

PiVoice ESP Assistant

Introducere

PiVoice ESP Assistant este un asistent vocal AI fizic, gandit ca un dispozitiv de birou. Utilizatorul apasa un buton, pune o intrebare cu voce, iar sistemul ii raspunde tot vocal.

- **Ce face:** asculta o comanda audio, o trimite catre un model AI online si reda vocal raspunsul prin difuzor.
- **Scopul:** un prototip functional de asistent vocal care imbina un microcontroler cu un single-board computer.
- **Ideea de la care am pornit:** voiam un dispozitiv simplu, fara tastatura sau monitor, care sa raspunda rapid la intrebari. M-am decis pentru buton fizic in loc de wake-word, ca sa nu complic prima versiune.
- **De ce este util:** acopera mai multe zone interesante pentru un proiect de PM, microcontroler real (ESP32-S3), sistem embedded Linux (Raspberry Pi 5), placa audio dedicata (ReSpeaker 2-Mics Pi HAT), comunicatie intre dispozitive, API-uri AI, speech-to-text si text-to-speech. Pentru mine este un proiect din care invat partea hardware si partea software in acelasi loc, iar pentru alti studenti poate fi un punct de start pentru proiecte similare.

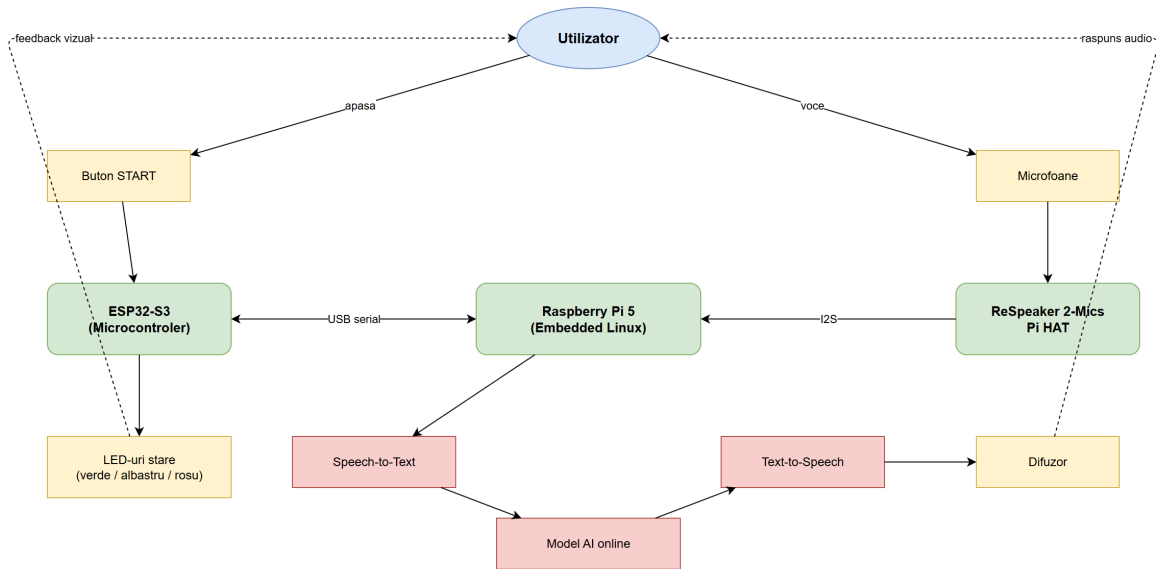
Descriere generala

Sistemul are doua zone:

- **Zona de microcontroler**, pe ESP32-S3, care se ocupa de interfata fizica: buton de start si LED-uri de stare.
- **Zona de procesare audio si AI**, pe Raspberry Pi 5 cu ReSpeaker 2-Mics Pi HAT, care se ocupa de captura audio, comunicatia cu modelul AI si redarea raspunsului.

ESP32-S3 nu proceseaza direct audio in prima varianta. ReSpeaker 2-Mics Pi HAT este facut pentru Raspberry Pi, deci e mai simplu sa lasam captura audio pe Pi. ESP32-S3 trimite catre Raspberry Pi un semnal de start si primeste inapoi starea curenta a sistemului ca sa aprinda LED-urile corespunzatoare.

Schema bloc



Pe scurt, fluxul este: utilizatorul apasa butonul de pe ESP32-S3, ESP32-S3 trimite "START_LISTEN" catre Raspberry Pi prin USB serial, Raspberry Pi inregistreaza audio de pe ReSpeaker, trimite intrebarea catre serviciul AI, primeste raspunsul, il reda prin difuzor si trimite inapoi starea catre ESP32-S3 pentru LED-uri.

Flux functional

1. Utilizatorul apasa butonul conectat la ESP32-S3.
2. ESP32-S3 trimite mesajul START_LISTEN catre Raspberry Pi 5.
3. Raspberry Pi trece in starea LISTENING si inregistreaza audio prin ReSpeaker.
4. Fisierul audio este salvat temporar pe Pi.
5. Audio-ul este transcris in text (speech-to-text).
6. Textul este trimis catre un model AI.
7. Raspunsul text este transformat in voce (text-to-speech).
8. Raspberry Pi reda raspunsul prin iesirea audio.
9. Raspberry Pi trimite catre ESP32-S3 starea finala (IDLE sau ERROR).
10. ESP32-S3 actualizeaza LED-urile.

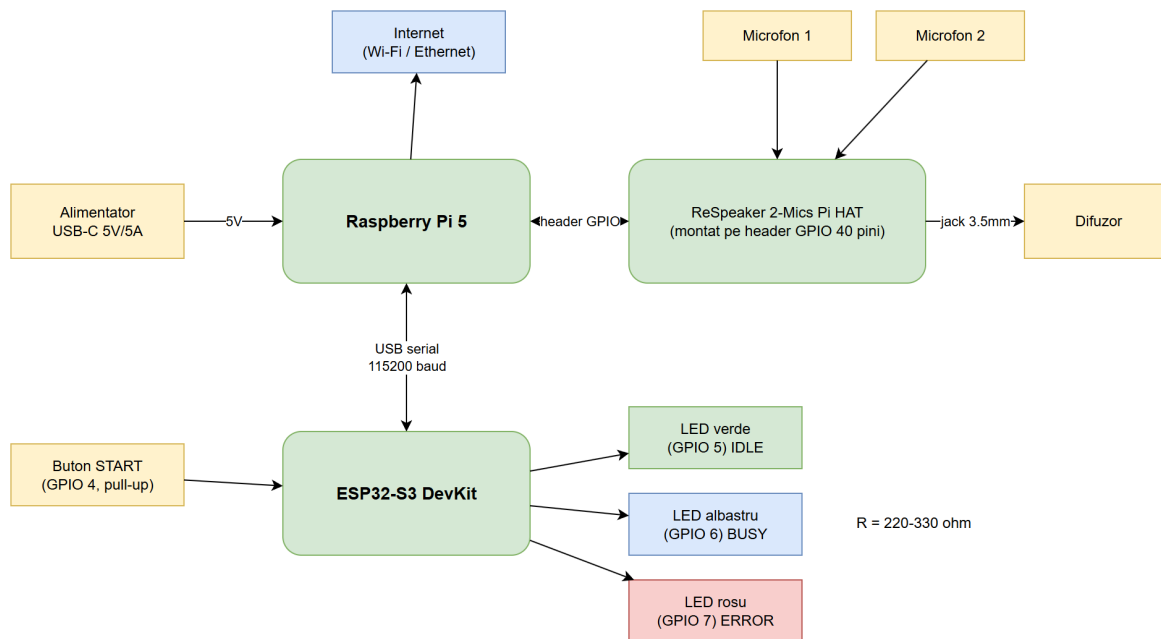
Hardware Design

Lista de componente

Componenta	Rol in proiect
Raspberry Pi 5	Single-board computer, ruleaza aplicatia principala in Python
ESP32-S3 DevKit	Microcontrolerul proiectului, citeste butonul si aprinde LED-urile
ReSpeaker 2-Mics Pi HAT	Placa audio cu 2 microfoane, codec si iesire audio
Card microSD 32/64 GB	Stocare pentru Raspberry Pi OS si aplicatie
Alimentator USB-C 5V/5A	Alimentare Raspberry Pi 5

Cablu USB pentru ESP32-S3	Alimentare si comunicatie seriala cu Raspberry Pi
Buton extern (push-button)	Pornirea unei sesiuni vocale, pe ESP32-S3
3 LED-uri (verde, albastru, rosu)	Indicare stare: idle, ascultare/procesare, eroare
Rezistente 220-330 ohm	Limitare curent pentru LED-uri
Difuzor sau boxa	Redarea raspunsului vocal
Fire Dupont, breadboard	Conexiuni de prototip pe ESP32-S3

Schema electrica



Conexiuni

Raspberry Pi 5 - ReSpeaker HAT. ReSpeaker se monteaza direct pe headerul GPIO de 40 pini. Audio se transfera prin I2S, configurarea codec-ului se face prin I2C si este gestionata de driverul din Linux.

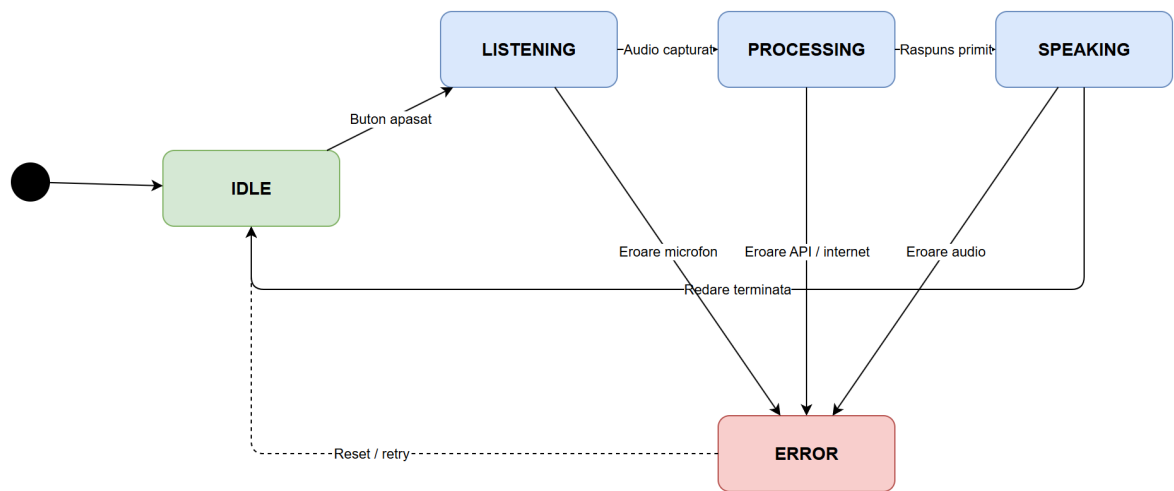
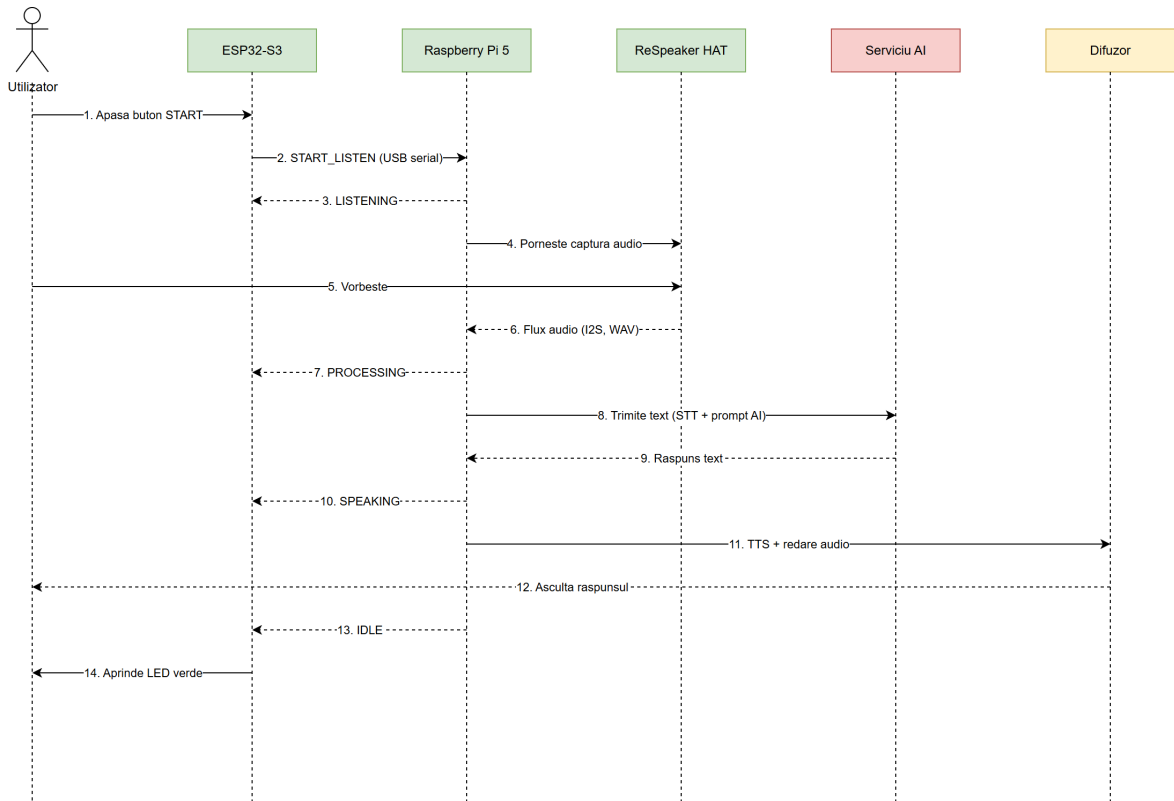
ESP32-S3 - Raspberry Pi 5. In prima versiune folosesc USB serial. ESP32-S3 se conecteaza cu un cablu USB la unul din porturile Pi-ului si este vazut ca `/dev/ttyACM0` sau `/dev/ttyUSB0`. Avantaj: nu trebuie adaptare de nivel logic, iar ESP-ul este si alimentat prin acelasi cablu.

Buton si LED-uri pe ESP32-S3:

Componenta	Pin ESP32-S3	Observatie
Buton START	GPIO 4	pull-up intern
LED verde	GPIO 5	stare IDLE
LED albastru	GPIO 6	stare LISTENING / PROCESSING
LED rosu	GPIO 7	stare ERROR
GND comun	GND	comun pentru buton si LED-uri

Diagrame de semnal

Secventa de declansare vocala (sequence diagram) si diagrama de stari a sistemului sunt mai jos.



Starile principale ale sistemului: **IDLE**, **LISTENING**, **PROCESSING**, **SPEAKING**, **ERROR**. Tranzitiile sunt declansate de butonul fizic si de mesajele primite de la modulele software (audio gata, raspuns primit, redare terminata, eroare).

Software Design

Medii de dezvoltare

Componenta	Mediu folosit
Raspberry Pi 5	Raspberry Pi OS 64-bit
Aplicatie Raspberry Pi	Python 3
ESP32-S3	Arduino IDE sau PlatformIO
Comunicatie ESP32-S3 - Raspberry Pi	USB serial (115200 baud)
Version Control	Git + GitHub

Software pe Raspberry Pi 5

Aplicatia principala este in Python. Asculta comenzile venite de la ESP32-S3 pe portul serial, controleaza fluxul audio si comunica cu serviciul AI.

Module software propuse:

Fisier	Rol
main.py	orchestrare aplicatie
serial_bridge.py	comunicatie cu ESP32-S3 prin USB serial
audio_recorder.py	inregistrare audio de pe ReSpeaker
stt_client.py	speech-to-text
ai_client.py	trimitere intrebare catre modelul AI
tts_client.py	text-to-speech
audio_player.py	redare raspuns audio
config.py	parametri aplicatie
logger.py	jurnalizare evenimente si erori

Biblioteci Python:

Biblioteca	Rol
pyserial	comunicatie seriala cu ESP32-S3
sounddevice	captura audio
soundfile	salvare fisiere WAV
openai	API AI, STT si TTS
requests	comunicatie HTTP, daca este nevoie
python-dotenv	incarcare chei API din fisier .env

Software pe ESP32-S3

Firmware scris in Arduino IDE. Functiile principale:

- initializeaza GPIO-urile pentru buton si LED-uri;

- detecteaza apasarea butonului si trimite START_LISTEN pe serial;
- citeste mesaje de la Raspberry Pi (IDLE, LISTENING, PROCESSING, SPEAKING, ERROR);
- aprinde LED-urile in functie de stare.

Protocol de comunicatie

Protocol text simplu peste USB serial, un mesaj pe linie.

De la ESP32-S3 catre Raspberry Pi:

Comanda	Semnificatie
START_LISTEN	utilizatorul a apasat butonul
CANCEL	anulare sesiune curenta (optional)
PING	test comunicatie

De la Raspberry Pi catre ESP32-S3:

Mesaj	Semnificatie
IDLE	sistemul este gata
LISTENING	sistemul inregistreaza
PROCESSING	se proceseaza intrebarea
SPEAKING	se reda raspunsul
ERROR	a aparut o eroare

Exemplu de schimb:

```
ESP32-S3 -> Pi: START_LISTEN  
Pi -> ESP32-S3: LISTENING  
Pi -> ESP32-S3: PROCESSING  
Pi -> ESP32-S3: SPEAKING  
Pi -> ESP32-S3: IDLE
```

Algoritmul aplicatiei pe Raspberry Pi

```
initializare_audio()  
initializare_serial()  
trimite_catre_esp32("IDLE")  
  
while True:  
    comanda = citeste_serial()  
    if comanda == "START_LISTEN":  
        trimite_catre_esp32("LISTENING")  
        inregistreaza_audio("cmd.wav")  
        trimite_catre_esp32("PROCESSING")  
        text = transcrie_audio("cmd.wav")
```

```
raspuns = trimite_catre_ai(text)
trimite_catre_esp32("SPEAKING")
genereaza_audio(raspuns, "reply.mp3")
reda_audio("reply.mp3")
trimite_catre_esp32("IDLE")
```

Algoritmul firmware ESP32-S3

```
void setup() {
  Serial.begin(115200);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
  pinMode(LED_GREEN, OUTPUT);
  pinMode(LED_BLUE, OUTPUT);
  pinMode(LED_RED, OUTPUT);
  setIdle();
}

void loop() {
  if (buton_apasat()) {
    Serial.println("START_LISTEN");
  }
  if (Serial.available()) {
    String mesaj = Serial.readStringUntil('\n');
    actualizeaza_leduri(mesaj);
  }
}
```

Rezultate Obtinute

Concluzii

Download

Jurnal

Bibliografie/Resurse

Resurse Hardware

- Raspberry Pi 5: <https://www.raspberrypi.com/products/raspberry-pi-5/>
- Seeed Studio ReSpeaker 2-Mics Pi HAT: https://wiki.seeedstudio.com/ReSpeaker_2_Mics_Pi_HAT/
- ReSpeaker 2-Mics Pi HAT V2: https://wiki.seeedstudio.com/respeaker_2_mics_pi_hat_raspberry_v2/
- Espressif ESP32-S3: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32s3/>
- ESP32-S3 I2S: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32s3/api-reference/peripherals/i2s.html>

Resurse Software

- Raspberry Pi OS: <https://www.raspberrypi.com/documentation/computers/os.html>
- Python 3: <https://docs.python.org/3/>
- pyserial: <https://pyserial.readthedocs.io/>
- sounddevice: <https://python-sounddevice.readthedocs.io/>
- OpenAI API (audio): <https://platform.openai.com/docs/guides/audio>
- Whisper: <https://github.com/openai/whisper>
- Piper TTS: <https://github.com/rhasspy/piper>

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2026/florin.stancu/matei.stroescu>



Last update: **2026/05/13 03:44**