is commonly used in alert systems, timers, security devices, and embedded projects. There are two main types: active buzzers, which have a built-in oscillator and produce sound when power is applied, and passive buzzers, which require a frequency signal from a microcontroller like an Arduino or ESP32 to generate different tones. Buzzer modules typically have three pins—VCC, GND, and a signal pin—and are easy to interface, making them ideal for beginner-level electronics and STEM projects.

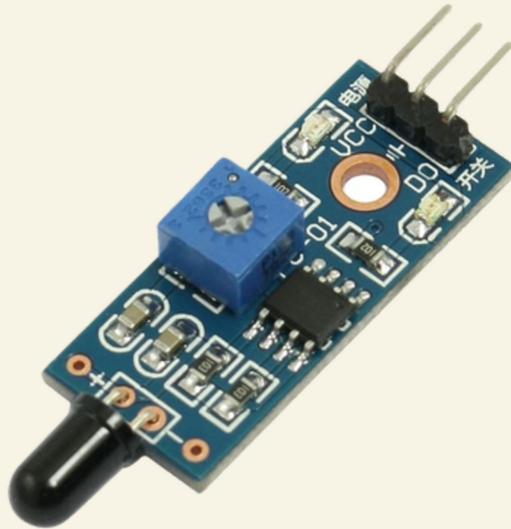
iii DHT11 sensor



The DHT11 is a digital sensor used to measure temperature and humidity. It contains a humidity sensing component and a thermistor for temperature measurement.

The DHT11 typically has 3 or 4 pins: VCC (5V) for power supply, DATA for digital output signal, and GND for ground (one pin may be unused in 4-pin versions). It can measure temperature from 0°C to 50°C and humidity from 20% to 90% RH with moderate accuracy. It is widely used in weather monitoring systems, home automation, and environmental monitoring projects.

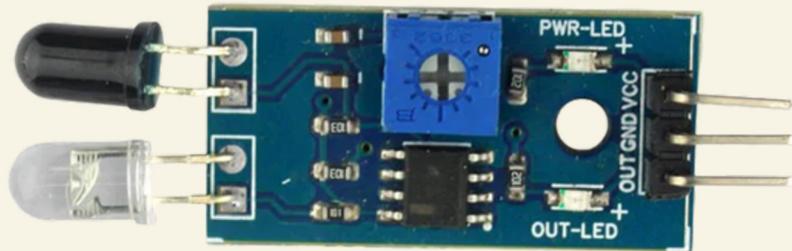
iv Flame IR sensor



A Flame IR Sensor is a device used to detect fire or flame by sensing infrared light emitted by flames. It is sensitive to wavelengths typically between 760 nm to 1100 nm, which are produced by fire. When a flame is detected, the sensor gives an output signal that can be used to trigger alarms, buzzers, or safety systems.

Most flame sensor modules (like KY-026) have three or four pins: VCC (5V) for power supply, GND for ground, DO (Digital Output) for flame detection signal, and sometimes AO (Analog Output) for intensity measurement. It is commonly used in fire detection systems, firefighting robots, and safety monitoring projects with Arduino.

v IR sensor



An IR (Infrared) sensor is an electronic device used to detect objects or obstacles by emitting and receiving infrared light. It consists of an IR transmitter (LED) and an IR receiver (photodiode). When an object comes in front of the sensor, the transmitted infrared light reflects back to the receiver, and the sensor detects the presence of the object.

IR sensors are commonly used in obstacle detection, line follower robots, automatic doors, and object counting systems.

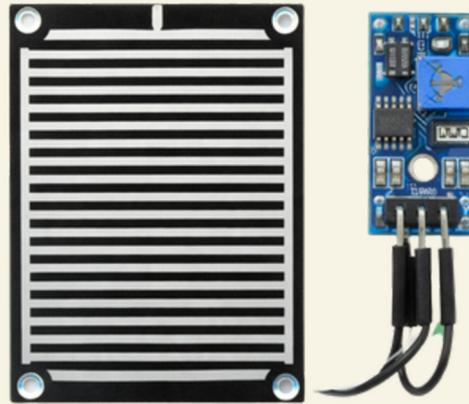
vi LED



An LED (Light Emitting Diode) is a semiconductor device that emits light when an electric current passes through it. It works on the principle of electroluminescence and converts electrical energy into light energy.

An LED has two terminals: anode (positive) and cathode (negative). The longer leg is the anode, and the shorter leg is the cathode. LEDs are commonly used in indicator lights, displays, traffic signals, and electronic projects with microcontrollers like Arduino. A current-limiting resistor is always used in series with an LED to prevent damage due to excess current.

vii Raindrop sensor



A Raindrop Sensor is an electronic device used to detect rain or water droplets. It works based on the principle of conductivity. When water falls on the sensing plate, it creates a conductive path between the tracks, changing the resistance and producing an output signal. It is commonly used in weather monitoring systems and automatic wiper control systems.

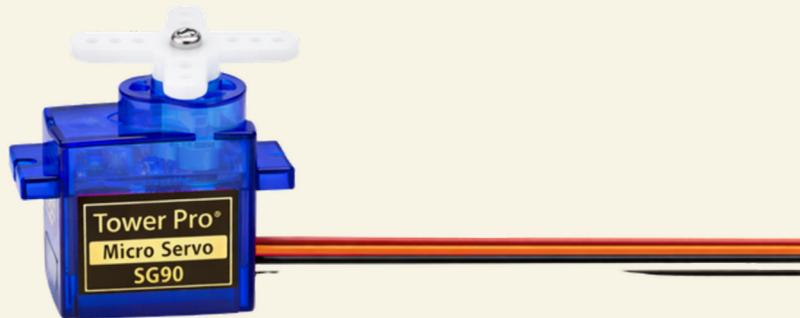
The typical rain sensor module consists of two parts: a rain detection plate and a control module (LM393 comparator). The module usually has four pins: VCC (5V) for power supply, GND for ground, DO (Digital Output) for rain detection signal, and AO (Analog Output) for measuring water intensity.

viii Resistor



A Resistor is a passive electronic component used to limit or control the flow of electric current in a circuit. It helps protect components like LEDs and microcontrollers from excess current. The value of a resistor is measured in ohms (Ω) and determines how much it resists the current.

ix Servo motor



A Servo Motor is a rotary actuator used for precise control of angular position. It can rotate to a specific angle (usually 0° to 180°) based on control signals from a microcontroller. Servo motors are widely used in robotics, automation systems, robotic arms, and remote-controlled vehicles.

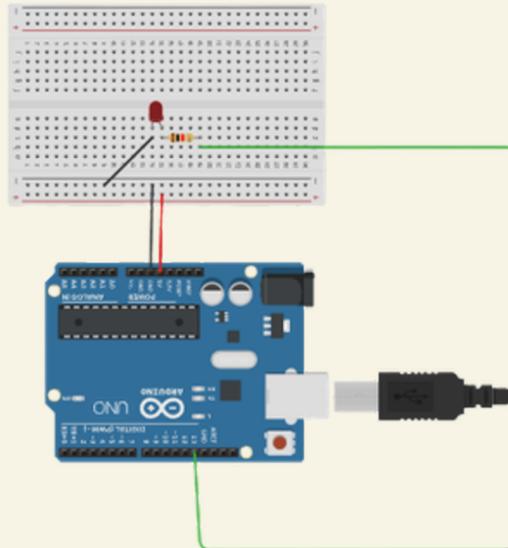
A typical micro servo (like SG90) has three wires: VCC (5V, Red) for power supply, GND (Brown/Black) for ground, and Signal (Orange/Yellow) for control. The signal wire receives PWM (Pulse Width Modulation) signals from a microcontroller like Arduino to set the desired angle.

x Ultrasonic sensor



An ultrasonic sensor module is a device used to measure distance by using high-frequency sound waves that are above the range of human hearing. It works by transmitting an ultrasonic pulse and then measuring the time taken for the echo to return after hitting an object. Based on this time delay, the distance to the object is calculated. One of the most commonly used modules in student projects is the HC-SR04, which operates at 40 kHz and can measure distances typically from 2 cm to 400 cm. It has four pins: VCC, GND, TRIG (to send the pulse), and ECHO (to receive the reflected signal). Ultrasonic sensors are widely used in obstacle detection, smart parking systems, water level monitoring, and robotics projects using boards like Arduino Uno or ESP32.

03 Smart LED Alert System



The LED Blink project is one of the most basic and important experiments in learning electronics and Arduino programming. It demonstrates how a microcontroller can control electronic components using simple code. In this project, an Arduino Uno is used to turn an LED ON and OFF at regular intervals, creating a blinking effect. This helps beginners understand digital output, circuit connections, and basic programming logic.

Connection Explanation :

- Connect the long leg (anode) of the LED to Arduino Pin 13 and the short leg (cathode) to one end of a 220 Ω resistor.
- Connect the other end of the resistor to GND of the Arduino Uno.
- The resistor limits the current flowing through the LED to prevent damage. When Pin 13 goes HIGH, the LED receives power and turns ON. When the pin goes LOW, the LED turns OFF.
- In the circuit wiring:
 - Red colour wire represents VCC (Power Supply)
 - Black colour wire represents Ground (GND)

Components Used :

- Arduino Uno
- Red LED
- Resistor (220 Ω)
- Breadboard

Code :

```
void setup()
{
  pinMode(13, OUTPUT); // Set pin 13 as output
}
```



```
void loop()
{
  digitalWrite(13, HIGH); // LED ON
  delay(1000);           // Wait for 1 second

  digitalWrite(13, LOW); // LED OFF
  delay(1000);           // Wait for 1 second
}
```

Results :

When the code is uploaded to the Arduino Uno, the LED connected to Pin 13 starts blinking. The LED turns ON for one second and then turns OFF for one second, continuously repeating this cycle.

Applications :

- Traffic Light Model – Used in school science exhibition projects to demonstrate traffic signals.
- Emergency Light Indicator – Shows warning or alert signals in simple systems.
- Simple Alarm System – LED can blink when a sensor detects motion or flame.
- Digital Timer Project – LED blinks at fixed intervals such as every second.

Obtained Results :

◆ ————— ◆

◆ ————— ◆

◆ ————— ◆

◆ ————— ◆

◆ ————— ◆

◆ ————— ◆

◆ ————— ◆

◆ ————— ◆

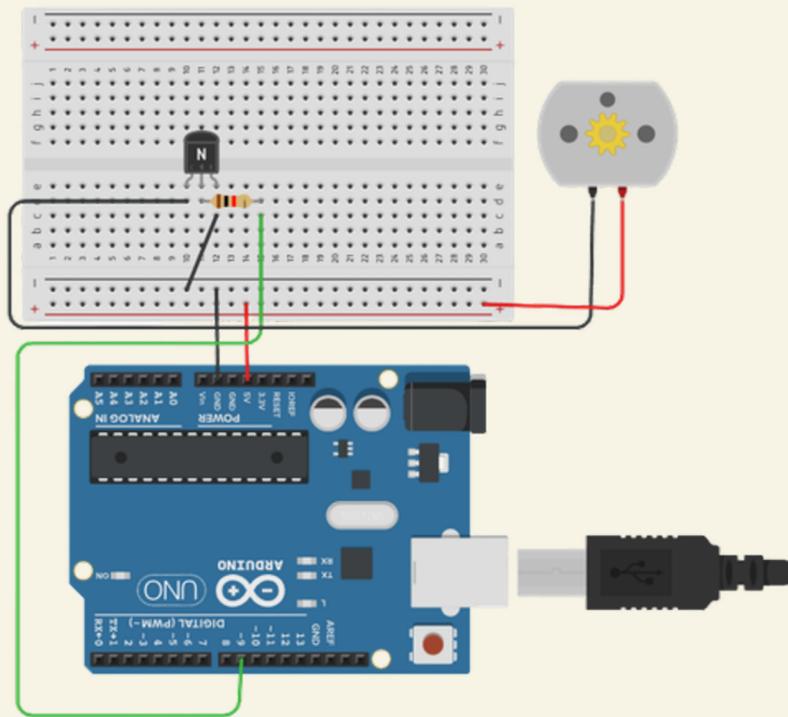
◆ ————— ◆

◆ ————— ◆

◆ ————— ◆



04 Precision Motion Control System



DC motors are widely used in electronics and robotics to convert electrical energy into mechanical motion. Since an Arduino pin cannot supply enough current to directly drive a motor, a transistor is used as a switch to control the motor safely. In this project, the Arduino sends a signal to an NPN transistor, which then allows current to flow through the motor. When the signal is HIGH, the motor rotates, and when it is LOW, the motor stops.

Connection Explanation :

- The positive terminal of the DC motor is connected to 5V of the Arduino.
- The negative terminal of the motor is connected to the collector of the NPN transistor.
- The emitter of the transistor is connected to GND.
- The base of the transistor is connected to Arduino Pin 9 through a 1k Ω resistor. The resistor limits the base current and protects the transistor and Arduino.
- When the Arduino sends a HIGH signal to Pin 9, the transistor switches ON and allows current to flow through the motor, causing it to rotate. When the pin goes LOW, the transistor switches OFF and the motor stops.

Components Used :

- DC Motor
- Arduino Uno
- NPN Transistor
- Resistor (1k Ω)

Code :

```
void setup() {  
  pinMode(9, OUTPUT); // Motor control pin  
}  
  
void loop() {  
  digitalWrite(9, HIGH); // Motor ON  
  delay(5000);           // Run for 5 seconds  
  
  digitalWrite(9, LOW); // Motor OFF  
  delay(2000);          // Stop for 2 seconds  
}
```

Results :

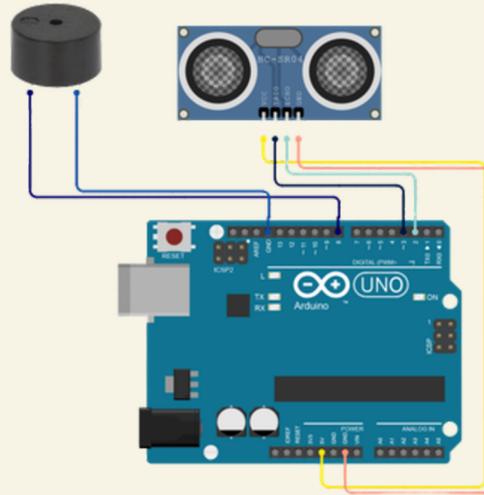
When the Arduino Pin 9 outputs HIGH, the transistor turns ON, allowing current to pass through the DC motor, which makes the motor rotate. When the pin outputs LOW, the transistor turns OFF, stopping the flow of current and the motor stops rotating.

Applications :

- Simple Robot Car – Used to drive wheels for robot movement.
- Mini Fan Model – Demonstrates motor rotation and airflow.
- Automatic Door Model – Opens and closes doors using motor control.
- Water Pump Model – Used in small irrigation or fountain demonstrations.
- Rotating Display Stand – Rotates models for exhibitions.
- Mini Agriculture Sprayer Model – Demonstrates a basic smart farming system.

Obtained Results :

05 Intelligent Obstacle Avoidance System



In modern smart systems, sensors help automate everyday tasks by detecting objects and responding automatically. One common example is vehicle detection using ultrasonic sensors, which is widely used in parking systems, obstacle detection, and safety alerts. In this project, an ultrasonic sensor is used to measure the distance of an approaching vehicle. When a vehicle comes close to the sensor, the Arduino Uno processes the signal and activates an LED and buzzer to indicate that the parking slot is occupied or that an object has been detected.

Connection Explanation :

- Connect the VCC and GND of the ultrasonic sensor to 5V and GND of the Arduino Uno. Connect the TRIG pin of the sensor to Pin 9 and the ECHO pin to Pin 10 of the Arduino.
- Connect the LED with a 220 Ω resistor to Pin 6 and GND. Connect the positive terminal of the buzzer to Pin 7 and the negative terminal to GND.
- When a vehicle or object is detected within a specific distance, the Arduino turns ON the LED and buzzer to provide a visual and audio alert.

Components Used :

- Ultrasonic Sensor
- Red LED
- Breadboard
- Buzzer

Code :

```
#define trigPin 9
#define echoPin 10
#define LED 6
#define buzzer 7
```



```
void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(LED, OUTPUT);
  pinMode(buzzer, OUTPUT);
}

void loop() {
  long duration;
  int distance;

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;

  if (distance < 10) { // Vehicle detected
    digitalWrite(LED, HIGH);
    digitalWrite(buzzer, HIGH);
  } else {
    digitalWrite(LED, LOW);
    digitalWrite(buzzer, LOW);
  }

  delay(300);
}
```

Results :

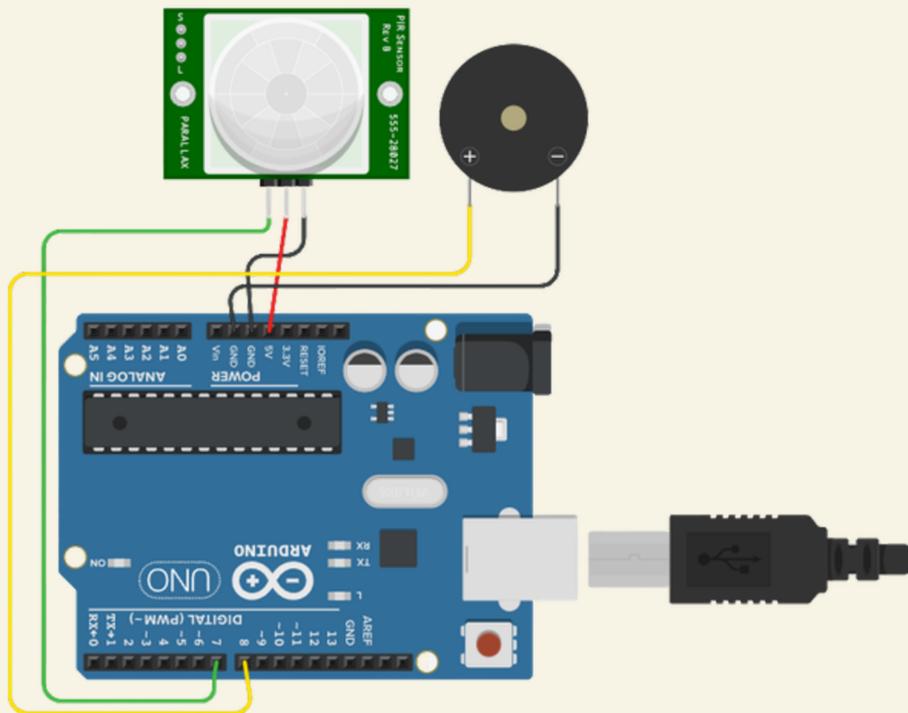
The ultrasonic sensor continuously measures the distance to detect whether a vehicle is present. If a vehicle comes closer than 10 cm, the Arduino turns ON the LED and activates the buzzer, indicating that the parking slot is occupied. If no vehicle is detected within that range, both the LED and buzzer remain OFF, indicating that the slot is empty.

Applications :

- Smart School Parking Model – Shows whether a parking slot is occupied or empty.



06 Intrusion Detection Door Security System



Motion detection is an important feature used in modern security and automation systems. A PIR (Passive Infrared) sensor detects movement by sensing changes in infrared radiation emitted by people or animals. In this project, the PIR sensor is connected to an Arduino Uno to detect motion. When movement is detected, the Arduino activates a buzzer to alert the user, creating a simple and effective security alarm system.

Connection Explanation :

- Connect the VCC and GND pins of the PIR sensor to 5V and GND of the Arduino Uno.
- Connect the OUT pin of the PIR sensor to digital Pin 7 of the Arduino.
- Connect the positive terminal of the buzzer to Pin 8 of the Arduino and the negative terminal to GND.
- When motion is detected, the PIR sensor sends a HIGH signal to the Arduino, and the Arduino turns ON the buzzer.

Components Used :

- Arduino Uno
- PIR Motion Sensor (HC-SR501)
- Buzzer
- Jumper Wires
- USB Cable / Power Supply

Code :

```
#define PIR 7
#define BUZZER 8

void setup() {
  pinMode(PIR, INPUT);
  pinMode(BUZZER, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int motion = digitalRead(PIR);

  if (motion == HIGH) {
    digitalWrite(BUZZER, HIGH); // Alarm ON
    Serial.println("Motion Detected!");
  }
  else {
    digitalWrite(BUZZER, LOW); // Alarm OFF
  }

  delay(500);
}
```

Results :

The PIR sensor detects changes in infrared radiation when a person or object moves within its range. When motion is detected, the sensor sends a HIGH signal to the Arduino. The Arduino then activates the buzzer, creating an alarm sound. When no motion is detected, the buzzer remains OFF.

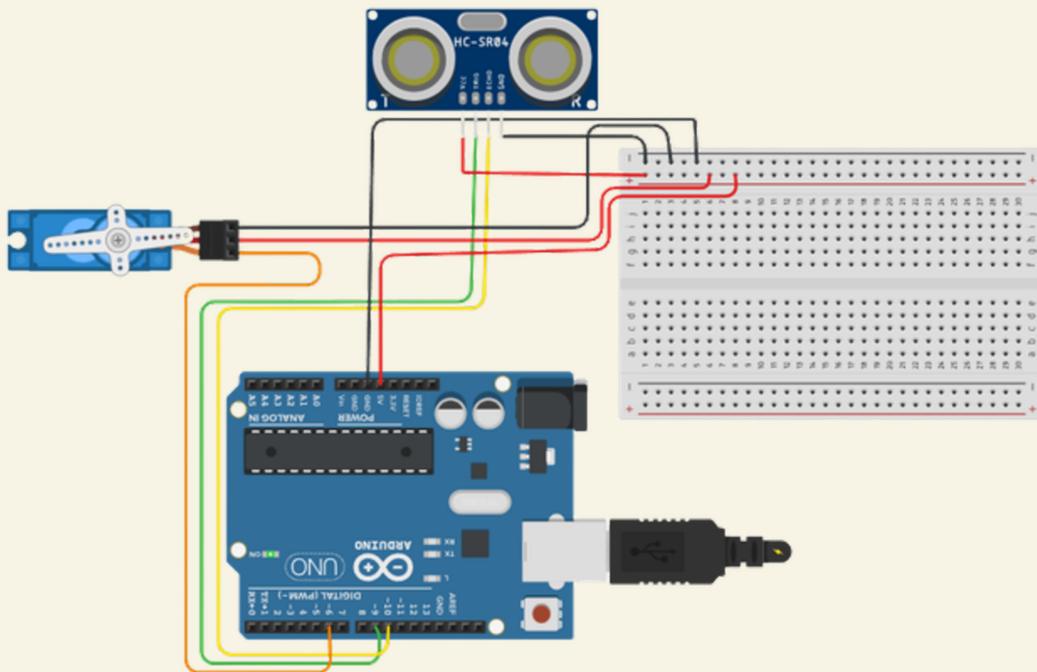
Applications :

- Classroom Security Alarm – Alerts when someone enters the classroom.
- Home Security Model – Detects unauthorized movement.
- Office Entry Alert System – Notifies when someone enters a room.
- Restricted Area Warning System – Provides alerts in sensitive areas.
- Basic Smart Security Project – Demonstrates motion detection in IoT and security systems.

Obtained Results :



07 Automated Toll Collection Barrier System



Automation systems are widely used to improve efficiency and reduce manual effort in everyday applications. One common example is an automatic gate system, where a gate opens automatically when a vehicle approaches. In this project, an ultrasonic sensor detects the distance of an approaching vehicle, and the Arduino Uno processes this data to control a servo motor. When the vehicle comes within a certain range, the servo motor rotates to open the gate, demonstrating a simple automated entry system.

Connection Explanation :

- The VCC and GND pins of the ultrasonic sensor are connected to 5V and GND of the Arduino Uno. The TRIG and ECHO pins of the sensor are connected to digital Pins 9 and 10 respectively to measure distance.
- The servo motor power wires are connected to the Arduino as follows:
 - Red wire → 5V
 - Brown/Black wire → GND
 - Signal wire → Pin 6
- When a vehicle comes close to the sensor, the ultrasonic sensor sends distance data to the Arduino, and the Arduino rotates the servo motor to open the gate.

Components Used :

- Arduino Uno
- Ultrasonic Sensor
- Servo Motor

Code :

```
#include <Servo.h>

#define trigPin 9
#define echoPin 10

Servo gateServo;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  gateServo.attach(6);
  gateServo.write(0); // Gate closed
}

void loop() {
  long duration;
  int distance;

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;

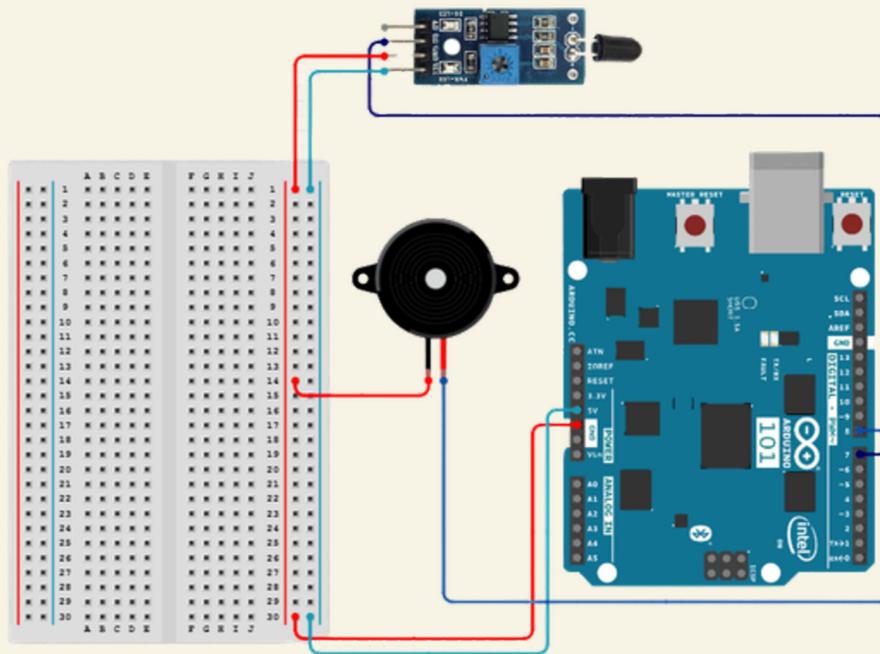
  if (distance < 15) { // Vehicle within 15 cm
    gateServo.write(90); // Open gate
    delay(3000);
    gateServo.write(0); // Close gate
  }

  delay(200);
}
```

Results :

The ultrasonic sensor continuously measures the distance of nearby objects. When a vehicle comes closer than 15 cm, the Arduino sends a signal to the servo motor to rotate to 90°, which opens the gate. After 3 seconds, the servo rotates back to 0°, closing the gate.

08 Fire Hazard Detection & Alert System



Fire detection systems are important safety solutions used in homes, industries, and public buildings. A flame sensor can detect fire by sensing infrared light emitted from flames. In this project, a flame sensor is connected to an Arduino Uno to detect the presence of fire. When the sensor detects a flame, the Arduino activates a buzzer to produce an alarm sound, alerting people nearby and demonstrating a basic fire detection system.

Connection Explanation :

- The VCC and GND pins of the flame sensor are connected to 5V and GND of the Arduino Uno. The DO (Digital Output) pin of the flame sensor is connected to Pin 7 of the Arduino to send fire detection signals.
- The positive terminal of the buzzer is connected to Pin 8 of the Arduino, and the negative terminal is connected to GND.
- When a flame is detected, the sensor sends a signal to the Arduino, and the Arduino activates the buzzer to produce an alarm sound.

Components Used :

- Flame Sensor Module (IR Type)
- Active Buzzer (5V)
- Arduino Uno
- Breadboard
- Jumper Wires
- USB Cable / Power Supply

Code :

```
#define flamePin 7
#define buzzer 8

void setup() {
  pinMode(flamePin, INPUT);
  pinMode(buzzer, OUTPUT);
}

void loop() {
  int flame = digitalRead(flamePin);

  if (flame == LOW) { // Flame detected
    digitalWrite(buzzer, HIGH);
  } else {
    digitalWrite(buzzer, LOW);
  }

  delay(200);
}
```

Results :

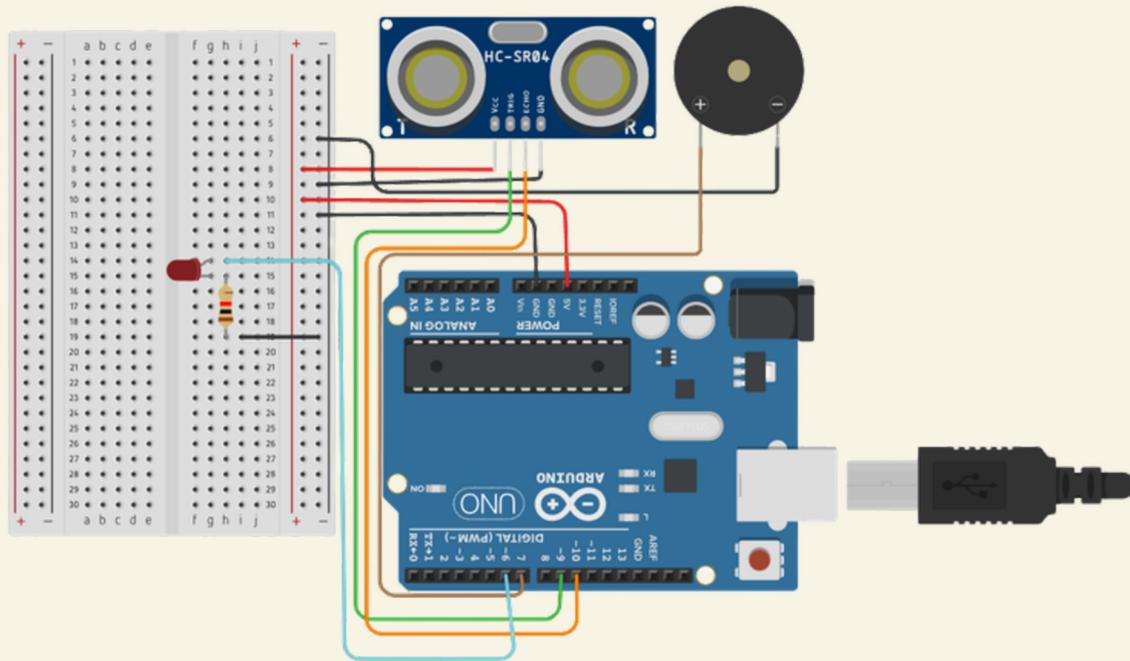
When a flame or fire source is brought near the flame sensor, it detects the infrared radiation emitted by the flame and sends a signal to the Arduino Uno. The Arduino then activates the buzzer, producing an alarm sound. When no flame is present, the buzzer remains OFF.

Applications :

- School Fire Safety Model – Demonstrates how fire alarm systems work.
- Home Fire Detection System – Provides early warning for houses.
- Office Safety System – Alerts when fire is detected in workplaces.
- Industrial Safety Model – Detects fire in factories or laboratories.
- Disaster Management Project – Used in emergency alert and safety systems.

Obtained Results :

09 IoT-Based Smart Parking Management System



Ultrasonic sensors are commonly used in smart systems to detect objects and measure distance without physical contact. They work by sending ultrasonic sound waves and calculating the time taken for the echo to return after hitting an object. In this project, an ultrasonic sensor is used with an Arduino Uno to detect the presence of a vehicle. When a vehicle comes close to the sensor, the Arduino activates an LED and a buzzer to indicate that the parking slot is occupied.

Connection Explanation :

- Connect the VCC and GND pins of the ultrasonic sensor to 5V and GND of the Arduino Uno. Connect the TRIG pin to Pin 9 and the ECHO pin to Pin 10 of the Arduino.
- Connect the LED with a 220Ω resistor to Pin 6 and GND. The resistor is used to limit the current and protect the LED.
- Connect the positive terminal of the buzzer to Pin 7 and the negative terminal to GND.
- When a vehicle or object comes within a certain distance, the Arduino turns ON both the LED and the buzzer to indicate detection.

Components Used :

- Ultrasonic Sensor
- Red LED
- Breadboard
- Buzzer

Code :

```
#define trigPin 9
#define echoPin 10
#define LED 6
#define buzzer 7

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(LED, OUTPUT);
  pinMode(buzzer, OUTPUT);
}

void loop() {
  long duration;
  int distance;

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;

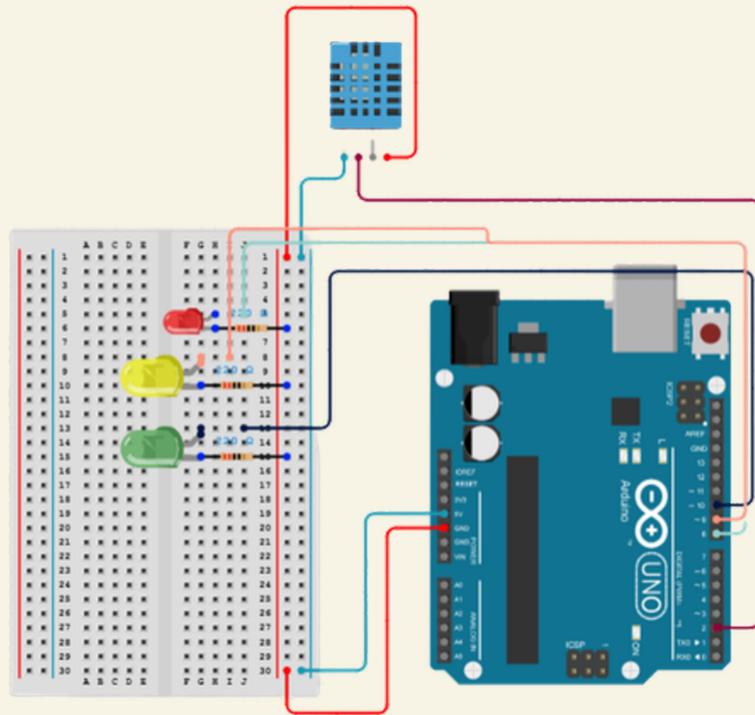
  if (distance < 10) { // Vehicle detected
    digitalWrite(LED, HIGH);
    digitalWrite(buzzer, HIGH);
  } else {
    digitalWrite(LED, LOW);
    digitalWrite(buzzer, LOW);
  }

  delay(300);
}
```

Results :

The ultrasonic sensor continuously measures the distance to detect whether a vehicle or object is present. If a vehicle comes closer than 10 cm, the Arduino turns ON the LED and activates the buzzer, indicating that the parking slot is occupied. If no vehicle is detected within that range, both the LED and buzzer remain OFF, indicating that the slot is empty.

10 Room Climate Monitor



Temperature monitoring plays an important role in environments such as homes, agriculture systems, and weather stations. The DHT11 sensor is a popular module that can measure both temperature and humidity. In this project, the DHT11 sensor is connected to an Arduino Uno to monitor temperature levels. Based on the measured temperature, different LEDs glow to indicate low, medium, or high temperature, providing a simple visual indication system.

Connection Explanation :

- The DHT11 sensor is connected by linking VCC to 5V, GND to GND, and the DATA pin to Pin 2 of the Arduino Uno. If a bare DHT11 sensor is used, a 10k Ω pull-up resistor is connected between VCC and DATA to ensure stable communication.
- Three LEDs are used to indicate temperature levels:
- Green LED \rightarrow Pin 8
- Yellow LED \rightarrow Pin 9
- Red LED \rightarrow Pin 10
- The cathode (short leg) of each LED is connected to GND through a 220 Ω resistor to limit the current and protect the LEDs.

Components Used :

- DHT11 Temperature & Humidity Sensor
- Red LED
- Yellow LED

- Green LED
- Three 220Ω Resistors (for LEDs)
- 10kΩ Resistor (if using bare DHT11 sensor)
- Breadboard
- Jumper Wires
- USB Cable / Power Supply

Code :

```
#include <DHT.h>
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

int greenLED = 8;
int yellowLED = 9;
int redLED = 10;

void setup() {
  dht.begin();
  pinMode(greenLED, OUTPUT);
  pinMode(yellowLED, OUTPUT);
  pinMode(redLED, OUTPUT);
}
void loop() {
  float temp = dht.readTemperature();

  if (temp < 25) {
    digitalWrite(greenLED, HIGH);
    digitalWrite(yellowLED, LOW);
    digitalWrite(redLED, LOW);
  }
  else if (temp >= 25 && temp <= 35) {
    digitalWrite(greenLED, LOW);
    digitalWrite(yellowLED, HIGH);
    digitalWrite(redLED, LOW);
  }
  else {
    digitalWrite(greenLED, LOW);
    digitalWrite(yellowLED, LOW);
    digitalWrite(redLED, HIGH);
  }

  delay(2000);
}
```




STRIVICTA