4) BUILD A BASIC SPEECH RECOGNITION SYSTEM FOR COMMAND-BASED CONTROL OF DEVICES USING AN EMBEDDED BOARD.

ANS: project guide to **build a basic speech recognition system** for **command-based control of devices** using an **embedded board** (like Arduino + voice module).

**✅ PROJECT OVERVIEW**

**🎯 Objective:**

* Use **speech commands** to **control devices (like LEDs, fans)**.
* Interface a **speech recognition module** with an **embedded board** (Arduino Uno).
* Perform actions based on recognized keywords like “ON”, “OFF”, “FAN”, etc.

**🧰 COMPONENTS REQUIRED**

| **Component** | **Quantity** |
| --- | --- |
| Arduino Uno | 1 |
| Voice Recognition Module V3 (Elechouse) | 1 |
| Relay Module / LEDs | 2–4 |
| Jumper Wires + Breadboard | As needed |
| USB Cable (for programming) | 1 |
| Power Supply (5V regulated) | 1 |

**🔌 SYSTEM DESIGN**

**🔹 Voice Recognition Module V3**

* The module is trained with voice commands like:
  + "Light On"
  + "Light Off"
  + "Fan On"
  + "Fan Off"

**🔹 Typical Wiring (Voice Module ↔ Arduino Uno)**

| **Voice Module Pin** | **Arduino Uno Pin** |
| --- | --- |
| VCC | 5V |
| GND | GND |
| RX | D2 |
| TX | D3 |

Use Software Serial on pins D2 (RX) and D3 (TX).

**ARDUINO CODE**

**#include <SoftwareSerial.h>**

**SoftwareSerial voice(2, 3); // RX, TX**

**const int lightPin = 8;**

**const int fanPin = 9;**

**void setup() {**

**pinMode(lightPin, OUTPUT);**

**pinMode(fanPin, OUTPUT);**

**Serial.begin(9600);**

**voice.begin(9600);**

**Serial.println("Voice control system ready.");**

**}**

**void loop() {**

**if (voice.available()) {**

**int command = voice.read();**

**Serial.print("Command Received: ");**

**Serial.println(command);**

**switch (command) {**

**case 0: // "Light On"**

**digitalWrite(lightPin, HIGH);**

**Serial.println("Light turned ON");**

**break;**

**case 1: // "Light Off"**

**digitalWrite(lightPin, LOW);**

**Serial.println("Light turned OFF");**

**break;**

**case 2: // "Fan On"**

**digitalWrite(fanPin, HIGH);**

**Serial.println("Fan turned ON");**

**break;**

**case 3: // "Fan Off"**

**digitalWrite(fanPin, LOW);**

**Serial.println("Fan turned OFF");**

**break;**

**default:**

**Serial.println("Unknown command");**

**break;**

**}**

**}**

**}**

**TO TRAIN THE MODULE**

1. **Connect Voice Module to PC via USB-to-Serial module (or directly to Arduino with passthrough sketch).**
2. **Use Voice Recognition Software from Elechouse.**
3. **Record commands:**
   * **Group 1: 0 = Light On, 1 = Light Off**
   * **Group 2: 2 = Fan On, 3 = Fan Off**

**BLOCK DIAGRAM (Textual)**

**[Voice Module]**

**| RX → D3 (TX)**

**| TX → D2 (RX)**

**| VCC → 5V**

**| GND → GND**

**↓**

**[Arduino Uno]**

**| Pin 8 → Light (LED / Relay)**

**| Pin 9 → Fan (LED / Relay)**

**📺 WORKING DEMO**

**🔹 When user says:**

* **"Light On" → LED or bulb turns ON**
* **"Fan Off" → Relay controlling fan turns OFF**

**You can demonstrate this using LEDs or relays connected to appliances.**

**✅ DELIVERABLE SUMMARY**

| **Deliverable** | **Status** |
| --- | --- |
| **✅ System Design** | **Arduino + Voice Module** |
| **✅ Code** | **Ready (Above)** |
| **✅ Demo Plan** | **Voice → Action Mapping** |
| **✅ Training Method** | **Elechouse software** |

**📦 OPTIONAL ENHANCEMENTS**

* **Use ESP32 + Google Assistant for cloud voice control**
* **Use MIT App Inventor for mobile-based command input**
* **Add LCD display to show recognized command**