**SPEECH RECOGNITION SYSTEM**

**1. Introduction**

This project presents the development of a **voice-controlled system** that interprets verbal commands to simulate the control of devices such as lights. Using **Python and SpeechRecognition APIs**, the system listens to real-time audio through the device's microphone and performs predefined actions based on recognized keywords.The solution demonstrates how voice interfaces can be integrated into embedded systems for automation applications — particularly useful in smart homes, accessibility tools, and hands-free environments.

**2. Design and Methodology**

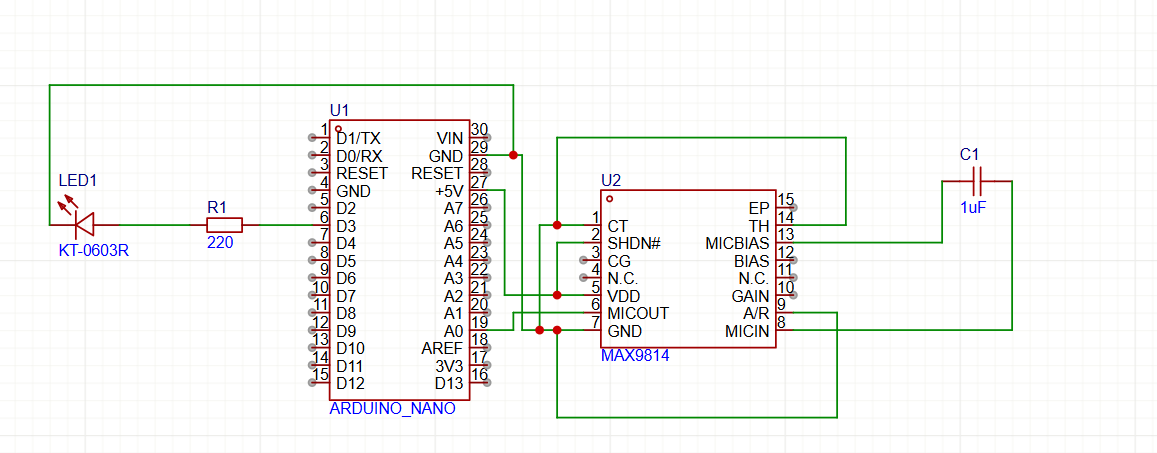
The system uses the **SpeechRecognition** library in Python to capture and interpret voice commands from the user via a built-in microphone. Based on the spoken command, the program simulates turning a device ON or OFF by printing a message or updating a GUI (optional).

The system's logic flow was designed in software and visually represented using **EasyEDA**, an online electronics design tool. Although the project uses a Python-based simulation without physical hardware, EasyEDA was used to illustrate the virtual circuit diagram to represent the conceptual flow of input, processing, and output.

**3.components Required**

| Type | Component / Tool |
| --- | --- |
| Software | Python 3.12+ |
| IDE | Visual Studio Code (VS Code) |
| Libraries | speech\_recognition, pyaudio, pipwin |
| Input Device | Built-in Microphone (Laptop) |
| Output Device | Simulated LED (console output / GUI) |
| Simulation | Arduino (conceptual only) |
| Design Tool | EasyEDA |

**4.Circuit Design**



**5.Code**

import speech\_recognition as sr

def control\_device(command):

    if "on" in command.lower():

        print(" Light turned ON")

    elif "off" in command.lower():

        print(" Light turned OFF")

    else:

        print("Unknown command")

recognizer = sr.Recognizer()

mic = sr.Microphone()

print(" Say 'Turn on light' or 'Turn off light'...")

while True:

    with mic as source:

        print("Listening...")

        audio = recognizer.listen(source)

    try:

        command = recognizer.recognize\_google(audio)

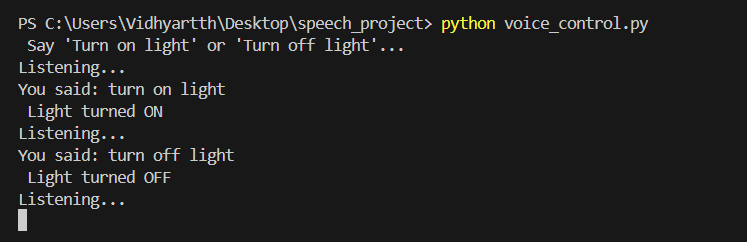
        print("You said:", command)

        control\_device(command)

    except:

        print("Sorry, I didn't catch that.")

**6. Output Demonstration**

****

**7.Conlusion**

This project successfully demonstrates a **software-based speech recognition system** designed for device control without physical hardware. By utilizing Python and open-source voice recognition libraries, we simulated an embedded automation system controlled entirely by voice input.

This not only showcases the potential for integrating **natural language interfaces** into embedded systems but also allows students and developers to prototype smart solutions in a fully virtual environment. It is an effective and accessible approach for internships, educational prototypes, and early-stage smart automation projects.