

# **Smart locking system**

Oprea George Bogdan(SSAI)  
Orjanu Oana Alexandra(SSAI)  
Pavelescu Roxana(SSAI)

## Introducere

Proiectul "Smart locking system" își propune să realizeze un sistem de închidere/deschidere a ușii bazat pe recunoașterea vorbitorului. Practic, sistemul își propune să înlocuiască actualul mod de deschidere/închidere bazat pe chei, cu un sistem automat acționat pe bazat recunoașterea vocii.

## Arhitectura

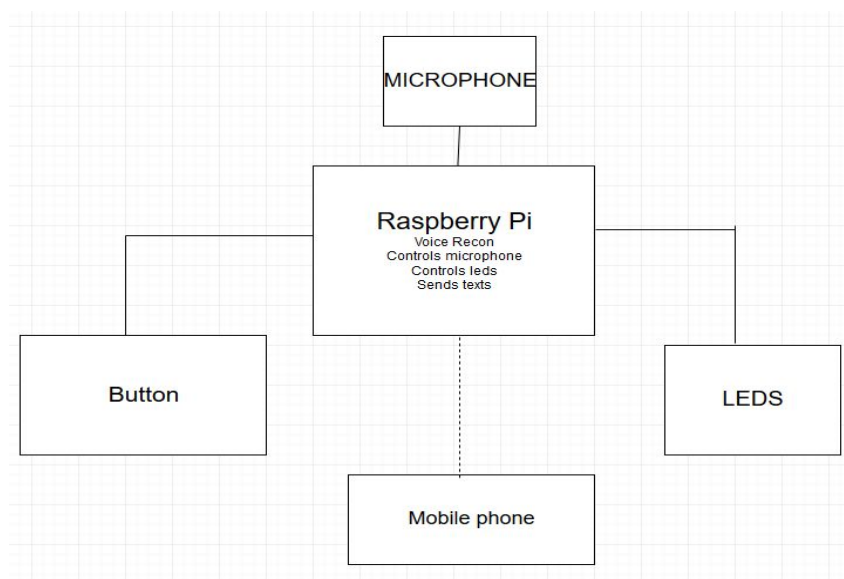


Fig. 1 Arhitectura proiectului

Așa cum se vede și din figura 1 proiectul conține: un buton, leduri, un microfon, un telefon și o placuță Raspberry Pi care controlează și acționează asupra întregului sistem. Placuța pornește algoritmul de recunoaștere vocală atunci când butonul de la intrare este apăsat, având activat microfonul în care utilizatorul trebuie să vorbească. În urma rulării recunoașterii vocale, placuța acționează apoi cele 2 leduri și telefonul mobil: aprinde unul din cele două leduri ca răspuns la faptul că persoana a fost sau nu recunoscută ca fiind de încredere și trimite un mesaj pe telefonul mobil tot în acest sens.

Microfonul este conectat la un port usb al Raspberry-ului, iar ledurile și butoanele sunt conectate la pini de pe o plăcuță auxiliara.

Telefonul mobil nu are o conexiune fizică cu plăcuța, aceasta trimițând mesaj către telefon folosind numărul de telefon.

## Implementare

Implementarea sistemului de smart locking este realizat în Python folosind diverse biblioteci.

### 1) Recunoașterea vocală

Bucata de recunoaștere vocală este realizată folosind biblioteca **piwho**. Acesta este un wrapper peste framework-ul de recunoaștere a vorbitorului MARF<sup>1</sup> pentru Raspberry Pi. Pentru recunoașterea vocală, algoritmul folosește pentru recunoaștere un model antrenat și date dintr-un fișier speakers.txt pentru a putea recunoaște persoana.

Pentru antrenarea modelului - datele se preprocesează și apoi se extrag trăsăturile folosind FFT<sup>2</sup> și LPC<sup>3</sup> iar transmit apoi vectorii sunt agregați și trimiși către rețelele de antrenare. Antrenarea folosește un singur net pentru toți vorbitorii.

Pentru clasificare, merit să pună o etichetă seturilor de date, folosește distanța Mahalanobis<sup>4</sup>. Procesul de antrenare este făcut folosind back propagation learning<sup>5</sup>.

Etape de recunoaștere folosește extragerea de trăsături(cu ajutorul LPC), după care se găsește media fundamentală a frecvenței de enunț. Rezultatele sunt concatenate pentru a forma vectorul de răspuns. Clasificatorul ia în considerare ponderea trăsăturilor - rețeaua face asta implicit dacă ajută match-ul cu un vorbitor, și stochastic poate fi modificat să dea o pondere mai mare frecvenței F0.

## 2) Trimiterea de mesaje către telefon

Trimiterea de mesaje către telefon este făcută folosind librăria **Twilio**<sup>6</sup>. Cu ajutorul acesteia practic atunci când se dorește trimiterea unui mesaj se face un request de tip HTTP către numărul care îi este acordat utilizatorului de pe plăcuță, iar în urma acestui request HTTP serviciul trimite apoi mesaj către numărul de telefon indicat în request.

Twilio oferă suport pentru numeroase limbaje de programare între care și Python, cel folosit.

## 3) Integrarea în tot sistemul

Pentru a folosi bibliotecile menționate mai sus și a integra totul în aplicația de smart locking, au fost necesari mai mulți pași:

- înregistrarea de utilizatori pentru antrenarea algoritmului - pentru aceasta s-a realizat scriptul script.sh care înregistrează 5 mesaje și crează un folder separat al utilizatorului
- rularea algoritmului de recunoaștere la apăsarea butonului - aceasta parte este făcută din recon.py, care la apăsarea butonului pornește mai întâi un script care înregistrează un mesaj al utilizatorului și apoi prelucrează fișierul salvat wav încercând să recunoască utilizatorul, iar în funcție de răspuns aprinde unul din becuri și trimite un mesaj.
- algoritmul de recunoaștere întoarce un scor de încredere, iar în funcție de acesta se stabilește dacă persoana este trusted sau nu

Pentru înregistrarea fișierului .wav s-a folosit comanda **arecord**. Frecvența de înregistrare a mesajelor este 16000 Hz pe un sampling de 16 biți little endian (frecvența este aleasă conform funcționării microfonului). Mesajul inițial este apoi transformat la o frecvență de 8000 de Hz deoarece piwho are nevoie la intrare de fișiere wav de frecvență de 8000 Hz.

---

<sup>1</sup> Modular Audio Recognition Framework - <http://marf.sourceforge.net/>

<sup>2</sup> FFT - FFT Feature Extraction

<sup>3</sup> LPC - Linear Predictive Coding

<sup>4</sup> [https://en.wikipedia.org/wiki/Mahalanobis\\_distance](https://en.wikipedia.org/wiki/Mahalanobis_distance)

<sup>5</sup> <https://en.wikipedia.org/wiki/Backpropagation>

<sup>6</sup> <https://www.twilio.com/docs/sms/send-messages>

## Utilizare

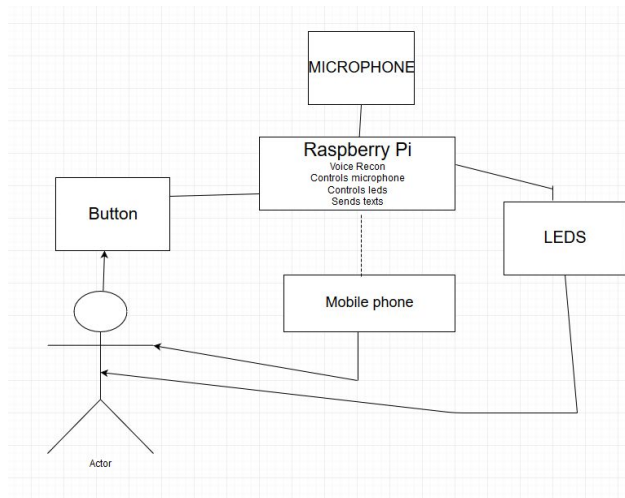


Fig. 2 Utilizarea sistemului

Pașii de utilizare sunt următorii

- 1) utilizatorul apasă butonul
- 2) utilizatorul vorbește în microfon
- 3) raspberry pi procesează .wav-ul făcut de utilizator rulând recunoașterea vocală
- 4) utilizatorul află dacă a fost recunoscut privind cele 2 leduri - led-ul verde se aprinde dacă utilizatorul e recunoscut, iar cel roșu dacă utilizatorul nu e recunoscut
- 5) persoana care are configurată numărul de telefon află dacă utilizatorul a fost recunoscut și cu ajutorul telefonului mobil - primește mesaj dacă a fost recunoscut sau nu - utilizatorul poate afla astfel toate persoanele care sunt lăsate să intre în casă sau dacă cineva necunoscut s-a aflat la ușă

## Concluzii

Proiectul smart locking system este momentan doar un proof of concept. Pentru a putea înlocui efectiv sistemul de deschidere sunt necesare mai multe teste pentru a vedea că sistemul de recunoaștere vocală funcționează cum trebuie, dar în viitor credem că acesta va înlocui sistemul cu cheie existent momentan.