**SPEECHLESS**

Voice Recognition Wake Word Model

Aman Subash

CB.EN.U4CSE21305

Amrita School Of Computing

Coimbatore, India

[cb.en.u4cse21305@cb.students.amrita.edu](mailto:cb.en.u4cse21305@cb.students.amrita.edu)

Rishi Pradeep Kumar

CB.EN.U4CSE21347

Amrita School Of Computing

Coimbatore, India

[cb.en.u4cse21347@cb.students.amrita.edu](mailto:cb.en.u4cse21347@cb.students.amrita.edu)

Gautham Sreedhar

CB.EN.U4CSE21313

Amrita School Of Computing

Coimbatore, India

[cb.en.u4cse21313@cb.students.amrita.edu](mailto:cb.en.u4cse21313@cb.students.amrita.edu)

Pranav Sreerag   
CB.EN.U4CSE21342  
Amrita School Of ComputingCoimbatore, India  
[cb.en.u4cse21342@cb.students.amrita.edu](mailto:cb.en.u4cse21342@cb.students.amrita.edu)

"Speechless" employs a Raspberry Pi Pico to create a Wake-Word Detection. Using the Pico's GPIO pins and a microphone input, we detect the words "yes" or "no". When the Raspberry Pi Pico detects the affirmative "yes," it illuminates an attached LED, providing a visual confirmation. Conversely, upon recognizing the negative response "no," the LED deactivates. This integrated system ensures potential for creating interactive and responsive systems through voice recognition and simple hardware integration.

Keywords—Raspberry Pi Pico, Microphone, LED

# Introduction

In the dynamic realm of embedded systems, the Raspberry Pi Pico, powered by the RP2040 microcontroller, emerges as a robust platform for innovative projects. This venture focuses on the captivating domain of wake-word detection, a cutting-edge application in voice recognition technology. The cornerstone of this setup lies in the integration of the Raspberry Pi Pico RP2040 with the ADMP401 microphone, a sensitive and compact device designed for audio capture. This helped form “Speechless”, a responsive system capable of discerning specific wake words from ambient sound. The ADMP401 microphone, with its high signal-to-noise ratio and omnidirectional sensitivity, becomes the ears of our setup, capturing audio input with precision. Its architecture mainly involves Raspberry Pi Pico's GPIO pins , which will be employed to interface with the ADMP401, facilitating seamless communication between the microcontroller and the microphone. “Speechless” epitomizes a versatile and potent platform for voice-controlled applications, paving the way for enhanced interactivity in the world of embedded systems.

# Architecture

Speechless’s architecture was meticulously chosen to facilitate seamless integration and efficient functionality. At the heart of this system lies the Raspberry Pi Pico microcontroller, meticulously selected for its compact form factor, GPIO (General Purpose Input/Output) pins, and processing capabilities. The Raspberry Pi Pico serves as the central control unit, effectively managing and coordinating interactions between various hardware components. Its GPIO pins facilitate communication with A microphone and audio circuitry to capture spoken audio and feed it into the Raspberry Pi Pico. The Raspberry Pi Pico runs the optimized TensorFlow Lite Micro wake word detection model, the microphone as audio input and runs locally for efficient inference and quick reaction time when detecting the specified wake words. Additionally, the microcontroller's compatibility with diverse sensor types, ease of programming, and low power consumption render it an ideal choice for the Speechless’s core functionality. The decision to employ the Raspberry Pi Pico stems from its robustness, versatility, and ability to serve as the neural hub orchestrating Speechless's operations.

# Components

|  |  |  |
| --- | --- | --- |
| S.no | Components | Quantity |
| 1 | Raspberry Pi Pico | 1 |
| 2 | ADMP401 Microphone | 1 |
| 3 | Connecting Wires | 3 |
| 4 | Breadboard | 1 |
| 5 | USB Cable | 1 |

## Raspberry Pi Pico

The Raspberry Pi Pico serves as the central processing unit and control hub of the Speechless system. This microcontroller offers a compact form factor, GPIO (General Purpose Input/Output) pins for interfacing with the microphone, and efficient processing capabilities. Its role involves coordinating sensor data acquisition, processing information, and triggering appropriate actions based on predefined conditions.

## ADMP401 Microphone

The ADMP401 microphone plays a pivotal role in the wake-word detection project, seamlessly interfacing with the Raspberry Pi Pico RP2040 to capture and process audio input. This compact yet powerful microphone is chosen for its exceptional sensitivity and high signal-to-noise ratio, ensuring accurate recognition of wake words amidst varying ambient noise levels. The ADMP401 boasts omnidirectional sensitivity, enabling it to pick up sound from all directions, making it ideal for applications where capturing nuanced audio cues is crucial.

## Connecting Wires

The process of connecting wires in the wake-word detection project involves establishing a seamless interface between the Raspberry Pi Pico RP2040 and the ADMP401 microphone. Begin by identifying the GPIO pins on the Raspberry Pi Pico, specifically selecting pins for data transfer, power, and ground.

## Breadboard

A breadboard serves as a fundamental tool for creating a prototype with organized and temporary connections. The breadboard simplifies the process of connecting wires, facilitating a modular and adjustable setup. The breadboard's rows and columns can be used to establish connections between the GPIO pins of the Raspberry Pi Pico and the respective pins on the ADMP401, such as power, ground, and data.

## USB Cable

The USB cable is used to connect the Raspberry Pi Pico to the system and is a critical component for power and data transfer. The USB cable facilitates a reliable and convenient link between the Raspberry Pi Pico and a host system, such as a computer or a power source. The USB cable typically features a USB Type-A connector on one end, which can be plugged into a USB port on a computer or a USB power adapter. The other end of the cable is equipped with a micro-USB connector, which is compatible with the Raspberry Pi Pico's micro-USB port. This connection serves a dual purpose: it provides power to the Raspberry Pi Pico and establishes a data link for programming and communication between the microcontroller and the host system.

# Methodology

## Requirement Analysis and Planning

The methodology for creating Speechless commenced with a comprehensive analysis of requirements and objectives. This involved defining the functionalities, such as Recognizing specific wake words, such as "yes" and "no," in real-time audio input. Planning encompassed outlining the hardware components, microphone selection, and defining software architecture to align with the project goals.

## Hardware Selection and Integration

The selection of hardware components, including the Raspberry Pi Pico, ADMP401 microphone, connecting wires, breadboard and USB cables was based on their compatibility, functionalities, and synergy with the intended system design. Integrating these components involved connecting the microphone to the Raspberry Pi Pico using appropriate connecting wires, configuring GPIO pins, and ensuring seamless communication between the hardware elements.

## Software Development

The software development process for the wake-word detection project involves creating a program that runs on the Raspberry Pi Pico RP2040 to capture audio input, process it for wake-word recognition, and control the LED based on the detected words. For the model, a neural network architecture suited for speech recognition was developed in TensorFlow.

For deployment, C code was written to capture live audio from the microphone, prepare the input, run inferences through the TFLite Micro model, and activate the wake word LED when detected.

## Integration and System Testing

The integration phase involved combining hardware and software components to create a coordinated system. System testing encompassed validating the entire Speechless system, conducting many tests in which multiple accents were used for saying the recognized wake words. Testing is aimed to ensure seamless interactions between the components.

## Iterative Refinement and Deployment

Iterative refinement involved aiming to enhance functionality, address discovered issues, and prepare for practical deployment. Issue Identification and User Feedback Incorporation were part of this phase. Upon successful testing, the Speechless system was deployed for real-world use, ready to offer efficient a responsive and accurate voice recognition system.

# Functionality

## Wake Word Detection

The core functionality is to listen for and detect a specific wake word or phrase like "Yes" or “No”. The machine learning model running on the RP2040 microcontroller is optimized to spot this phrase in real-time.

## Low Power Operation

The project emphasizes low power consumption for always-on listening. The Pico SDK and MicroPython environment enable processing the audio data from the microphone in an efficient manner on battery power.

## Custom Wake Words

Unlike commercial wake word systems, this allows creating custom models to detect unique phrases specified by the user. The model can be re-trained on new datasets to update the wake vocabulary.

## Real-Time Responsiveness

 Spotting the wake word triggers a real-time output, lighting an LED indicator. This showcases the capabilities for responsive wake word agents that can turn on voice applications after the detected trigger phrase.

# Usage Instructions:

* **Speak Clearly:** Vocalize the words properly and make it loud and clear.
* **Attach the pins correctly:** Make sure the pins are connected properly to the ground and power.

# Conclusion

The Speechless system represents a huge leap in voice recognition technology, seamlessly blending quantization, MicroPython integration, and GPIO triggering to redefine voice recognition. By advocating the use of ADMP401 microphone , we were able to come up with Speechless. Training in TensorFlow, model optimization with TensorFlow Lite Micro, and integration with microphone hardware and output triggers resulted in an accurate and responsive voice trigger system. The Speechless wake word detection system can pave the way for voice control and commands. As the Speechless emphasizes the convergence of cutting-edge technology and practical utility, it also shows that the ability to listen and respond to verbal cues unlocks smarter interfaces and more intuitive control.

# Future Scope

**Voice Control for Custom Devices**: The wake word trigger can activate voice control and commands for homemade smart devices. For example, using it to control 3D printed robots, IoT gadgets, or electronics projects like a voice-activated home automation system.

**Voice User Interfaces in Cars:** The project could be integrated into cars to provide wake word activation for speech control over entertainment systems, navigation, windows etc. Useful for drivers to keep hands engaged in driving.

**Voice Control for Games:** It can provide a customizable and affordable smart speaker solution with custom hot words for activating music playback. Great for hobbyist smart speaker builds.

Expanding the Speechless's functionalities in the above given examples helps to promote affordable and easy way to add offline control interface. The integration of more sophisticated features for specific use cases could further enhance its usability, establishing it as an indispensable asset in voice recognition technology.

# Softwares Used

## Thonny

* *Usage:* It is used to monitor the output

## CMake

* *Usage:* It is used to build the program files

## C/C++

* *Usage:* The programming language used for the model

## TensorFlowLite

* *Usage:* It is used to import the model.