ROBOTICS CLUB OF CEG

CODE SPEAK CONNECT

PHONETIX

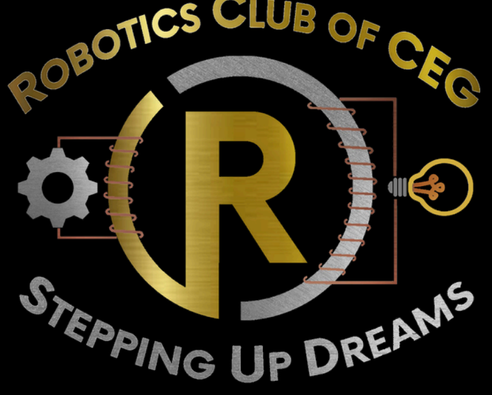
VOICE CONTROLLED BOT

ABOUT ROBOTICS CLUB OF CEG

- Official Robotics club of College of Engineering, Guindy
- Student driven technical club

Focus on:

- Robotics
- Embedded systems
- Automation and intelligent systems
- Promote practical learning, innovation, and teamwork



WHAT WE DO

- Conduct technical workshops
- Develop robotics and embedded systems projects
- Conduct technical events
- We mentor juniors in robotics and embedded systems.

Conduct Project Open Calls:

- Juniors propose innovative robotics ideas
- Seniors mentor teams
- Guidance on hardware design, coding, and testing

Design and development of:

- Embedded hardware solutions
- Robotic and autonomous systems

WORKSHOP OVERVIEW

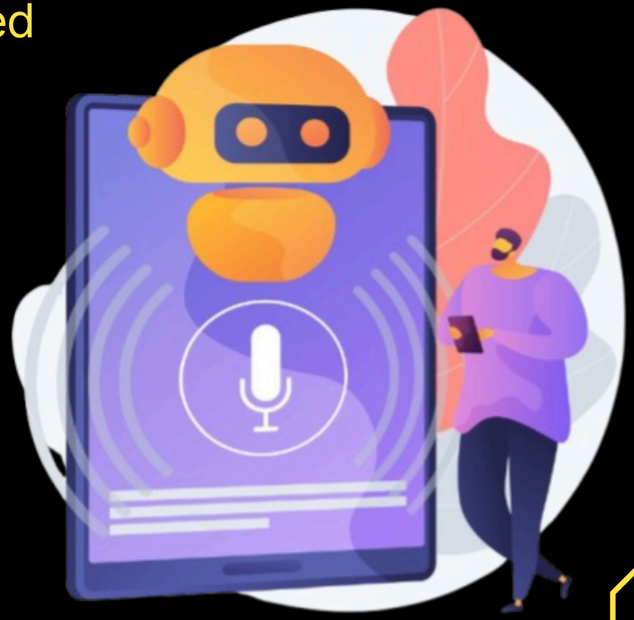
- Introduction to voice controlled robotics
- Embedded systems basics
- Explanation of each component used in the bot
- Introduction to Arduino IDE
- Building the bot step-by-step
- Code structure and programming workflow
- Coding, testing, and debugging the bot
- Live demonstrations and hands on practice

WHAT IS A VOICE CONTROLLED BOT?

VOICE CONTROLLED BOT

What it is?

- A bot that responds to human voice commands
- Commands are given through a mobile phone or microphone enabled device
- Voice is converted into control signals to operate the bot



How it works?

- User gives a voice command via a microphone
- Command is transmitted using Bluetooth
- Microcontroller processes the command
- Motors and actuators execute the action

APPLICATIONS

- Smart home automation
- Assistive robots for elderly and disabled
- Industrial material handling
- Educational robotics and learning platforms
- Surveillance and remote controlled systems

LEARNING OUTCOMES

- Basics of embedded system design
- Interfacing sensors and actuators
- Voice command processing
- Arduino programming workflow
- Hardware debugging and troubleshooting skills

EMBEDDED SYSTEMS BASICS

EMBEDDED SYSTEMS

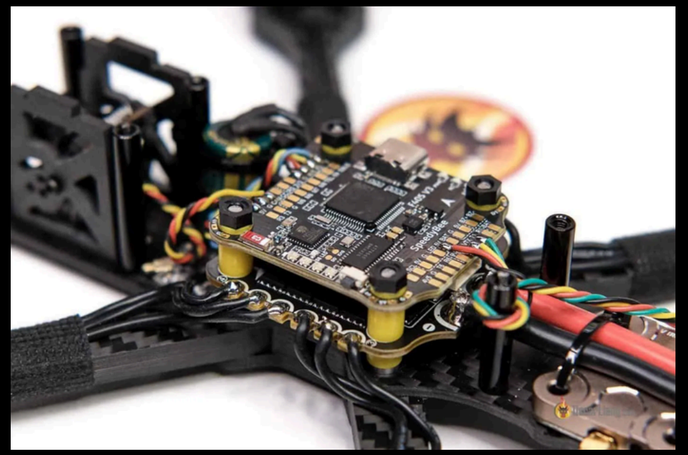
- An Embedded System is a combination of hardware and software designed to perform a specific, dedicated task within a larger system.
- It is not a general-purpose computer.

◆ Simple Explanation

“It is a small computer built inside a device to control it.”

REAL WORLD EXAMPLES

- Washing machine controller
- Car ECU (Engine Control Unit)
- Smart watch
- ATM machine
- Drone flight controller
- IoT devices



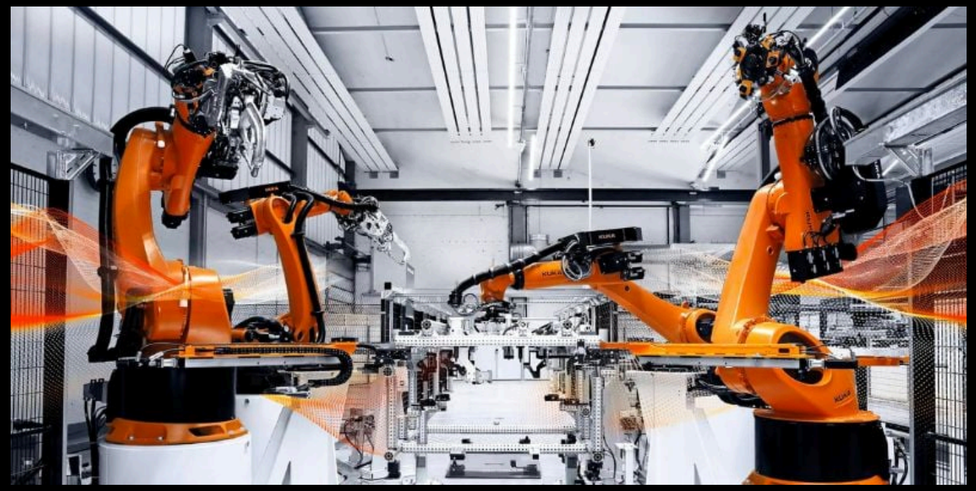
ROBOTICS

Robotics is the branch of engineering that deals with designing, building, and programming robots.

A programmable machine capable of performing tasks automatically.

- Robotics = Combination of:

Embedded Systems
Control Systems
Sensors
Actuators
AI (optional advanced level)

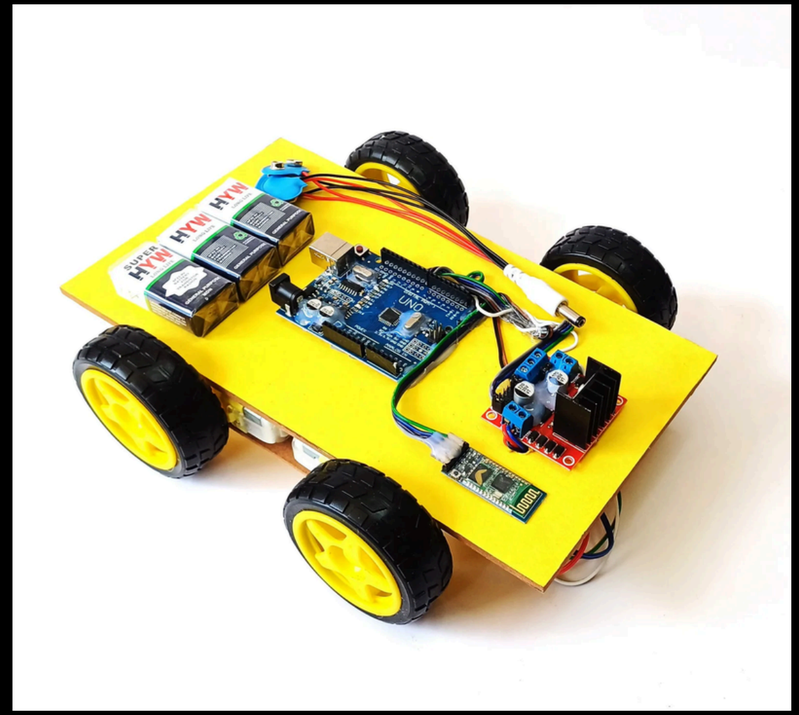


HOW ARE THEY CONNECTED ?

👉 Embedded system = Brain
👉 Sensors = Eyes & Ears
👉 Actuators = Hands & Legs

Without embedded systems → No robotics.

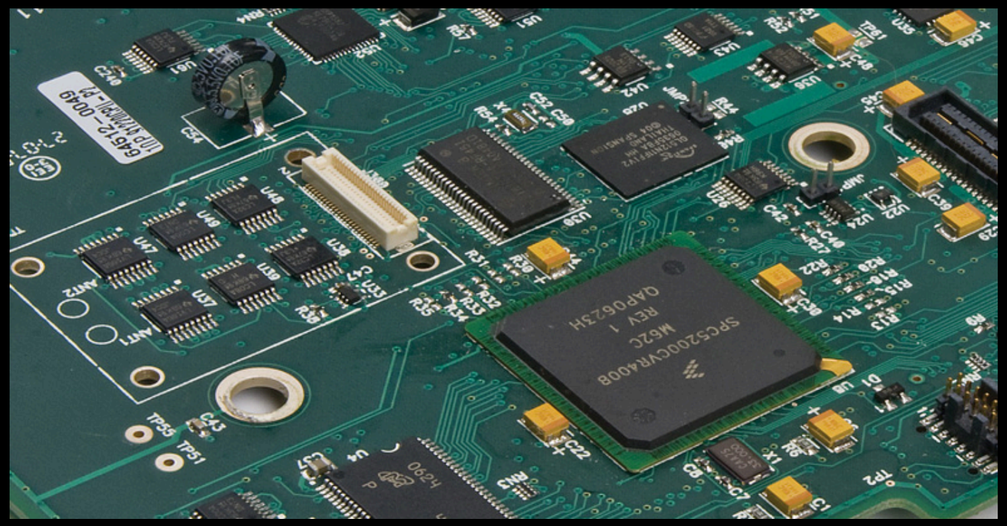
Voice Controlled Bot is a perfect example:
Voice → Controller → Motor Driver → Movement



IMPORTANCE

ECE students study:

- Digital Electronics
- Microprocessors & Microcontrollers
- Signals & Systems
- Communication
- Control Systems
- VLSI
- All of these are used in embedded systems.



STUDENTS SHOULD FOCUS ON

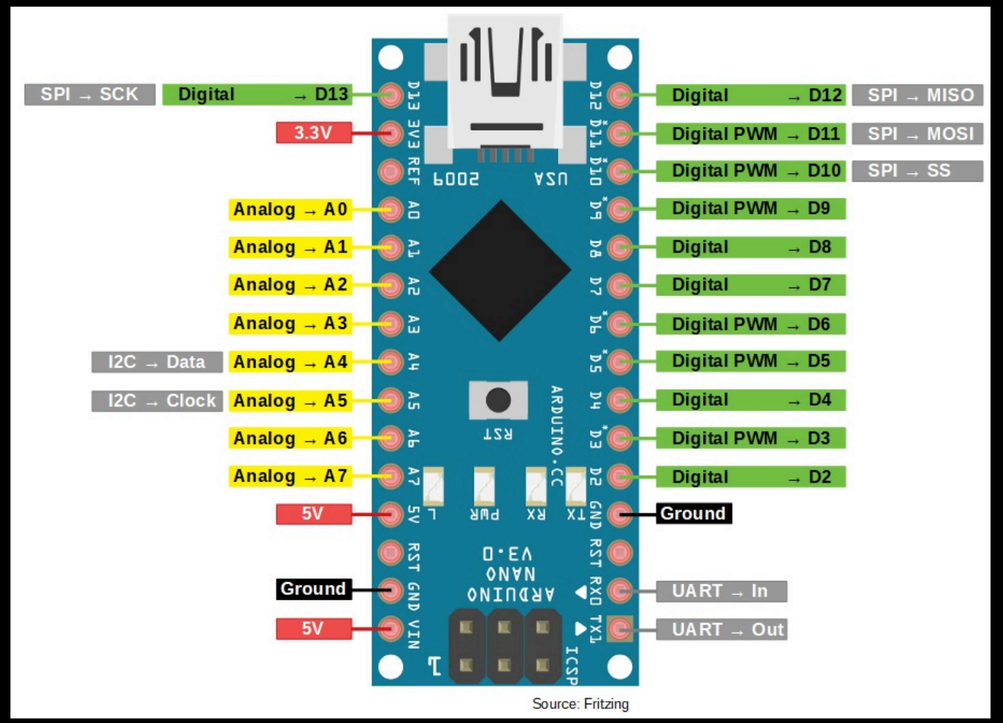
- Strong C Programming - Pointers, Structures, Bitwise operations, Memory Management, Interrupt Handling logic
- Microcontrollers Architecture - 8051, ARM, STM32, ESP32, Arduino Boards
- Digital Electronics - Flipflops, Counters, Registers, State Diagrams
- Communication Protocols - UART, SPI, I2C, CAN
- RTOS - Tasks, Scheduling, Mutex, Semaphores
- Basic PCB Designing - KiCad, Altium, EasyEDA

ARDUINO NANO

The Arduino Nano is a compact microcontroller board based on ATmega328P. It is the brain of the robot.

Key Features

- Based on ATmega328P
- Operating Voltage: 5V
- Digital I/O Pins: 14
- Analog Input Pins: 8
- Clock Speed: 16 MHz
- Small size and breadboard friendly



Source: Fritzing

ARDUINO NANO

- Role in Voice Controlled Bot
  - Receives commands from Bluetooth module
  - Processes the command
  - Sends control signals to motor driver
  - Controls direction of motors

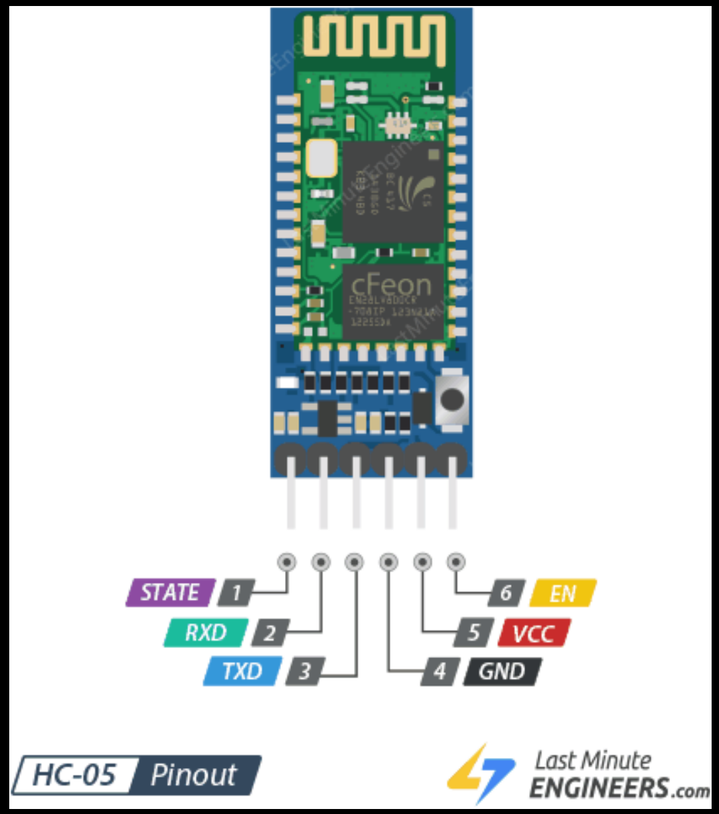
👉 Without Arduino → No decision making

BLUETOOTH MODULE

A Bluetooth module allows wireless communication between mobile phone and Arduino.

Commonly used module:
HC-05

- Key Features
  - Wireless communication (2.4 GHz)
  - Range: ~10 meters
  - Serial communication (UART)
  - Operates at 3.3V logic



BLUETOOTH MODULE

- Role in Voice Controlled Bot
  - Receives voice command from mobile app
  - Converts it into serial data
  - Sends data to Arduino via TX/RX pins

Example:

Voice → Mobile App → Bluetooth → Arduino

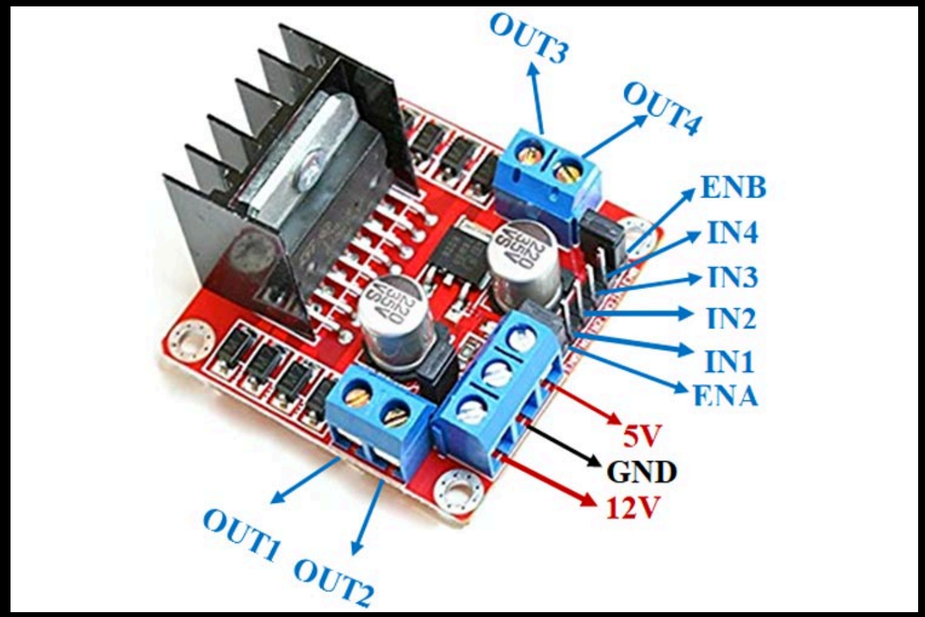
MOTOR DRIVER

Arduino cannot directly drive motors because:

- Motors require high current
- Arduino pins provide very low current

So we use a Motor Driver IC.

Common motor driver:
L298N



- Key Features
  - Dual H-Bridge motor driver
  - Controls 2 DC motors
  - Allows forward & reverse rotation
  - Separate motor power supply

MOTOR DRIVER

- Role in Bot
  - Receives control signals from Arduino
  - Supplies required current to motors
  - Controls motor direction

👉 It acts like a power amplifier between Arduino and motors.

DC MOTORS

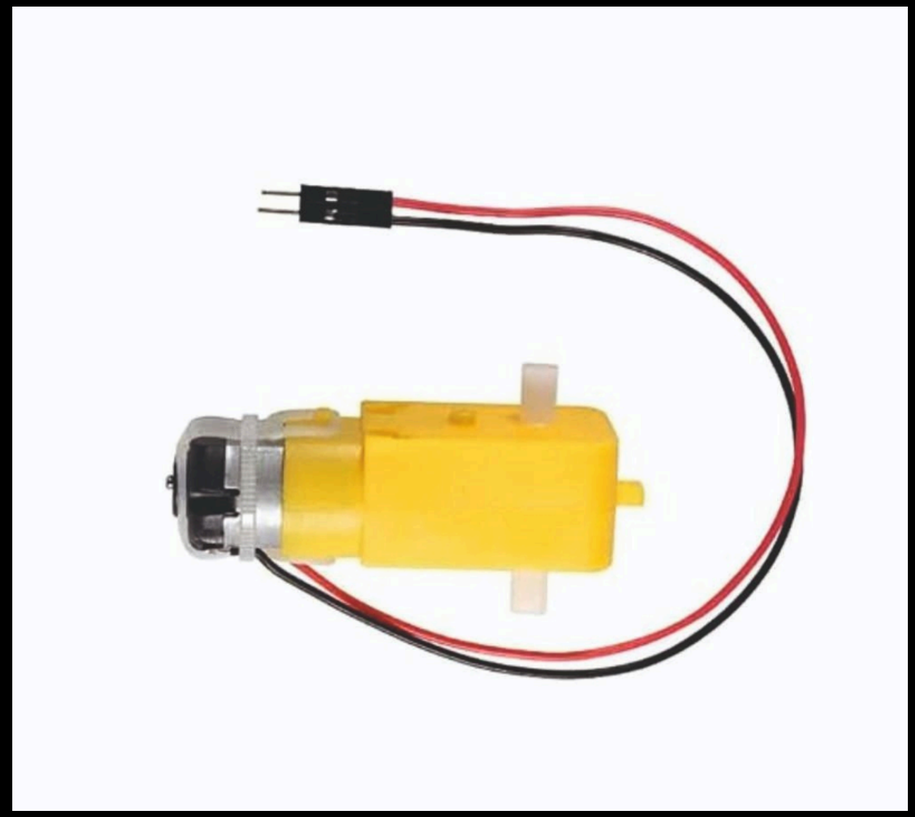
A DC motor converts electrical energy into mechanical rotation.

Used to move the robot.

- Types Used
  - Geared DC motors
  - 100–300 RPM motors (commonly used in bots)
- Role in Voice Controlled Bot
  - Convert electrical signals into movement
  - Enable forward, backward, left & right motion

Movement logic:

- Both motors forward → Forward
- Both motors reverse → Backward
- One motor forward → Turn



HARDWARE WORKING FLOW

Voice Command

↓

Mobile App

↓

Bluetooth Module

↓

Arduino Nano

↓

Motor Driver

↓

DC Motors

↓

Robot Movement

BREAK

Let’s Take a Break 15 Minutes!

HOW ANY ROBOT WORKS

BASIC ROBOT ARCHITECTURE

🎤 Input (Sensor)

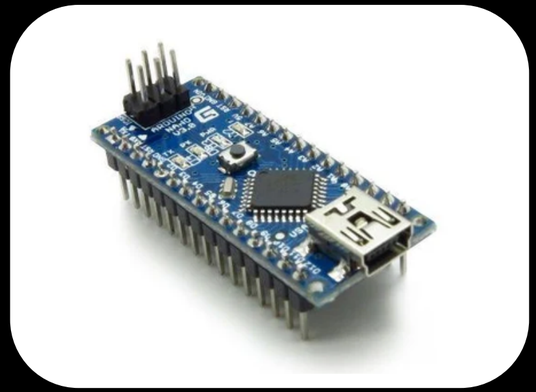
🧠 Processor (Brain)

⚙️ Output (Actuator)

HOW ANY ROBOT WORKS

BASIC ROBOT ARCHITECTURE

🎤 Input (Sensor) ⟶ 🧠 Processor (Brain) ⟶ 🛞 Output (Actuator)



🎯VOICE TO WHEEL COMMUNICATION

🎤 **Mobile App → 📡 HC-05 Bluetooth→ 🧠 Arduino→ ⚡ Motor Driver→ 🛞 Wheels**

STRUCTURE OF ARDUINO PROGRAM

Declaration Section
📌 Libraries
📌 Pin definitions

setup()

📌 Initialisation forlibrary
📌 Configure pin modes
📌 Initial conditions

loop()

🔄 Read inputs
🧠 Process data
⚡ Control outputs

setup()
Runs only once

loop()
Runs continuously

```
/* ========= LIBRARIES ========= */
// #include <LibraryName.h>

/* ========= PIN DEFINITIONS ========= */

/* ========= GLOBAL VARIABLES ========= */
void setup()
{
  /* ---- INITIALIZATION ---- */
}

void loop()
{
  /* ---- INPUT SECTION ---- */

  /* ---- PROCESS SECTION ---- */

  /* ---- OUTPUT SECTION ---- */

}
```

ARDUINO BASIC FUNCTION AND PROGRAM

pinMode() - Configure a pin as INPUT or OUTPUT
pinMode(pinNumber, mode);

Example: pinMode(3, OUTPUT);
pinMode(2, INPUT);

digitalWrite() - Give HIGH or LOW output

digitalWrite(pinNumber, value);

Example: digitalWrite(3, HIGH);

digitalRead() - Read HIGH or LOW from a pin

int value = digitalRead(pinNumber);

analogWrite() -PWM output (speed / brightness control)

analogWrite(pinNumber, value);

0 - 0% duty (OFF)
255 - 100% duty (FULL ON)

Example: analogWrite(5, 128);

analogRead() – Reading Real World Signals

📈 0 → 1023 scale graphic

int value = analogRead(pin);

VOICE CONTROLLED BOT - CODE

```
#include <SoftwareSerial.h>

// HC-05 Bluetooth
SoftwareSerial BT(10, 11); // RX, TX

// Motor pins
const int IN1 = 3, IN2 = 4, ENA = 5;    // Left motor
const int IN3 = 6, IN4 = 7, ENB = 9;    // Right motor

const int SPEED = 100;      // 0–255
const int RUN_TIME = 2000; // 2 seconds
void setup() {
  Serial.begin(9600);
  BT.begin(9600);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(ENA, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENB, OUTPUT);
  stopBot();
  Serial.println("Voice Controlled Bot Ready");
}
```

#include <SoftwareSerial.h>
Allows Arduino to create an extra serial port using digital pins.
SoftwareSerial BT(10, 11);

- Pin 10 - RX (receives data from Bluetooth)
- Pin 11 - TX (sends data to Bluetooth)

Start Serial Communication:
Serial.begin(9600);
Used to: Display messages on Serial Monitor
Example: Serial.println("Voice Controlled Bot Ready

Start Bluetooth Communication
BT.begin(9600);
Starts communication with the HC-05 module.

Pin	Purpose
IN1, IN2	Left motor direction
ENA	Left motor speed (PWM)
IN3, IN4	Right motor direction
ENB	Right motor speed (PWM)

CONTROL LOOP

```
void loop() {
  if (BT.available()) {
    char cmd = BT.read();
    Serial.print("Command: ");
    Serial.println(cmd);
    switch (cmd) {
      case 'f':
        forward();delay(RUN_TIME);
        stopBot();break;
      case 'b':
        backward();delay(RUN_TIME);
        stopBot();break;
      case 'l':
        left();delay(RUN_TIME);
        stopBot();break;
      case 'r':
        right();delay(RUN_TIME);
        stopBot();break;
      case 's':
        stopBot();break;
    }
  }
}
```

MOTOR DRIVER CONTROL LOGIC

ENA	IN1	IN2	The State of DC Motor A
0	X	X	Stop
1	0	0	Brake
1	0	1	Rotate Clockwise
1	1	0	Rotate Counterclockwise
1	1	1	Brake

FORWARD LOGIC

```
void forward() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  analogWrite(ENA, SPEED);
  analogWrite(ENB, SPEED);
}
```

Movement	Left Motor	Right Motor
Forward	Forward	Forward
Backward	Backward	Backward
Left	Backward	Forward
Right	Forward	Backward

DESIGN THE MOVEMENT LOGIC FOR THE ROBOT

BOT MOVEMENT LOGIC :

```
void right() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
  analogWrite(ENA, SPEED);
  analogWrite(ENB, SPEED);
}

void stopBot() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
  analogWrite(ENA, 0);
  analogWrite(ENB, 0);
}
```

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

void left() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  analogWrite(ENA, SPEED);
  analogWrite(ENB, SPEED);
}
```

THANKYOU

Robotics Is Shaping Tomorrow