FUTURE GURUKULS

A Step Towards Future

AI & ROBOTICS HAND BOOK

Class 6

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?

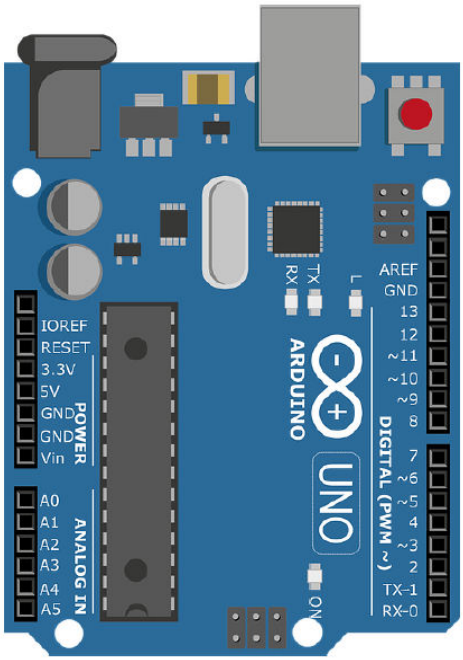
AI Stands for *Artificial Intelligence* Machines That Think

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 6

ACTIVITY 1

Introduction to Arduino – The Brain of Smart Projects



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

Arduino Syntax/Function

- **pinMode();**
- **Serial.begin();**
- **Serial.print();**
- **Serial.println();**
- **delay();**
- **digitalWrite();**
- **analogWrite();**
- **digitalRead();**
- **analogRead();**
- **delayMicroseconds();**

pinMode();

For deciding a particular pin whether it will behave as input or output.

Syntax:

pinMode(pinNumber, type);

Example:

pinMode(11, OUTPUT);
or
pinMode(A0, INPUT);

pinMode(5, INPUT);
or
pinMode(4, OUTPUT);

digitalWrite();

For making output pins at HIGH position or LOW position

Syntax:

digitalWrite(pinNumber, state);

Example:

digitalWrite(13, HIGH);
or
digitalWrite(13, LOW);

digitalWrite(7, 1);
or
digitalWrite(7, 0);

analogWrite();

For setting the voltage of an output pins.

Syntax:

```
analogWrite (5, any value between 0 to 255);
```

Example:

```
analogWrite(9, 128);
or
analogWrite(3, 64);
```

digitalRead();

Reading the input voltage of digital pins.

Syntax:

```
digitalRead(pinNumber);
```

Example:

```
digitalRead(6);
or
digitalRead(12);
```

analogRead();

Reading the input voltage of analog or PWM pins.

Syntax:

analogRead(pinNumber);

Example:

analogRead(A0);

or

analogRead(A5);

Serial.begin();

For communicating with Serial Monitor.

Sets up a way for your Arduino to send messages to your computer.
It needs to know how fast to talk.

Syntax:

Serial.begin(speed);

Example:

Serial.begin(9600);

Start talking to the computer at a speed of 9600 bits per second.
(This is a common speed)

Serial.print();

For printing any message or data in serial monitor.

Syntax:

Serial.print("whatToSay");

Example:

Serial.print("*Hello Future Gurukuls*");

Serial.println();

Same as Serial.print(), but after sending the message, it moves to a new line, like pressing "Enter" on your keyboard!

Syntax:

Serial.println("whatToSay");

Example:

Serial.println("*Hello Future Gurukuls*");

delay();

Makes the Arduino wait for a certain amount of time. The time is given in milliseconds (1000 milliseconds = 1 second).

Syntax:

```
delay(milliseconds);
```

Example:

```
delay(1000);
or
delay(500);
```

delayMicroseconds();

Makes the Arduino wait for a very short amount of time. The time is given in microseconds (1,000,000 microseconds = 1 second).

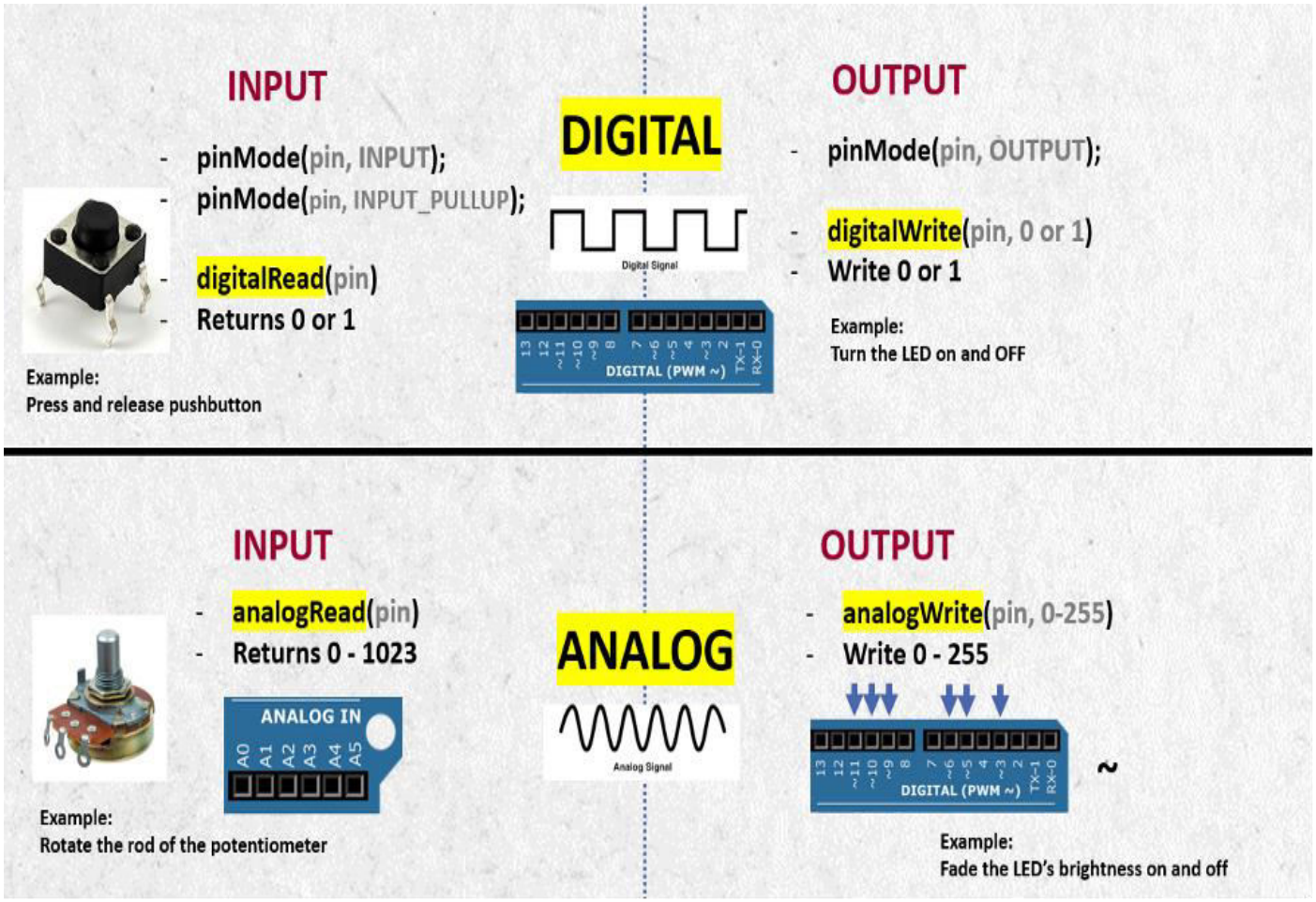
Syntax:

```
delayMicroseconds(microseconds);
```

Example:

```
delayMicroseconds(10);    or    delayMicroseconds(1000);
```

"Wait for 10 millionths of a second."
"Wait for 1 thousandth of a second." (Same as delay(1))


INPUT
- pinMode(pin, INPUT);
- pinMode(pin, INPUT_PULLUP);
- digitalRead(pin)
- Returns 0 or 1
Example:
Press and release pushbutton
DIGITAL
Digital Signal
DIGITAL (PWM ~)
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
ANALOG IN
A0 A1 A2 A3 A4 A5
Example:
Rotate the rod of the potentiometer
ANALOG
Analog Signal
OUTPUT
- analogWrite(pin, 0-255)
- Write 0 - 255
DIGITAL (PWM ~)
Example:
Fade the LED's brightness on and off

9V BATTERY


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.
9V
6LR61
amazonbasics
ALKALINE

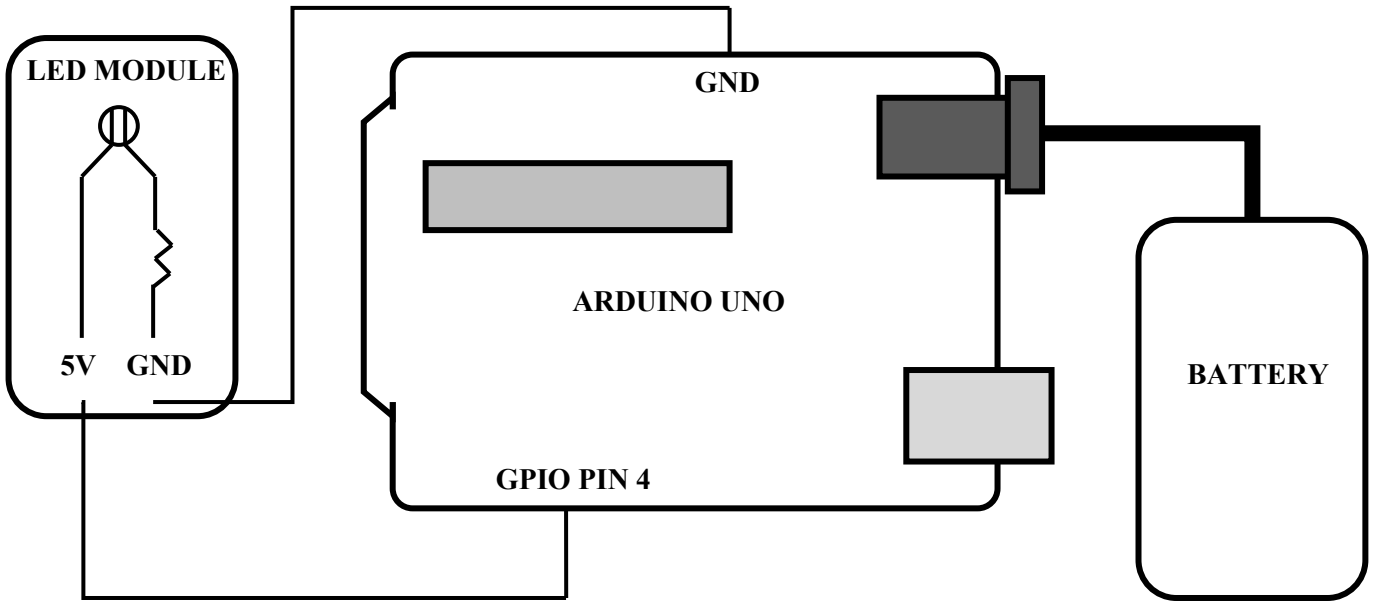
ACTIVITY 2

Getting Started with Arduino IDE – LED Blinking

REQUIRED COMPONENTS:

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

CONNECTION DIAGRAM OF LED BLINKING:



PROGRAM CODE OF LED BLINKING:

```
void setup() {
  pinMode(4,OUTPUT);
}
void loop() {
  digitalWrite(4,HIGH);
  delay(1000);
  digitalWrite(4,LOW);
  delay(1000);
}
```

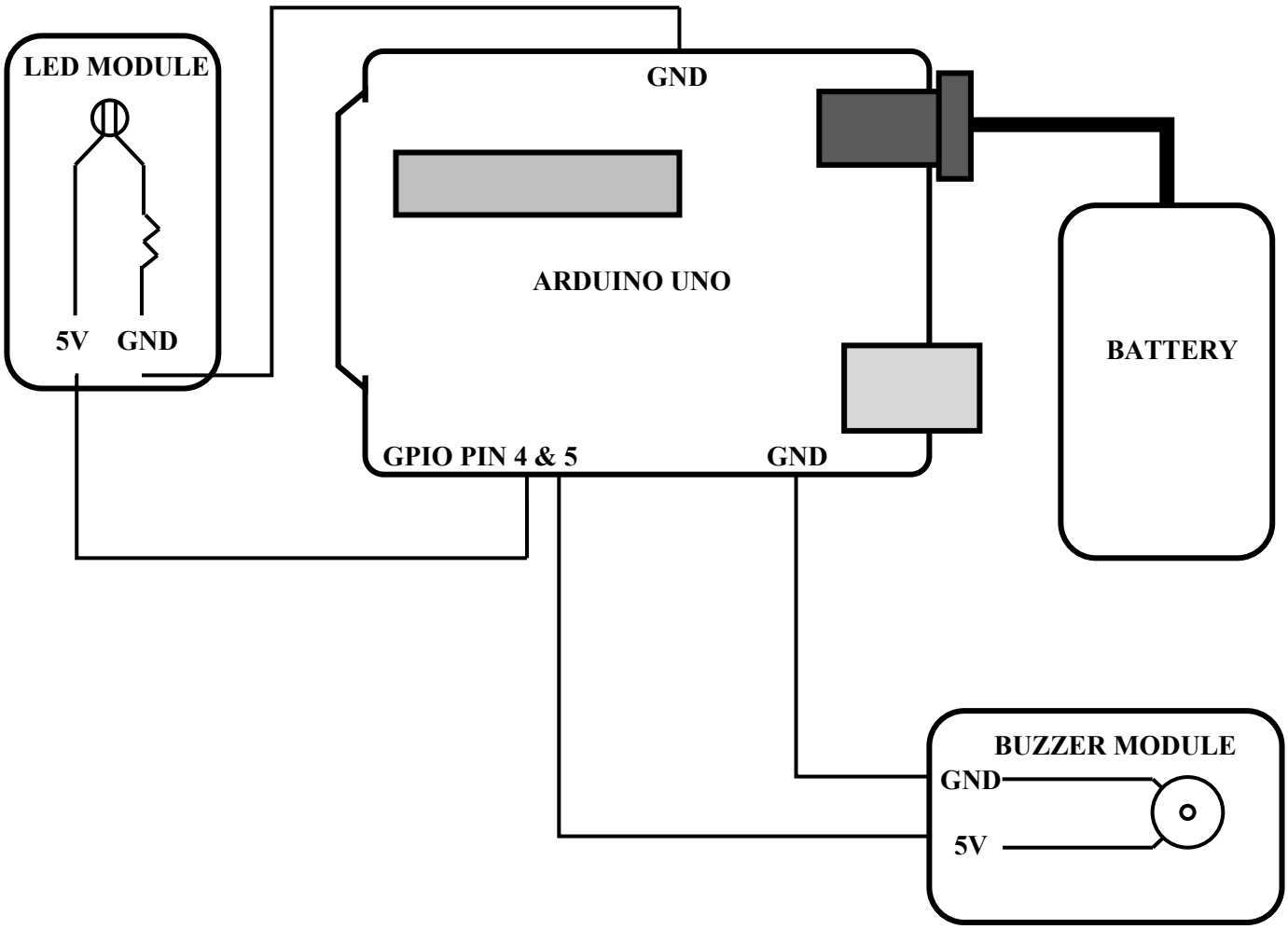
ACTIVITY 3

Multiple GPIO Control – LED & Buzzer Control

REQUIRED COMPONENTS:

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

CONNECTION DIAGRAM OF MULTIPLE GPIO:



PROGRAM CODE OF MULTIPLE GPIO:

```
void setup() {
  // put your setup code here, to run once:
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
}
void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(4,HIGH);
  digitalWrite(5,LOW);
  delay(1000);
  digitalWrite(4,LOW);
  digitalWrite(5,HIGH);
  delay(1000);
}
```

ACTIVITY 4

Introduction to Sensor – Soil Moisture Sensor

REQUIRED COMPONENTS:

1. Arduino Uno
2. Soil Moisture Sensor
3. Buzzer Module
4. Motor Driver L298D
5. Water Pump
6. Tube Pipe
7. 1-Pin Jumper Wires
8. Battery
9. Battery Cap

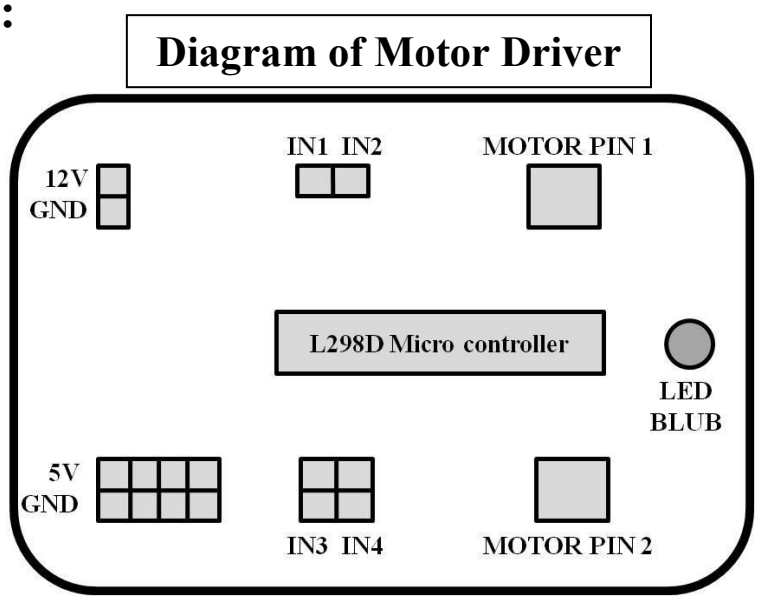
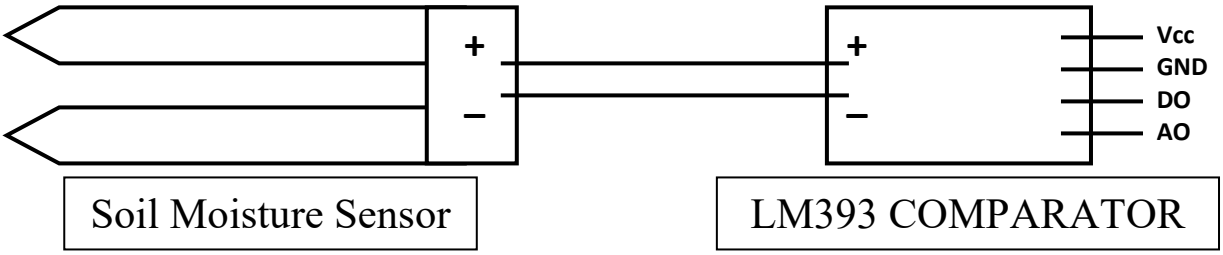
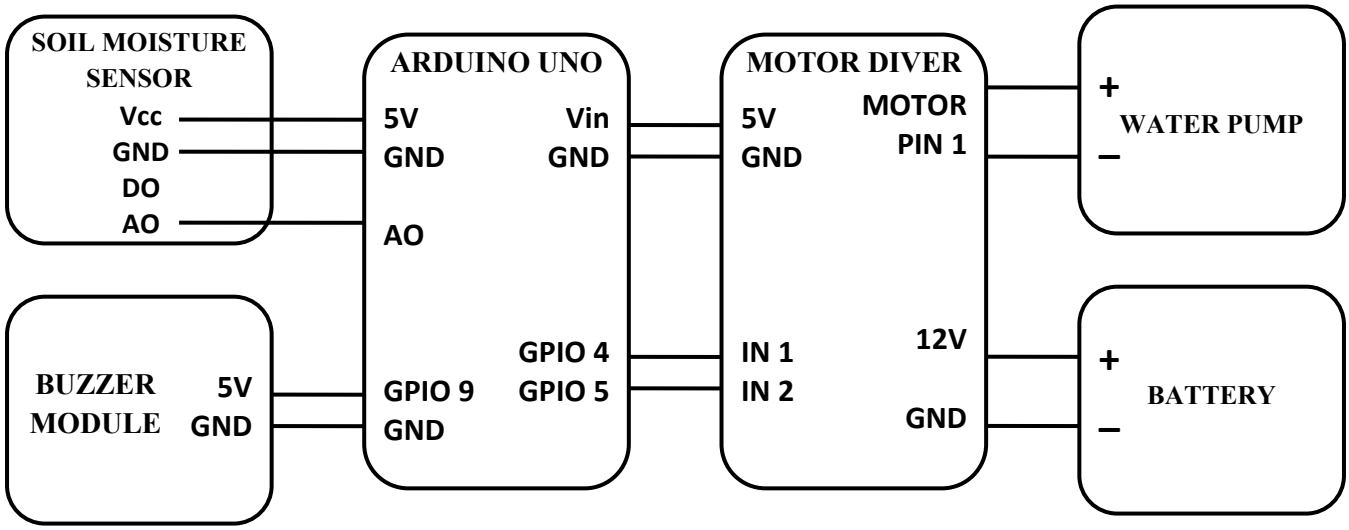


DIAGRAM OF SOIL MOISTURE SENSOR:



CONNECTION DIAGRAM OF SMART GARDENING:



PROGRAM CODE OF SMART GARDENING USING SOIL MOISTURE SENSOR

```
// Pin Definitions
const int sensorPin = A0; // Soil moisture sensor O/P
const int buzzerPin = 9; // Buzzer positive terminal
const int pumpPin1 = 4; // Motor Driver Input 1
const int pumpPin2 = 5; // Motor Driver Input 2
// Variables
int moistureLevel = 0;
int threshold = 600; // Adjust this based on your
soil/sensor (0-1023)

void setup() {
  pinMode(buzzerPin, OUTPUT);
  pinMode(pumpPin1, OUTPUT);
  pinMode(pumpPin2, OUTPUT);
  Serial.begin(9600); // Open serial monitor for
calibration
  digitalWrite(buzzerPin, LOW);
  stopPump();
}

void loop() {
  moistureLevel = analogRead(sensorPin);
  Serial.print("Moisture Level: ");
  Serial.println(moistureLevel);
  // If soil is dry (value is above threshold)
  if (moistureLevel > threshold) {
    Serial.println("Soil dry! Watering now...");
    alertUser();
    startPump();
    delay(3000); // Water for 3 seconds
  } else {
    Serial.println("Soil is moist.");
    stopPump();
  }
  delay(1000); // Check every second
}
```

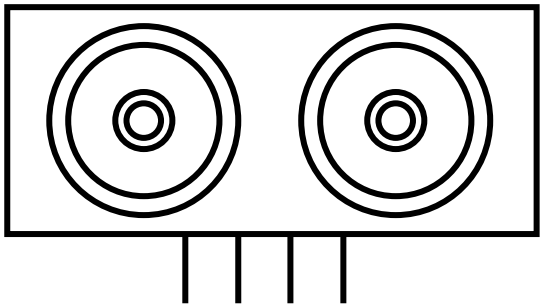
```
void startPump() {
  digitalWrite(pumpPin1, HIGH);
  digitalWrite(pumpPin2, LOW);
}

void stopPump() {
  digitalWrite(pumpPin1, LOW);
  digitalWrite(pumpPin2, LOW);
}

void alertUser() {
  digitalWrite(buzzerPin, HIGH);
  delay(200);
  digitalWrite(buzzerPin, LOW);
}
```

ACTIVITY 5

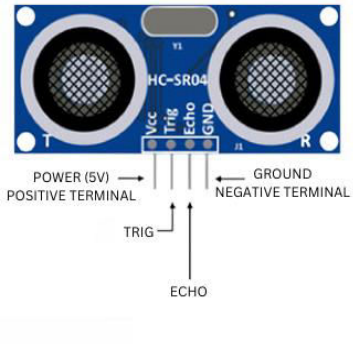
Advance sensor – Smart Goggles

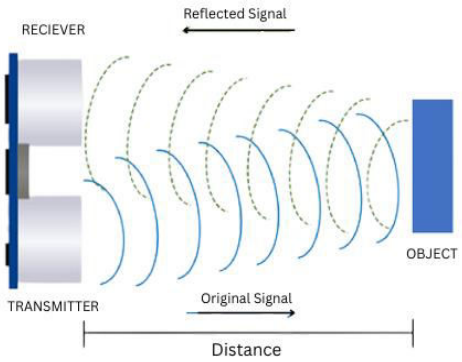
REQUIRED COMPONENTS:

1. Arduino Uno
2. Ultrasonic Sensor
3. Buzzer Module
4. 1-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)

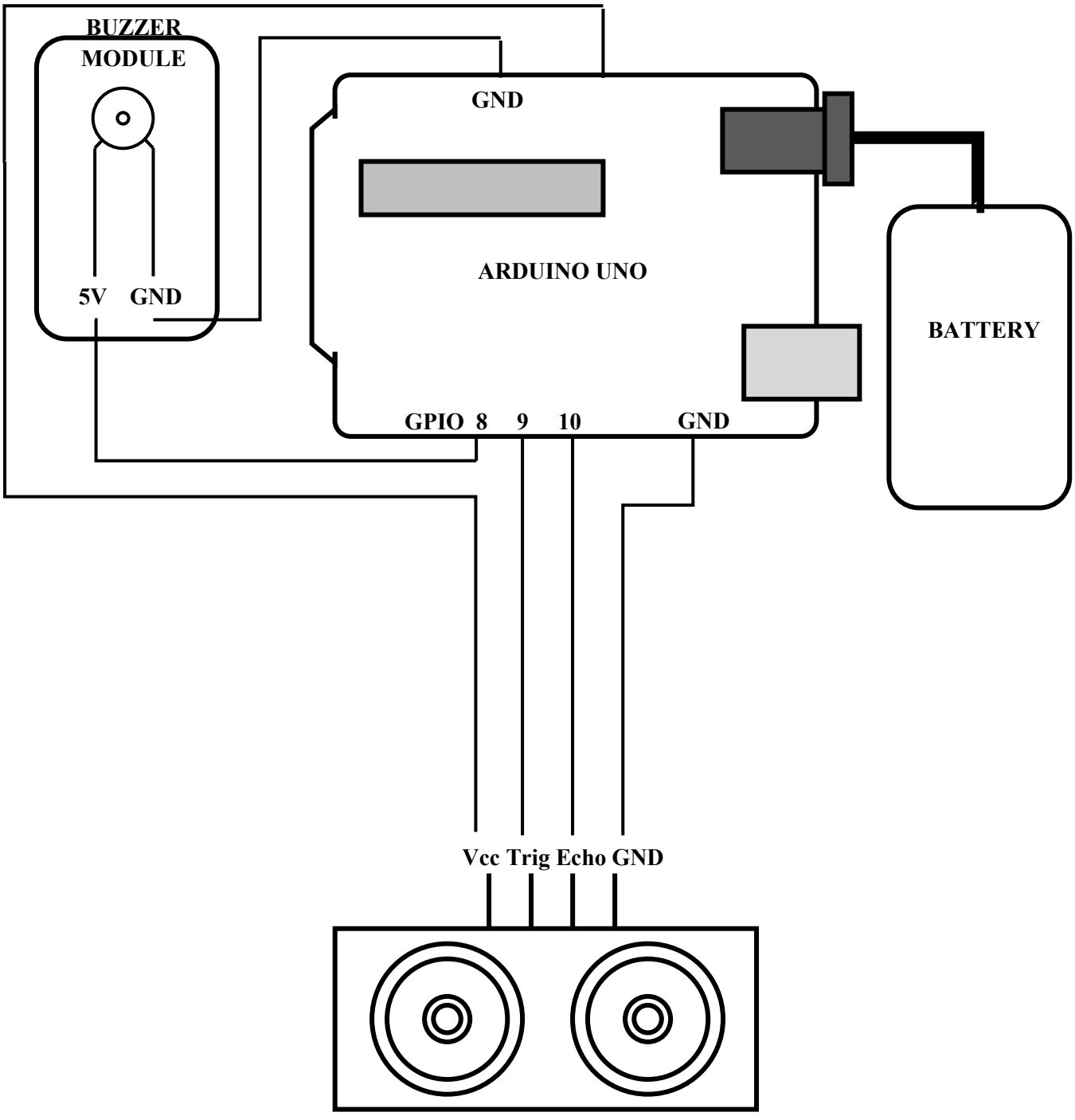
Formula:

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

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

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

CONNECTION DIAGRAM OF SMART GOGGLES:



Ultrasonic Sensor

PROGRAM CODE OF SMART GOGGLES USING ULTRASONIC SENSOR

```
#define trigPin 9
#define echoPin 10
#define buzzer 8
long duration;
int distance;
void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(buzzer, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  // Clear trig pin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Send ultrasonic pulse
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Read echo time
  duration = pulseIn(echoPin, HIGH);
  // Calculate distance (cm)
  distance = duration * 0.034 / 2;
  Serial.print("Distance: ");
  Serial.println(distance);
  if (distance > 0 && distance <= 50) {
    digitalWrite(buzzer, HIGH);
    delay(distance * 5);   // closer = faster beep
    digitalWrite(buzzer, LOW);
    delay(distance * 5);
  }
  else {
    digitalWrite(buzzer, LOW);
  }
}
```