FUTURE GURUKULS

A Step Towards Future

AI & ROBOTICS HAND BOOK

Class 7

SUBROTO CHATTERJEE

(AI & ROBOTICS TRAINER)

FUTURE GURUKULS EDUTECH PRIVATE LIMITED.

IMPORTANT LINKS:

https://www.futuregurukuls.in
https://in.linkedin.com/company/futuregurukuls
https://www.instagram.com/futuregurukuls
https://www.youtube.com/@futuregurukuls
https://kitvit.in

ACKNOWLDGEMENT

I am deeply grateful and honored to present this *AI & Robotics Handbook* designed for students from Class 4 to 12. As a first-time author, this book represents not only my professional experience but also my passion for empowering young minds with practical knowledge in Artificial Intelligence, Robotics, IoT, and emerging technologies.

I sincerely thank **Future Gurukuls Edutech Private Limited** for providing me with the opportunity, platform, and continuous support to develop this comprehensive STEM learning resource. Their vision of delivering quality technical education to students has been a constant source of inspiration throughout this journey.

I also extend my heartfelt gratitude to my students, whose curiosity, enthusiasm, and innovative ideas motivated me to structure this handbook in a simple, practical, and activity-based format. Their questions and creativity helped shape the clarity and approach of this content.

Special appreciation goes to my mentors, colleagues, and well-wishers who encouraged me to take this step as a new author and guided me during the preparation of this handbook.

This book is a sincere effort to make AI and Robotics education accessible, practical, and engaging for school students. I hope it inspires learners to explore, experiment, and innovate in the field of technology.

Subroto Chatterjee
AI & Robotics Trainer
Future Gurukuls Edutech Private Limited

INTRODUCTION

IMPORTANT DEFINITION

(COMMON FOR ALL CLASS)

1- What is a Robot?

Robot is a smart machine which can take decision, it works like a human and makes our works easier.

2- What is IoT?

IoT Stands for *Internet Of Things.*

Connecting objects with internet and controlling it from anywhere around the world is called IoT.

3- What is Computer programming?

Set of instructions or command by which we can communicate with computer, machine or robot is called programming.
There are various types of programming languages like C, C++, JAVA, Python, Scratch, Arduino etc.

4- What is programming language?

A programming language is a vocabulary and set of grammatical rules for instructing a computer or computing device to perform specific tasks.

Example - C. C++, C#, Arduino, Java, Python, Scratch, Cobol etc.

5- What is Input Device?

Such device which collects data from outside and send it inside of any brain, are called input devices.
Ex- IR Sensor, LDR Sensor, Flame Sensor, Ultrasonic Sensor etc.
Basically, all types of sensors are input devices.

6- What are output devices?

Such devices which execute the final task according to the given input or give the final result, we called them output devices.
Ex- LED Module, Buzzer, DC Motor, Servo Motor etc.

7- What are Sensors?

Sensors are the sensing device ***which detect any change in environment*** and send signals to the microcontroller.

Ex- IR Sensor, LDR Sensor, Flame Sensor, Ultrasonic Sensor etc.

Basically, all types of sensors are input devices.

8- What are Actuators?

Actuators are the electronic device ***which makes any change in environment*** by signals from microcontroller.

Ex- LED Module, Buzzer, DC Motor, Servo Motor etc.

Basically, all types of sensors are Output devices.

9- What is Microcontroller?

A **Microcontroller** is a small computer on a single chip that is used to ***control devices and machines***. It has a **processor, memory, and input/output pins** inside one chip.

Ex- Arduino, ESP8266 / ESP32, Raspberry Pi etc.

10- What is AI?

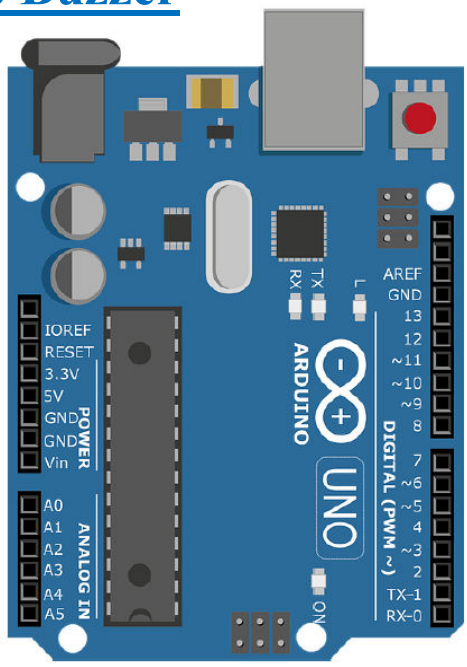
AI Stands for *Artificial Intelligence* Machines That Think

AI means *making machines smart* so they can *think and learn like humans*. It helps computers and robots to *learn from data and past experiences*. AI can see patterns, make decisions, and understand language.

ACTIVITIES FOR CLASS 7		
ACTIVITIES:	**CURRICULUM ACTIVITIES**	**PAGE**
ACTIVITY 1:	About Arduino, Digital & Analog Signals of Arduino – Voltage-Controlled LED & Buzzer	6
ACTIVITY 2:	Servo Motor Control – Smart Dustbin	10
ACTIVITY 3:	Robots in Motion – Programmable Car	13
ACTIVITY 4:	Obstacle Avoiding Robot	17
ACTIVITY 5:	IoT in Real Life – Smart Home System	22
ACTIVITIES:	**EXTRA ACTIVITIES**	**PAGE**
ACTIVITY 1:	Introduction to Arduino - LED Blinking	
ACTIVITY 2:	Multiple GPIO Control	
ACTIVITY 3:	Introduction to Sensor	
ACTIVITY 4:	Smart Train Gate	
ACTIVITY 5:	Robotic Human Hand	
ACTIVITY 6:	Human Following Suitcase	
ACTIVITY 7:	RFID-Based Door Lock System	

ACTIVITY 1

About Arduino, Digital & Analog Signals of Arduino – Voltage-Controlled LED & Buzzer



What is Arduino UNO?

Arduino uno is a programmable board which works like a artificial brain. We can call Arduino a computer for physical world. With the help of this board one can build 1000s of projects.

- **DC Power Jack:**
  Used to provide power to Arduino.
- **USB Jack:**
  Used to upload code and provide power supply.

Pins of Arduino UNO

Pin Name	Pin Number/Range	Total Pins
Digital Input/output Pin (GPIO PIN)	0 to 13 pin	14 pins
Analog Output Pin (PWM ~)	3,5,6,9,10,11 pin	6 pins
Analog Input Pin	A0 to A5 pin	6 pin
Power Pin	Vin, 5V, 3.3V	3 pin
Ground Pin	GND	3 pin

GPIO Stands for :
G - General.
P - Purpose.
I - Input.
O - Output.

PWM Stands for :
P - Pulse.
W - Width
M - Modulation

INPUT

- pinMode(pin, INPUT);
- pinMode(pin, INPUT_PULLUP);
- digitalRead(pin)
- Returns 0 or 1

Example:
Press and release pushbutton

DIGITAL

OUTPUT

- pinMode(pin, OUTPUT);
- digitalWrite(pin, 0 or 1)
- Write 0 or 1

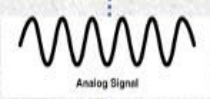
Example:
Turn the LED on and OFF

INPUT

- analogRead(pin)
- Returns 0 - 1023

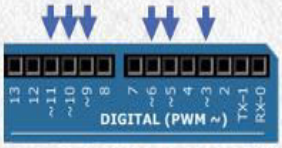
Example:
Rotate the rod of the potentiometer

ANALOG



OUTPUT

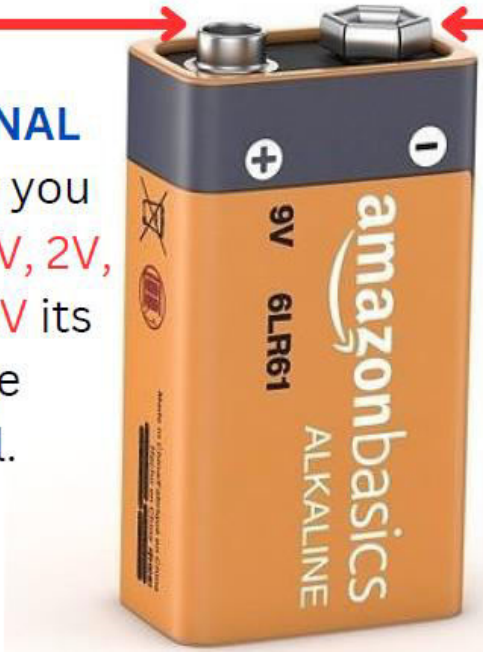
- analogWrite(pin, 0-255)
- Write 0 - 255



Example:
Fade the LED's brightness on and off

POSITIVE TERMINAL

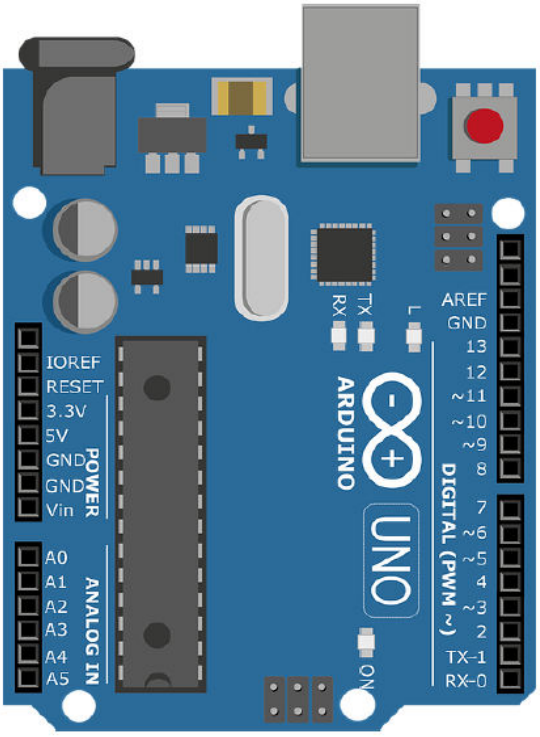
Note - Wherever you see +, Vin, Vcc, 1V, 2V, 3V, 4V, 5V..... NV its clarly indicate the Positive Terminal.



NEGATIVE TERMINAL

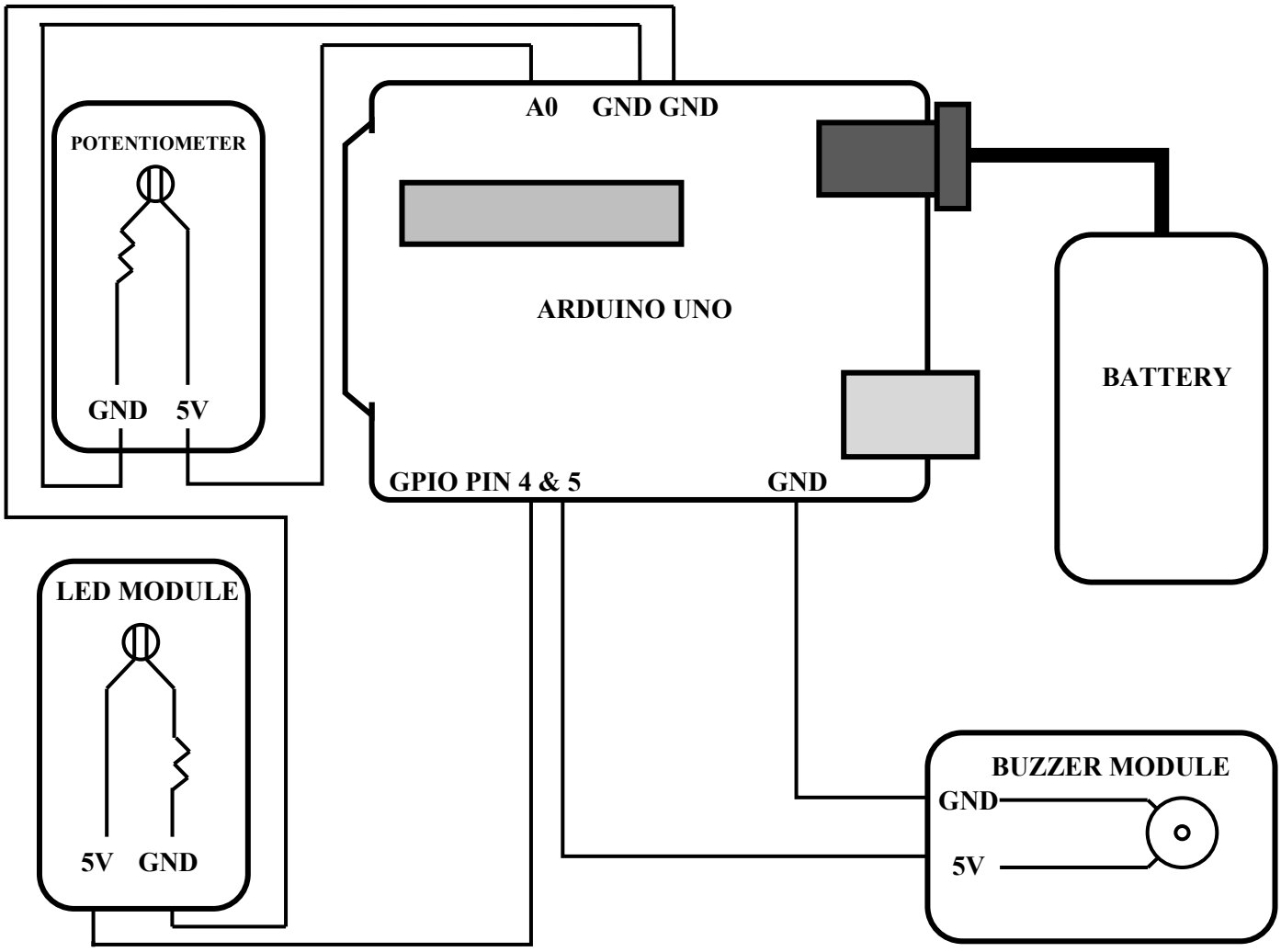
Note - Wherever you see -, GND, Ground its clarly indicate the Negative Terminal.

REQUIRED COMPONENTS:



1. Arduino Uno
2. LED Module
3. Buzzer Module
4. Potentiometer
5. 1-Pin Jumper Wires
6. Battery
7. Battery Cap

CONNECTION DIAGRAM OF VOLTAGE-CONTROLLED LED & BUZZER:



PROGRAM CODE OF VOLTAGE-CONTROLLED LED & BUZZER:

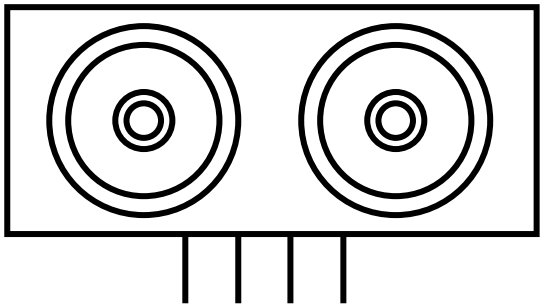
```
int potPin = A0;       // Input voltage control
int ledPin = 9;        // LED output
int buzzerPin = 8;     // Buzzer output

int value = 0;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
}

void loop() {
  value = analogRead(potPin);// Read input voltage (0-1023)

  int ledBrightness = map(value, 0, 1023, 0, 255);
  analogWrite(ledPin, ledBrightness); // Control LED
brightness

  if (value > 600) {   // Threshold voltage level
    digitalWrite(buzzerPin, HIGH); // Buzzer ON
  } else {
    digitalWrite(buzzerPin, LOW);  // Buzzer OFF
  }

  delay(50);
}
```

ACTIVITY 2

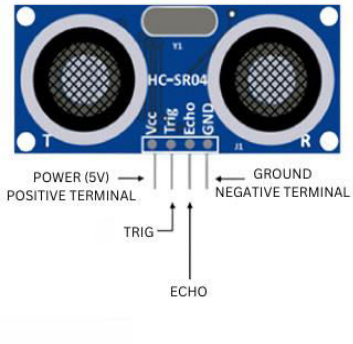
Servo Motor Control – Smart Dustbin

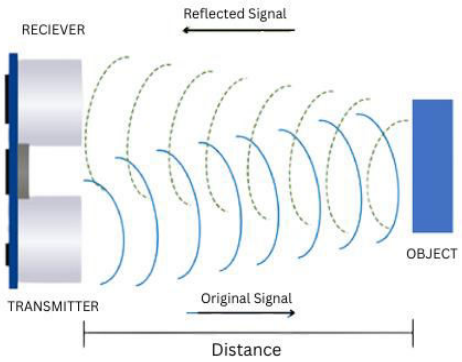
REQUIRED COMPONENTS:

1. Arduino Uno
2. Ultrasonic Sensor
3. Servo Motor
4. 1-Pin & 2-Pin Jumper Wires
5. Battery
6. Battery Cap



Ultrasonic Sensor





Microcontroller Interface Pins:

- **VCC:** Power supply (typically 5V).
- **GND:** Ground connection(Negative Terminal).
- **Trigger Pin:** Receives a signal from the microcontroller to start the transmission of ultrasonic sound waves.
- **Echo Pin:** Sends a signal back to the microcontroller when the echo is received.

Calculating Distance:

- The speed of sound in air is approximately 343 meters per second (0.0343 cm/µs).
- The division by 2 accounts for the round trip of the sound waves (to the object and back)

Formula:

$$Speed = \frac{Distance}{Time}$$

Where Speed is 0.0343 cm/µs and time is twice.

$$Distance = 0.0343 \times Time / 2$$

Servo Motor:

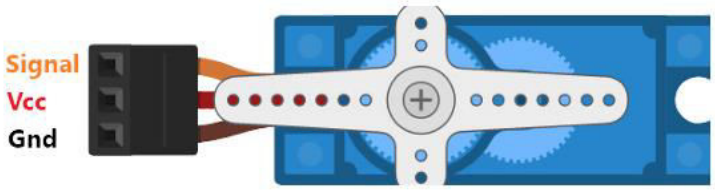
- Servo motors are handy devices primarily used to control angular or linear positions and specific velocities and accelerations.
- They work by receiving a pulse width modulated (PWM) signal to move to a desired position or speed.
- Typically, servo motors have three wires: power, ground, and the control signal.
- When activated, they rotate between fixed points, usually between 0° and 180°, relative to the motor's starting and ending positions.

Pins of Servo

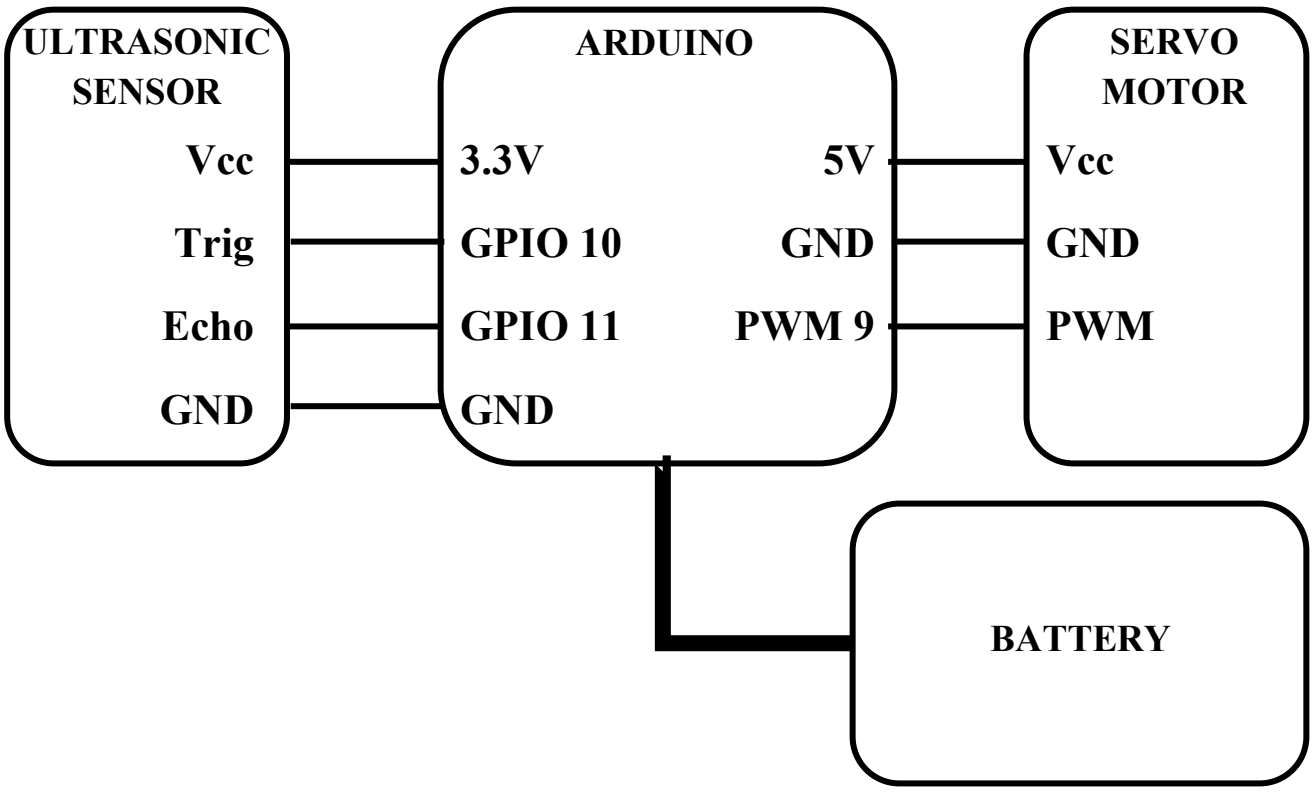
5V: Power supply (5V DC) (Red).

GND: Ground connection (Brown).

PWM Pulse Width Modulation (Orange) or SIGNAL.



CONNECTION DIAGRAM OF SMART DUSTBIN:



PROGRAM CODE OF SMART DUSTBIN

```
#include <Servo.h>

#define TRIG_PIN 10
#define ECHO_PIN 11
#define SERVO_PIN 9

Servo lidServo;

void setup() {
  Serial.begin(9600);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  lidServo.attach(SERVO_PIN);
  lidServo.write(0);  // Initially closed
}

void loop() {
  long duration, distance;
  // Send ultrasonic pulse
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  // Read echo time
  duration = pulseIn(ECHO_PIN, HIGH);
  distance = duration * 0.034 / 2;  // Convert to cm
  Serial.print("Distance: ");
  Serial.println(distance);
  if (distance < 20) { // If object is within 20 cm
    lidServo.write(90); // Open lid
    delay(5000);        // Keep lid open for 5 seconds
    lidServo.write(0);  // Close lid
  }
  delay(500); // Wait before next reading
}
```

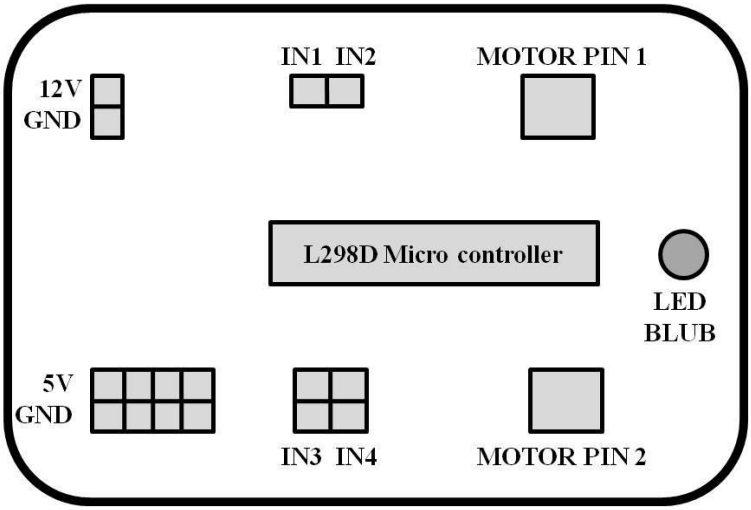
ACTIVITY 3

Robots in Motion – Programmable Car

REQUIRED COMPONENTS:

1. Car Frame
2. Wheels
3. Castor Wheel
4. Screw & Nuts
5. Geared Motors
6. Motor Driver L298D
7. Power Board
8. 0 & 1-PIN Jumper Wire
9. Rechargeable Battery
10. Arduino Uno

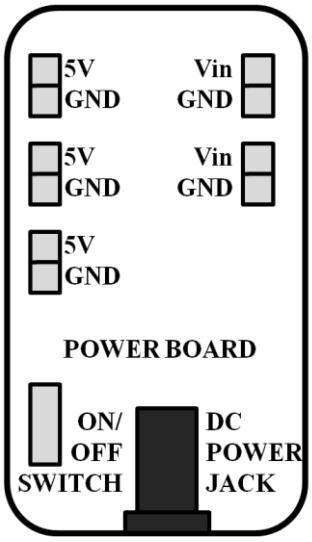
Diagram of Motor Driver



Motor Driver L298D:

A motor driver is an electronic circuit or module that controls the speed, direction, and other parameters of an electric motor.

Diagram of Power Board



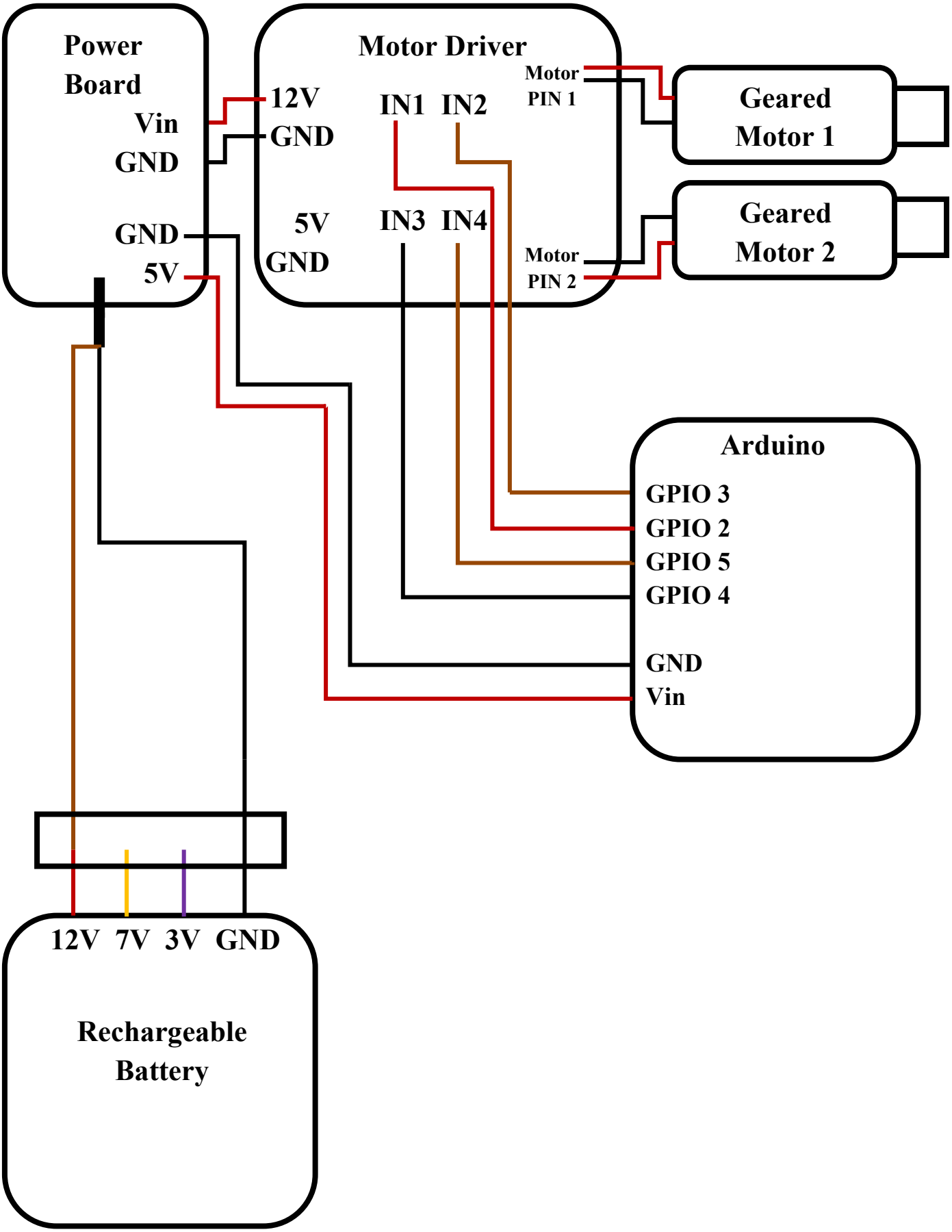
Power Board:

A power board is an electronic circuit or module that distributes the power supply to all components from battery. It can handle maximum supply of 12V.

Power board has pins:

- Two: Vin pins
- Three: 5V pins
- Five: GND pins

CONNECTION DIAGRAM OF PROGRAMMABLE CAR:



PROGRAM CODE OF PROGRAMMABLE CAR

```
// Simple Programmable Car using L298N and Arduino
// Pins: IN1, IN2 (Left Motor) | IN3, IN4 (Right Motor)
const int IN1 = 2;
const int IN2 = 3;
const int IN3 = 4;
const int IN4 = 5;

void setup() {
  // Set all pins as OUTPUT
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  // Start stopped
  stopCar();
}

void loop() {
  // Example program sequence:
  forward(); delay(2000);   // Move forward 2s
  stopCar(); delay(500);
  backward(); delay(2000);  // Move backward 2s
  stopCar(); delay(500);
  turnLeft(); delay(1000);  // Turn left 1s
  stopCar(); delay(500);
  turnRight(); delay(1000); // Turn right 1s
  stopCar(); delay(1000);
}

// -------- Car Movement Functions --------
void forward() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}
```

```
void backward() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}

void turnLeft() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}

void turnRight() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}

void stopCar() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
}
```

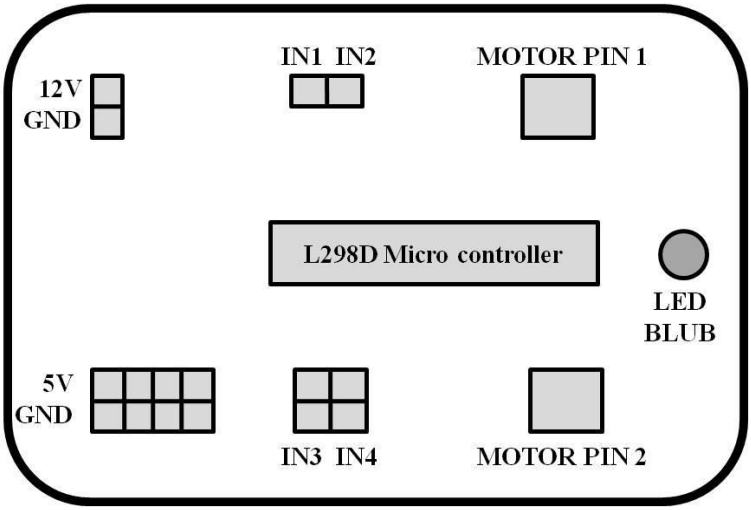
ACTIVITY 4

Obstacle Avoiding Robot

REQUIRED COMPONENTS:

1. Car Frame
2. Wheels
3. Castor Wheel
4. Screw & Nuts
5. Geared Motors
6. Motor Driver L298D
7. Power Board
8. 0 & 1-PIN Jumper Wire
9. Rechargeable Battery
10. Arduino Uno
11. Ultrasonic Sensor

Diagram of Motor Driver



Ultrasonic Sensor Calculating Distance:

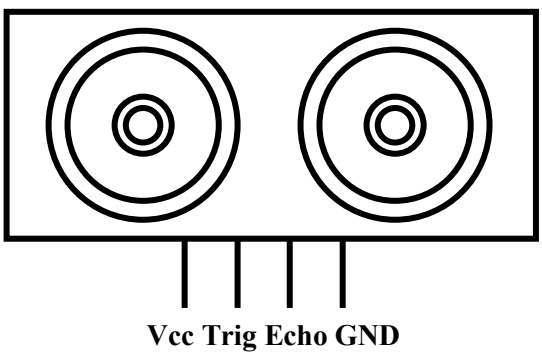
- The speed of sound in air is approximately 343 meters per second (0.0343 cm/µs).
- The division by 2 accounts for the round trip of the sound waves (to the object and back)

Formula:

$$Speed = \frac{Distance}{Time}$$

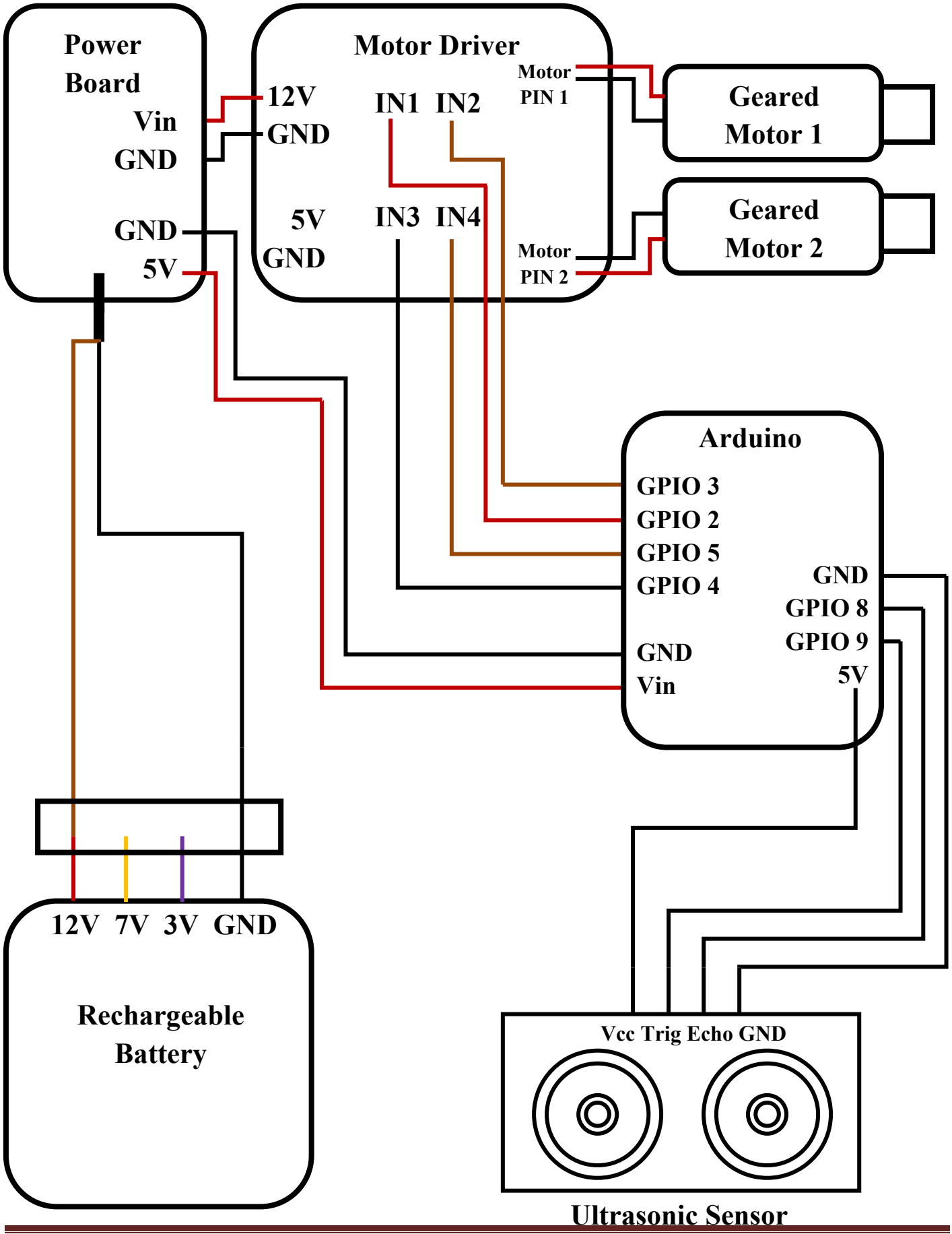
Where Speed is 0.0343 cm/µs and time is twice.

$$Distance = 0.0343 \times Time / 2$$



Ultrasonic Sensor

CONNECTION DIAGRAM OF PROGRAMMABLE CAR:



PROGRAM CODE OF PROGRAMMABLE CAR

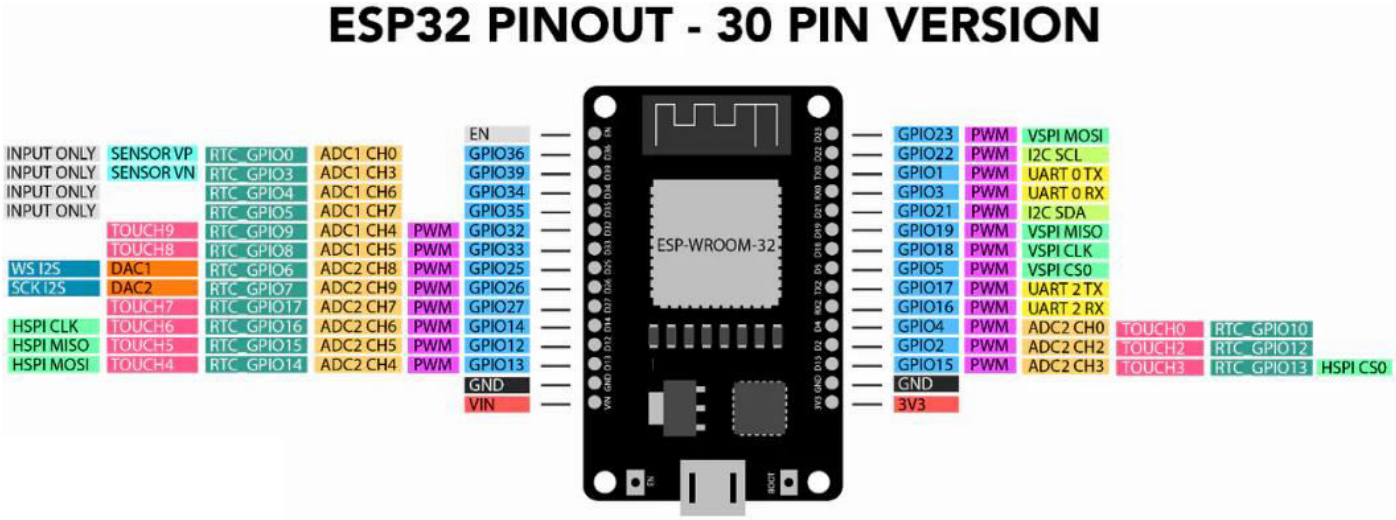
```
// === Pin Configuration ===
#define trigPin 9
#define echoPin 8

// Motor A (Left)
#define IN1 2
#define IN2 3

// Motor B (Right)
#define IN3 4
#define IN4 5
long duration;
int distance;
// === Functions ===
// Get distance from ultrasonic sensor
int getDistance() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

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

  duration = pulseIn(echoPin, HIGH, 20000); // timeout at
~20ms (~3.4m)

  int d = duration * 0.034 / 2;

  if (d == 0) {
    d = 100;  // If sensor returns 0, treat as 100
  }
  return d;
}
```

```
// Motor controls
void forward() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}

void backward() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}

void leftTurn() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}

void rightTurn() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}

void stopCar() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
}
```

```
// === Setup ===
void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  Serial.begin(9600);
}
// === Main Loop ===
void loop() {
  distance = getDistance();
  Serial.print("Distance: ");
  Serial.println(distance);
  if (distance > 30) {
    forward();  // No obstacle → move forward
  } else {
    stopCar();
    delay(200);
    backward(); // Go back a little
    delay(400);
    stopCar();
    delay(200);
    // Random turn left or right for smooth path change
    if (random(0, 2) == 0) {
      leftTurn();
    } else {
      rightTurn();
    }
    delay(500);
    stopCar();
    delay(200);
  }
}
```

ACTIVITY 5

IoT in Real Life – Smart Home System



ESP32 DEV MODULE – COMPLETE PIN DETAILS (TABULAR FORMAT)

Board Pin (Written on Board)	GPIO No. (Full Form)	Pin Name / Type	Main Function	Other Functions (If Any)	Important Notes / Details
VIN	—	Power Input	External power input (5V)	—	Can power ESP32 from 5V source (regulated on board to 3.3V).
GND	—	Ground	Common ground	—	Must be connected to ground of circuit.
3V3	—	Power Output	Provides 3.3V	—	Output from onboard regulator, max ~600mA (varies).
EN	—	Enable Pin (Enable Reset)	Chip Enable	Reset when pulled LOW	Must be HIGH for ESP32 to run.
VP (GPIO36)	GPIO36	Input Only	ADC1 Channel 0	Hall Sensor Input	Cannot be used as output.
VN (GPIO39)	GPIO39	Input Only	ADC1 Channel 3	Hall Sensor Input	Input only, no pull-up.
D34 (GPIO34)	GPIO34	Input Only	Digital Input	ADC1 Channel 6	Cannot output signal.
D35 (GPIO35)	GPIO35	Input Only	Digital Input	ADC1 Channel 7	Input only.

Board Pin (Written on Board)	GPIO No. (Full Form)	Pin Name / Type	Main Function	Other Functions (If Any)	Important Notes / Details
D32 (GPIO32)	GPIO32	I/O Pin	Digital I/O	Touch Sensor T9, ADC1 Ch 4	Safe for most uses.
D33 (GPIO33)	GPIO33	I/O Pin	Digital I/O	Touch Sensor T8, ADC1 Ch 5	Can be used for LED, sensor, etc.
D25 (GPIO25)	GPIO25	I/O Pin	Digital I/O	DAC1 (Digital to Analog Converter)	Can generate analog voltage.
D26 (GPIO26)	GPIO26	I/O Pin	Digital I/O	DAC2, ADC2 Ch 9	Good for audio output.
D27 (GPIO27)	GPIO27	I/O Pin	Digital I/O	Touch Sensor T7, ADC2 Ch 7	Can be used normally.
D14 (GPIO14)	GPIO14	I/O Pin	Digital I/O	Touch T6, ADC2 Ch 6	Also supports SPI clock in some modes.
D12 (GPIO12)	GPIO12	I/O Pin	Digital I/O	Touch T5, ADC2 Ch 5	⚠ Boot-strapping pin (avoid critical use).
D13 (GPIO13)	GPIO13	I/O Pin	Digital I/O	Touch T4, ADC2 Ch 4	Normal use OK.
D9 (GPIO9)	GPIO9	Flash SPI (Do not use)	Connected to internal Flash	—	❌ Do NOT use.
D10 (GPIO10)	GPIO10	Flash SPI (Do not use)	Connected to internal Flash	—	❌ Do NOT use.
D11 (GPIO11)	GPIO11	Flash SPI (Do not use)	Connected to internal Flash	—	❌ Do NOT use.
D8 (GPIO8)	GPIO8	Flash SPI (Do not use)	Connected to internal Flash	—	❌ Do NOT use.
D7 (GPIO7)	GPIO7	Flash SPI (Do not use)	Connected to internal Flash	—	❌ Do NOT use.
D6 (GPIO6)	GPIO6	Flash SPI (Do not use)	Connected to internal Flash	—	❌ Do NOT use.

Board Pin (Written on Board)	GPIO No. (Full Form)	Pin Name / Type	Main Function	Other Functions (If Any)	Important Notes / Details
D5 (GPIO5)	GPIO5	I/O Pin	Digital I/O	VSPI CS (Chip Select)	Safe pin for LED.
D4 (GPIO4)	GPIO4	I/O Pin	Digital I/O	Touch T0, ADC2 Ch 0	Commonly used for sensors.
D0 (GPIO0)	GPIO0	Boot Button / I/O	Must be LOW for flashing	Touch T1, ADC2 Ch 1	Press BOOT while uploading code.
D2 (GPIO2)	GPIO2	I/O + Onboard LED	Controls onboard LED	Touch T2, ADC2 Ch 2	LED is active LOW.
D15 (GPIO15)	GPIO15	I/O Pin	Digital I/O	Touch T3, ADC2 Ch 3	Boot-strapping pin (be careful).
RX2 (GPIO16)	GPIO16	UART2 RX	Serial Receive	—	Used for Serial2 communication.
TX2 (GPIO17)	GPIO17	UART2 TX	Serial Transmit	—	Used for Serial2 communication.
RX0 (GPIO3)	GPIO3	UART0 RX	USB Serial Receive	—	Used by Serial Monitor.
TX0 (GPIO1)	GPIO1	UART0 TX	USB Serial Transmit	—	Used by Serial Monitor.
D22 (GPIO22)	GPIO22	I/O Pin	I2C SCL (Clock)	General I/O	Common for I2C sensors.
D21 (GPIO21)	GPIO21	I/O Pin	I2C SDA (Data)	General I/O	Common for I2C sensors.
D19 (GPIO19)	GPIO19	I/O Pin	VSPI MISO	General I/O	SPI communication.
D18 (GPIO18)	GPIO18	I/O Pin	VSPI SCK (Clock)	General I/O	SPI communication.
D23 (GPIO23)	GPIO23	I/O Pin	VSPI MOSI	General I/O	SPI communication.

Important Terms & Their Full Forms (Easy Explanation)

Term	Full Form	Meaning (Simple)
GPIO	General Purpose Input Output	Pins that can act as input or output.
ADC	Analog to Digital Converter	Converts analog voltage to digital value.
DAC	Digital to Analog Converter	Converts digital signal to analog voltage.
SPI	Serial Peripheral Interface	High-speed communication protocol.
I2C	Inter-Integrated Circuit	Two-wire communication (SDA, SCL).
UART	Universal Asynchronous Receiver Transmitter	Serial communication (TX, RX).
VIN	Voltage Input	External power supply input.
GND	Ground	0V reference.
EN	Enable	Turns ESP32 ON/OFF.
BOOT	Boot Mode	Used while uploading code.

Best Pins to Use in Projects (Recommended)

Use these safely:

- D2, D4, D5, D13, D14, D18, D19, D21, D22, D23, D25, D26, D27, D32, D33

Avoid using (if possible):

- GPIO0, GPIO2, GPIO12, GPIO15 (Boot-related pins)
- GPIO6–GPIO11 (Flash pins)

ESP32 – Explanation (with Full Form)

Full Form of ESP32:

Espressif Systems Processor 32-bit

(It is commonly just called **ESP32**.)

What is ESP32?

ESP32 is a **microcontroller + Wi-Fi + Bluetooth module** made by **Espressif Systems (a company in China).**

It is used in:

- IoT (Internet of Things) projects
- Robotics
- Home automation
- Smart devices
- Wireless communication projects

Main Features of ESP32

Feature	Explanation
Processor	Dual-core 32-bit CPU (faster than Arduino)
Clock Speed	Up to 240 MHz
Wi-Fi	Yes (Built-in)
Bluetooth	Yes (Classic + BLE)
GPIO Pins	Around 30
ADC Pins	Many analog input pins
DAC Pins	2 (for audio, etc.)
Uses	IoT, automation, robotics, smart systems
Programming	Arduino IDE, ESP-IDF, MicroPython

Why is ESP32 Powerful?

Because it has:

- More speed than Arduino
- Built-in Wi-Fi
- Built-in Bluetooth
- More memory
- More pins and features

NodeMCU

Full Form of NodeMCU:

Node MicroController Unit

What is NodeMCU?

NodeMCU is **not a microcontroller itself** — it is a **development board** that uses the **ESP8266 chip**.

So **NodeMCU = Board + ESP8266 chip + USB + voltage regulator**

It is mainly used for **Wi-Fi based IoT projects.**

Main Features of NodeMCU

Feature	Explanation
Microcontroller Chip	ESP8266
Wi-Fi	Yes (Built-in)
Bluetooth	❌ No
GPIO Pins	About 15
ADC Pins	Only 1
Speed	Slower than ESP32
Programming	Arduino IDE, Lua
Uses	Basic IoT projects

What is an MCU? (Simple Definition)

An **MCU (Microcontroller Unit)** is a **small computer on a single chip** that is designed to control electronic devices.

It contains:

- Processor (CPU)
- Memory (RAM + Flash)
- Input/Output (I/O) pins all inside **one single integrated circuit (IC).**

ESP32 vs NodeMCU

Feature	ESP32	NodeMCU (ESP8266)
Company	Espressif	Espressif
Wi-Fi	✅ Yes	✅ Yes
Bluetooth	✅ Yes	❌ No
Processor	Dual-core	Single-core
Speed	Faster	Slower
GPIO Pins	More (~30)	Less (~15)
ADC Pins	Many	Only 1
Best for	Advanced IoT	Simple IoT

5V Relay Module – Pin Explanation (Active LOW)

Most common 5V relay modules have **4 pins**:

Pin Written on Board	Full Form / Meaning	Type	Function	Explanation (Easy)
VCC	Voltage Common Collector	Power Input	Supplies 5V to the relay module	Connect to **5V** from Arduino/ESP32/NodeMCU power supply. This powers the relay coil and circuit.
GND	Ground	Power Ground	Completes the power circuit	Connect to **GND** of Arduino/ESP32/NodeMCU (common ground is very important).
IN	Input	Control Signal	Controls ON/OFF of relay	This pin receives signal from microcontroller (Arduino/ESP32). In **Active LOW**, relay turns **ON when IN = LOW (0V)**.
COM	Common	Relay Switch Terminal	Common contact of relay switch	This is the middle terminal of the relay. Your device's power line connects here.
NO	Normally Open	Relay Switch Terminal	Connected to COM only when relay is ON	Normally disconnected. Connects to COM when relay is triggered.
NC	Normally Closed	Relay Switch Terminal	Connected to COM when relay is OFF	Normally connected. Disconnects when relay is triggered.

What does Active LOW mean? (Important)

Active LOW = Relay turns ON when IN pin gets LOW (0V / GND).

Logic Table (Active LOW Relay)

IN Pin Signal	Relay Status
HIGH (5V / 3.3V)	❌ OFF
LOW (0V / GND)	✅ ON

How it works (Step-by-step)

When IN = HIGH (5V)

- Relay coil is **not energized**
- COM is connected to **NC**
- Device connected to NO remains **OFF**

When IN = LOW (0V)

- Relay coil gets energized
- Relay clicks **ON**
- COM connects to **NO**
- Device connected to NO turns **ON**

Simple Example (Bulb Control)

Wiring:

- Bulb live wire → COM
- Bulb other wire → NO
- Neutral directly to bulb

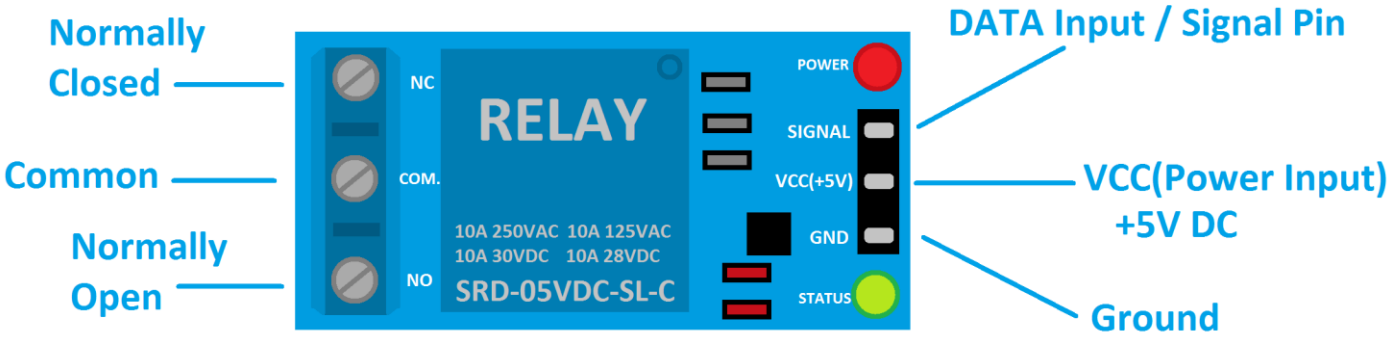
Operation:

- IN = HIGH → Bulb OFF
- IN = LOW → Bulb ON

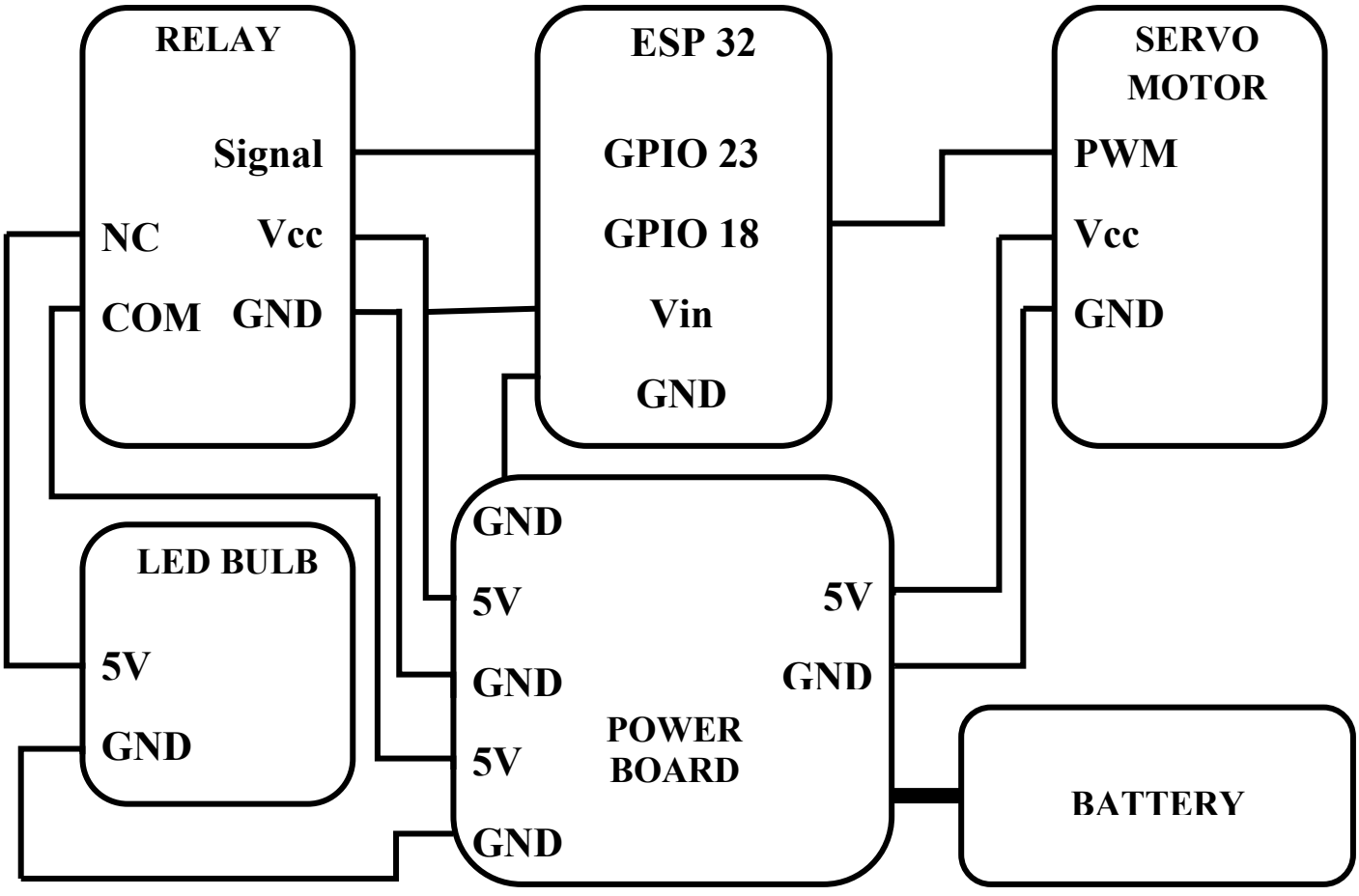
REQUIRED COMPONENTS:

1. ESP32
2. Relay Module
3. LED Bulb
4. Servo Motor
5. Power Board
6. 1-PIN Jumper Wire
7. Rechargeable Battery

Diagram of Relay Module



CONNECTION DIAGRAM OF SMART HOME SYSTEM



PROGRAM CODE OF SMART HOME SYSTEM

```
#include <WiFi.h>
#include <ESP32Servo.h>
const char* ssid = "H";
const char* password = "12345678";
WiFiServer server(80);
Servo gateServo;
#define relayPin 23   // Light relay
#define servoPin 18   // Servo motor pin
String header;
String lightState = "OFF";
int servoAngle = 0;
void setup() {
  Serial.begin(115200);
  pinMode(relayPin, OUTPUT);
  digitalWrite(relayPin, LOW); // Light OFF initially
  gateServo.attach(servoPin);
  gateServo.write(0); // Gate closed position
  Serial.print("Connecting to WiFi...");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nWiFi Connected!");
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP());
  server.begin();
}
```

```
void loop() {
  WiFiClient client = server.available();

  if (client) {
    String request = client.readStringUntil('\r');
    client.flush();

    if (request.indexOf("/light/on") != -1) {
      digitalWrite(relayPin, HIGH);
      lightState = "ON";
    }
    if (request.indexOf("/light/off") != -1) {
      digitalWrite(relayPin, LOW);
      lightState = "OFF";
    }

    if (request.indexOf("/gate/open") != -1) {
      gateServo.write(90); // Open gate
      servoAngle = 90;
    }
    if (request.indexOf("/gate/close") != -1) {
      gateServo.write(0); // Close gate
      servoAngle = 0;
    }

    // Web Page
    client.println("HTTP/1.1 200 OK");
    client.println("Content-type:text/html");
    client.println("Connection: close");
    client.println();
```

```
    client.println("<!DOCTYPE html><html>");
    client.println("<head><title>ESP32 Home
Automation</title></head>");
    client.println("<body style='text-align:center;'>");
    client.println("<h1>ESP32 Smart Home</h1>");

    client.println("<h2>Light Control</h2>");
    client.println("<a href='/light/on'><button>Light
ON</button></a>");
    client.println("<a href='/light/off'><button>Light
OFF</button></a>");
    client.println("<p>Light Status: " + lightState +
"</p>");

    client.println("<h2>Gate Control</h2>");
    client.println("<a href='/gate/open'><button>Open
Gate</button></a>");
    client.println("<a href='/gate/close'><button>Close
Gate</button></a>");
    client.println("<p>Gate Position: " +
String(servoAngle) + "°</p>");

    client.println("</body></html>");

    client.stop();
  }
}
```