

Instructivo Paso a Paso: Implementación del Sistema de Reconocimiento de Voz ESP32

Este instructivo detalla el proceso completo para implementar, configurar y utilizar el sistema de reconocimiento de voz para ESP32 utilizando PlatformIO.

1. Configuración del Entorno PlatformIO

1.1 Requisitos Previos

- Visual Studio Code instalado
- Extensión PlatformIO instalada en VS Code
- Drivers USB para ESP32 instalados

1.2 Creación del Proyecto

1. Abrir VS Code
2. Hacer clic en el ícono de PlatformIO en la barra lateral izquierda
3. Seleccionar "PlatformIO Home" → "New Project"
4. Configurar el proyecto:
 - **Name:** ESP32_Voice_Recognition
 - **Board:** Espressif ESP32 Dev Module
 - **Framework:** Arduino
 - **Location:** (seleccionar ubicación o dejar la predeterminada)
5. Hacer clic en "Finish"

1.3 Configuración del Archivo platformio.ini

Una vez creado el proyecto, abre el archivo `platformio.ini` y reemplaza su contenido con:

ini

[platformio]

name = ESP32 Voice Recognition

description = Sistema de reconocimiento de voz para ESP32 (versión simplificada)

default_envs = esp32dev

[env:esp32dev]

platform = espressif32

board = esp32dev

framework = arduino

monitor_speed = 115200

board_build.partitions = default.csv

build_flags =

-DCORE_DEBUG_LEVEL=3

2. Implementación del Código

2.1 Estructura del Código

El sistema utiliza un único archivo fuente `main.cpp`. No se requieren bibliotecas adicionales.

2.2 Implementación del Código

1. Abre el archivo `src/main.cpp`
2. Elimina todo el contenido existente
3. Copia y pega el código completo proporcionado en la sección "Código Completo" al final de este instructivo

3. Compilación y Carga

3.1 Compilación del Proyecto

1. En la barra lateral de PlatformIO, expande el proyecto
2. Selecciona "Project Tasks" → "Build"
3. Verifica que la compilación finalice sin errores

3.2 Carga al ESP32

1. Conecta el ESP32 a tu computadora mediante USB
2. Asegúrate de que el dispositivo sea reconocido (aparecerá un puerto COM)
3. En "Project Tasks", selecciona "Upload"
4. Espera a que finalice la carga del firmware

3.3 Monitoreo Serial

1. En "Project Tasks", selecciona "Monitor" (o presiona el ícono enchufe en la barra de estado)

Verifica que aparezca el mensaje de inicialización:

- Sistema de Reconocimiento de Voz ESP32 (Simple)
- I2S inicializado correctamente

4. Utilización del Sistema

4.1 Operación Normal

1. Con el sistema cargado y funcionando, acércate al micrófono (15-30 cm)
2. Pronuncia claramente una de las palabras clave:
3. Observa el LED correspondiente iluminarse y los mensajes en el monitor serial

4.2 Ajuste de Parámetros

Si el sistema no responde adecuadamente, puedes ajustar estos parámetros en el código:

1. **Umbral de energía:** Si no detecta comandos, reduce `ENERGY_THRESHOLD`
cpp
`#define ENERGY_THRESHOLD 2000000` // Probar con valores menores, ej:
`1000000`
2. **Umbral de similitud:** Si hay falsos positivos, aumenta el valor de `bestScore`
cpp
`float bestScore = 0.65;` // Aumentar a 0.70 o 0.75 para mayor precisión

5. Calibración del Sistema

El sistema incluye un modo de calibración para adaptarse a tu voz. Para utilizarlo:

5.1 Ingreso al Modo Calibración

1. Apaga el ESP32
2. Conecta GPIO15 a GND (pin de calibración)
3. Enciende el ESP32

Verifica en el monitor serial el mensaje:

=== MODO DE CALIBRACIÓN ===

- Este modo te ayudará a ajustar los perfiles para cada palabra.

5.2 Proceso de Calibración

1. Sigue las instrucciones en el monitor serial
2. Cuando se indique, pronuncia claramente cada palabra
3. El sistema capturaré 5 muestras de cada palabra

Para cada palabra, obtendrás un perfil como este:

Patrón detectado para 'adelante':

Bajas: 0.45, Medias: 0.75, Altas: 0.95, Muy altas: 0.30

Copia estos valores en el código:

4. `{"adelante", LED_ADELANTE, 0.45, 0.75, 0.95, 0.30},`

5.3 Aplicar Calibración

1. Anota los perfiles generados para cada palabra
2. Abre el código fuente (`main.cpp`)

Localiza la sección de definición de comandos:

cpp

```
VoiceCommand commands[4] = {  
    {"adelante", LED_ADELANTE, 0.3, 0.5, 1.0, 0.2},  
    {"atras", LED_ATRAS, 0.5, 1.0, 0.4, 0.1},  
    {"derecha", LED_DERECHA, 0.4, 0.7, 0.7, 0.3},  
    {"izquierda", LED_IZQUIERDA, 1.0, 0.6, 0.3, 0.1}
```

3. };

4. Reemplaza con los nuevos valores obtenidos durante la calibración

5. Recompila y carga el nuevo firmware

6. Solución de Problemas

6.1 No se detecta el micrófono

- Verifica las conexiones del micrófono INMP441
- Confirma que el ESP32 está correctamente alimentado (5V USB)
- Asegúrate de que el pin L/R del micrófono está conectado a GND

6.2 No responde a comandos de voz

- Reduce el valor de **ENERGY_THRESHOLD** (probar con 1000000 o menos)
- Habla más cerca del micrófono y con mayor volumen

6.3 Asegúrate de que no hay ruido excesivo en el ambiente Reconocimiento incorrecto

- Realiza la calibración adaptada a tu voz
- Pronuncia las palabras de manera clara y consistente
- Aumenta el número de bandas de frecuencia (**NUM_BANDS**) para mayor precisión

6.4 LEDs no se encienden

- Verifica las conexiones de los LEDs
- Comprueba que las resistencias estén correctamente instaladas
- Confirma que los pines GPIO están correctamente definidos

7. Expansión y Personalización

7.1 Cambiar Comandos de Voz

Para cambiar los comandos reconocidos:

1. Modifica los nombres en el array **commands**:

cpp

```
VoiceCommand commands[4] = {  
    {"arriba", LED_ADELANTE, 0.3, 0.5, 1.0, 0.2}, // Cambiar "adelante" por "arriba"
```

```
// ... otros comandos
```

```
};
```

2. Realiza una nueva calibración para los nuevos comandos

7.2 Añadir Más Comandos

Para añadir más comandos (requiere modificaciones más extensas):

1. Aumenta el tamaño del array `commands`
2. Define nuevos pines para los LEDs adicionales
3. Actualiza las funciones `activateLED` y `recognizeCommand`