**Visvesvaraya National Institute of Technology, Nagpur**

Department of Electronics and Communication Engineering

# IoT and Embedded Systems

Arvia Project Report: A Pet AI Assistant Robot

Prepared by:
**Polastee Bhoi (BT23EEE042)**

Submitted to:
**Dr. Ankit A. Bhurane**

Assistant Professor, Department of ECE

November 17, 2025

# Arvia: Gemini AI Voice Assistant (ESP32)

## Project Overview and Objectives

### Project Name

Arvia: Gemini AI Voice Assistant (ESP32)

### Project Goal

The primary objective of this project was to develop a compact, expressive, and network-flexible embedded system capable of providing natural language interaction using modern large language models (LLMs). The device serves as a personalized voice assistant, leveraging a single OLED display for both expressive visual feedback and text responses.

### Key Objectives Achieved

- Successful integration of a microcontroller (ESP32) with external services (Deepgram, Gemini API) over WiFi.
- Implementation of a reliable audio pipeline for recording user voice via an I2S microphone (INMP441) and saving it to the internal SPIFFS filesystem.
- Development of a responsive, single-screen user interface featuring animated facial expressions and clear text output on SSD1306 OLED.
- Incorporation of a flexible WiFi Manager web portal for easy configuration without hardcoding credentials.

## Tools and Components Used

### Hardware Components

Table 1: Hardware Components List

| Component | Purpose | Details (Reference GPIO) |
|---|---|---|
| Microcontroller | Core processing and networking | ESP32 Dev Board |
| I2S Microphone | Audio input for voice commands | INMP441 Digital MEMS Mic (GPIO 26, |
| OLED Display | Visual feedback/output | SSD1306 128x64 OLED Display (I2C) |
| Record Sensor | Triggers audio recording | TTP223 Capacitive Touch (GPIO 13) |
| Mode/Reset Sensor | Toggles AI mode/resets WiFi setup | TTP223 Capacitive Touch (GPIO 12) |
| Expression Sensor | Triggers emotion animations | TTP223 Capacitive Touch (GPIO 14) |

### Software and API Tools

- **Arduino IDE / ESP32 Core:** Firmware development and compilation environment.
- **U8g2 Library:** Driver and graphics utilities for SSD1306 OLED display.

- **WebServer & SPIFFS:** WiFi Manager web portal and persistent storage for credentials.
- **ArduinoJson:** Parsing JSON responses from Deepgram and Gemini APIs.
- **Deepgram API:** Speech-to-Text (STT): Transcribing recorded WAV files into text.
- **Gemini API:** Natural Language Processing (NLP): Generating concise answers from user queries.

## System Schematic (Connections)

### Connection Diagram

```
ESP32 Pinout Connections:
=========================
INMP441 Microphone:
  - L/R (WS)   -> GPIO 27
  - DIN (SD)   -> GPIO 35
  - BCK (SCK)  -> GPIO 26
  - 3.3V       -> 3.3V
  - GND        -> GND

SSD1306 OLED (I2C):
  - SDA        -> GPIO 21
  - SCL        -> GPIO 22
  - VCC        -> 3.3V
  - GND        -> GND

Touch Sensors:
  - Record     -> GPIO 13
  - Mode/Reset -> GPIO 12
  - Expression -> GPIO 14
  - VCC        -> 3.3V
  - GND        -> GND
```

### I2S Microphone Connections (INMP441)

- Word Select (I2S_WS/LR): GPIO 27
- Data (I2S_SD/DIN): GPIO 35
- Bit Clock (I2S_SCK/BCK): GPIO 26

### OLED Display Connections (SSD1306 I2C)

The SSD1306 OLED display connects to the standard I2C bus pins (GPIO 21 for SDA and GPIO 22 for SCL). The single display handles both facial expressions and text output through dynamic state management.

### Touch Sensor Connections

- Record Trigger: Connected to GPIO 13
- Mode/Reset Toggle: Connected to GPIO 12
- Expression Control: Connected to GPIO 14

## Basic Theory of Operation

The Arvia voice assistant operates across four core functional phases:

### A. Network Setup (WiFi Manager)

On startup, the device attempts to load saved WiFi credentials from the SPIFFS internal flash memory. If unsuccessful, or if a long press on the Mode/Reset button is detected, the device enters Access Point mode (SSID: `arvia`, Password: `adminpola1`) and hosts a lightweight WebServer to capture new WiFi details. This configuration portal allows users to connect to the ”arvia” network and access a web interface at `192.168.4.1` to enter their WiFi credentials, ensuring portability and easy reconfiguration without code modifications.

### B. Input Processing (Recording and STT)

- **Recording:** Touching the Record Sensor (GPIO 13) initiates audio capture using the ESP32’s I2S driver with the INMP441 microphone. The audio data is buffered, amplified (`GAIN_BOOSTER_I2S = 10`), formatted into a WAV file, and stored temporarily on the SPIFFS.
- **Transcription:** The stored WAV file is then streamed to the Deepgram API via a secure HTTPS connection. Deepgram performs the Speech-to-Text conversion and returns the transcribed text.

### C. AI Response and Output

- **NLP:** The transcribed text is sent to the Gemini 2.0 Flash API. A system prompt ensures the response is concise (max 15 words).
- **Display:** The short text response from Gemini is then displayed on the SSD1306 OLED in the `ANSWER_DISPLAY` state, while the display switches from expressive animations to show the text response.

### D. Expressive Animation and Idle Management

The device dynamically manages its visual state based on user input and activity timers using the single SSD1306 display:

- **Interaction:** Short taps and long presses on the Expression Sensor (GPIO 14) trigger ANGRY or HAPPY animations, respectively.
- **Inactivity:** If the device receives no interaction for 25 seconds (`IDLE_TIMEOUT_MS`), it enters a SAD animation for 10 seconds, followed by a Clock Display mode to conserve power and provide ambient utility.

## Visual Representation (Conceptual)

The following figure illustrates the expected physical arrangement and single-screen concept:

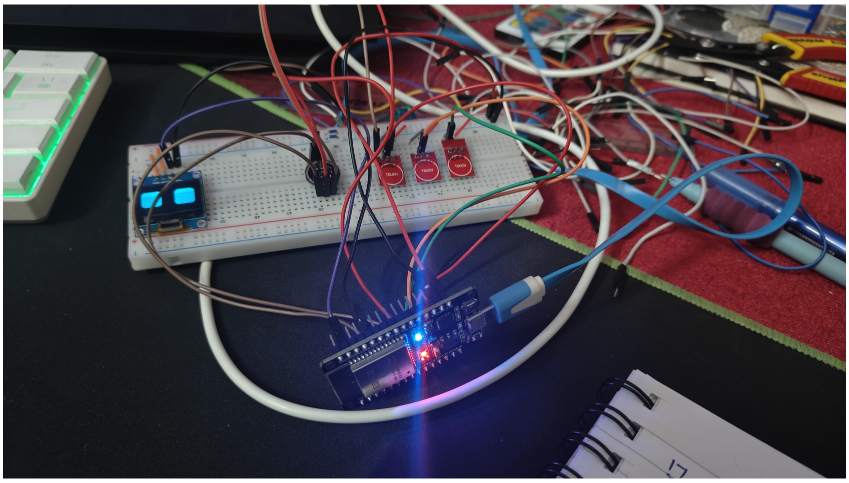

Figure 1: Arvia Device Concept with Single OLED Display

## GitHub Repository

The complete source code, schematics, and documentation for this project are available at:
`https://github.com/polabhoi/arvia-petbot/tree/main`

## Conclusions and Future Usage

The Arvia AI Voice Assistant project successfully integrates complex cloud AI services with a lightweight embedded device, offering an interactive and emotionally engaging user experience through a single SSD1306 OLED display. The modular design, particularly the integrated WiFi Manager, makes the hardware easily deployable in various environments.

Future enhancements could include implementing Text-to-Speech (TTS) output to vocalize the Gemini response, adding persistent memory for chat history, and tuning the INMP441 I2S microphone gain and filtering for improved transcription accuracy in noisy environments. The platform provides a solid foundation for further experimentation in low-power, edge-AI applications.