

**Universitatea
Transilvania
din Brașov**

**FACULTATEA DE INGINERIE ELECTRICĂ
ȘI ȘTIINȚA CALCULATOARELOR**

Home Assistant

Arhitectura și organizarea calculatoarelor

Autor:

Dinu Ionuț Vlăduț

Calculatoare anul IV

Grupa 4LF781

ionut.dinu@student.unitbv.ro

BRAȘOV, 2021

Continut

1. [Rezumat](#)
 - 1.1. Descrierea proiectului
 - 1.2. Descrierea implementarii
2. [Introducere](#)
 - 2.1. Prezentarea temei
 - 2.2. Motivatia alegerii temei
 - 2.3. Sisteme/Implementari existente pe piata
 - 2.3.1. HomeKit
 - 2.3.2. Google Assistant
3. [Prezentarea platformei hardware](#)
 - 3.1. Raspberry Pi 4
 - 3.2. Electronica folosita
 - 3.2.1. DHT11
 - 3.2.2. MQ-2
 - 3.2.3. Bec
 - 3.3. Alimentarea echipamentului hardware
4. [Diagrama concept](#)
5. [Schema bloc](#)
6. [Costurile proiectului](#)
7. [Prezentarea implementarii software](#)
 - 7.1. Limbajul Python
 - 7.2. Biblioteca GPIO-zero
 - 7.3. Voice Recognition
 - 7.4. Framework-ul Angular
 - 7.5. Framework-ul Flask
 - 7.6. InfluxDB
8. [Performante](#)
9. [Concluzii](#)
10. [Dezvoltari ulterioare](#)
11. [Bibliografie](#)
12. [Anexa](#)

1. Rezumat

1.1. Descrierea proiectului

Acest proiect are in vedere realizarea unei ustensile atat hardware cat si software pentru monitorizarea mediului dintr-o camera folosindu-se de o platforma web, construita de la zero, si de senzorii montati si legati la un WeMos D1 Mini care va fi un server si cu care va comunica placuta de Raspberry Pi 4. Datele de la senzori vor fi persistate intr-o baza de data online, InfluxDB. Mai mult decat atat, va fi implementat un modul de recunoastere de voce care va actiona un motor si led-urile montate pe pini diferiti prin diferite cuvinte cheie.

Platforma web creata va fi o punte de comunicare intre placuta Raspberry si utilizator, loc in care se pot observa datele de la senzori si aprinderea led-urilor fiecarei camera.

1.2. Descrierea implementarii

Pentru ca proiectul sa fie inteles pe deplin acesta va fi realizat in urmatoarele moduri:

1. Se vor monta senzorii si se vor aloca pini diferiti pentru fiecare tip de echipament hardware
2. Se va prelua date de la senzorii montati
3. Se vor conecta casti USB in portul alocat
4. In cazul in care de la casti se va recunoaste unul din mesajele prestabilite pentru diferitele comenzi stabilite, acestea se vor executa, ca de exemplu la comanda "open the lights" un bec va fi alimentat si aprins
5. Datele preluate de la senzori vor fi trimise catre o baza de date unde acestea vor fi persistate
6. Pagina web va prelua aceste date si le va afisa utilizatorului

2. Introducere

2.1 Prezentarea temei

În cadrul realizării acestui proiect, tinta este de a crea un Home Assistant care poate fi refolosit de oricine dorește implementarea unui astfel de utilitar atât pentru casa cât și pentru diferite alte locații.

2.2 Motivatia alegerii temei

În prezent, casele devin tot mai inteligente iar oamenii doresc cât mai mult integrarea utilităților de genul pentru a-și face viața mai ușoară și pentru a avea o monitorizare asupra mediului de trai cât și pentru a economisi resursele plantei pentru a reduce costurile de trai și de a polua mai puțin. Prezenta abundență a acestor utilități este din ce în ce mai accentuată însă fiecare vine cu diferite pachetări pentru a atrage sume cât mai mari de la client atât prin serviciile lor cât și prin echipamentele hardware. Noi intenționăm a crea un utilitar open-source, care să fie folosit și la care să se poată adăuga diferite module personalizabile atât de către client cât și de partenerii care doresc să ajute la implementare. O caracteristică importantă pe care o prezintă proiectul este faptul că se poate încadra într-o investiție mică dar cu performanțe pe o durată lungă de timp.

2.3 Sisteme/Implementări existente pe piață

2.3.1 HomeKit



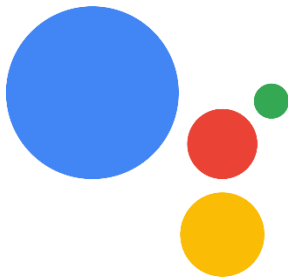
HomeKit este soluția pe care cei de la Apple au pus-o la dispoziția oamenilor pentru a-și crea o casă inteligentă, folosindu-se de echipamentele pe care le vând. Acestea pot varia

de la becuri inteligente, la prize, termostate sau chiar si jaluzele, toate actionate prin aplicatia lor care este prezenta pe platforma IOS.

Prin intermediul acestei aplicatii se pot crea planuri configurabile pentru diferite momente ale zilei. De exemplu, noaptea lumina din camere sa nu depaseasca o anumita valoare de luminozitate, poarta si usile casei sa fie inchise si sa porneasca alarma.

Aceste dispozitive inteligente se conecteaza la aplicatie direct prin Wi-Fi sau Bluetooth sau la un Hub care apoi este conectat la platforma.

2.3.2 Google Assistant



Google Assistant, la fel ca si HomeKit, poate conecta diferite dispozitive inteligente care sunt compatibile cu acest software. O diferenta importanta intre cele 2 este ca Google Assistant exista pe toate platformele de mobil, nu doar IOS.

Un aspect interesant al acestui produs este faptul ca fiecare dispozitiv poate fi clasat pe mai multe categorii. De exemplu, o grupare de becuri pot face parte din grupul 1, care se afla in camera X din apartamentul F1. Astfel, se pot comanda separate toate dispozitivele conectate prin numele lor sau prin diferitele grupari ca in exemplul mentionat anterior.

O implementare interesanta pe care cei de la Google au pus-o in functiune este starea de functionare a acestui produs. Google Assistant este un dispozitiv cu microfon si difuzor care foloseste procesare audio pentru activare. Astfel ca, atunci cand este pornit el ramane in starea de stand-by pana cand aude cuvintele "Hey Google" sau "Ok Google", iar apoi face tranzitia in starea de functionare normala.

3. Prezentarea platformei folosite

3.1 Raspberry PI 4

Raspberry PI 4 este un asa zis SBC (Single Board Computer), adica un calculator pe o placuta. Acesta prezinta atat un CPU cat si memorie RAM, memorie FLASH, 40 pini GPIO cat si 4 porturi USB si 2 porturi micro-HDMI. Arhitectura procesorului este ARM ceea ce il face sa foloseasca ca si sistem de operare o distributie Linux numita Raspberry PI OS.

3.2 Electronica folosita

3.2.1 DHT11

DHT11 este un senzor care masoara atat temperatura cat si umiditatea. Este o varianta de senzor care nu prezinta o acuratete destul de buna pentru uz intens insa este perfect pentru un proiect de IoT de inceput pentru ca este un senzor universal. Acesta prezinta vine in diferite configuratii de pini. Un exemplu ar fi cel de 4 pini care prezinta 2 pini de alimentare, unul de date si unul care nu se conecteaza. Un alt exemplu este cel de 3 pini care functioneaza pe interfata I2C si care are 2 pini de alimentare si unul de date. Ca si specificatii tehnice, acesta este potrivit pentru masuratori ale umiditatii intre 20-80% si a temperaturii intre 0 si 50 °C.

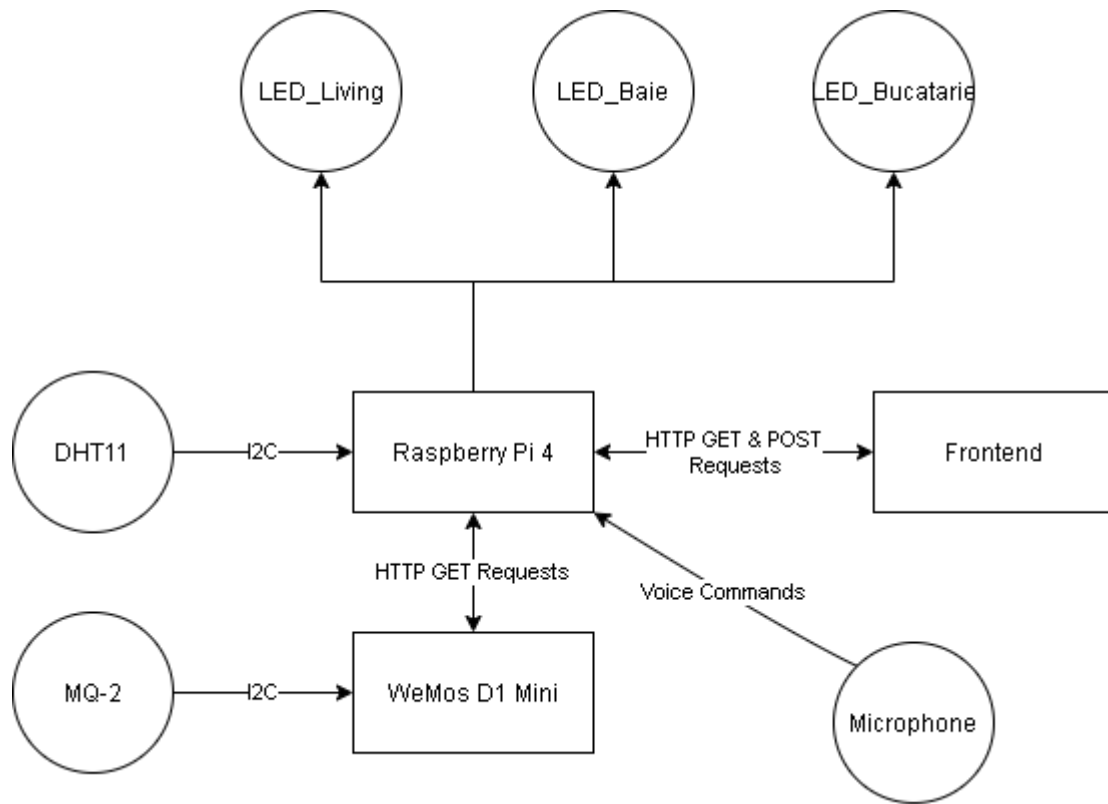
3.2.2 MQ-2

MQ-2 este un senzor care masoara cantitatea de gaz si fum. Acesta prezinta caracteristici care-l fac sa fie un senzor eficient din punct de vedere al raportului performanta : cost. Are o sensibilitate buna si o raza mare de masura pentru gazele detectate. Materialul senzitiv este oxid de staniu care prezinta o conductivitate mica in aer curat. Atunci cand sunt prezente gaze in prejurul acestui senzor, conductivitatea sa creste.

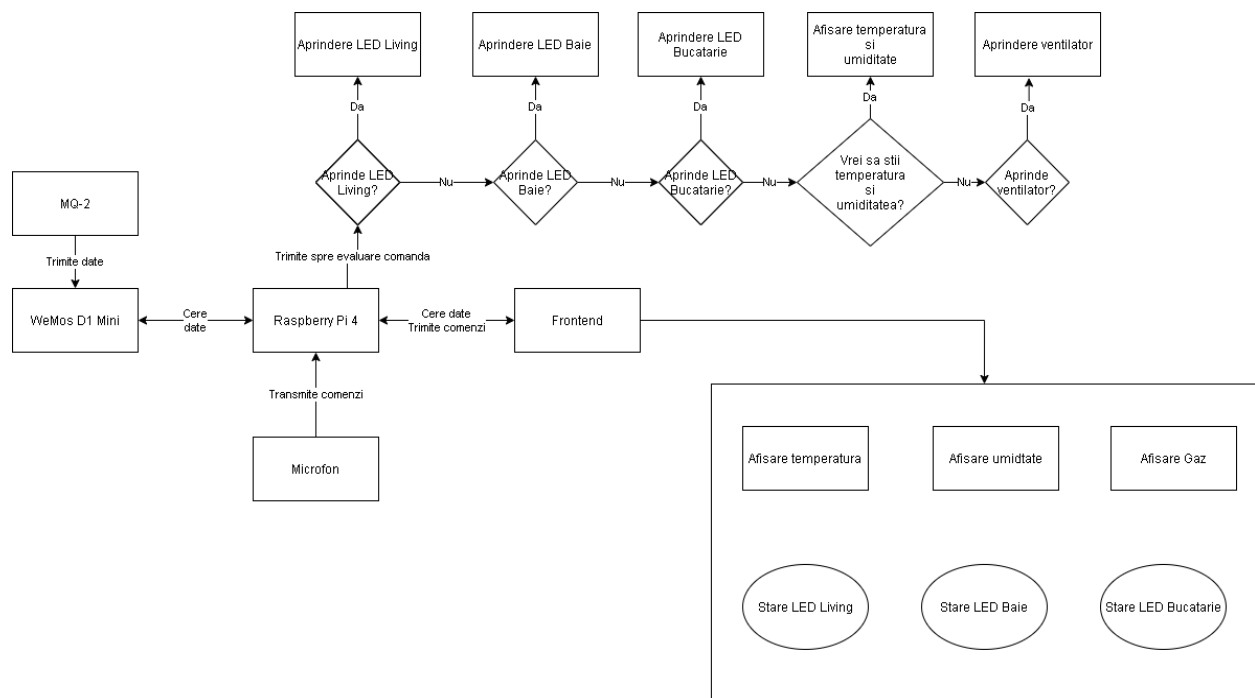
3.2.3 Led

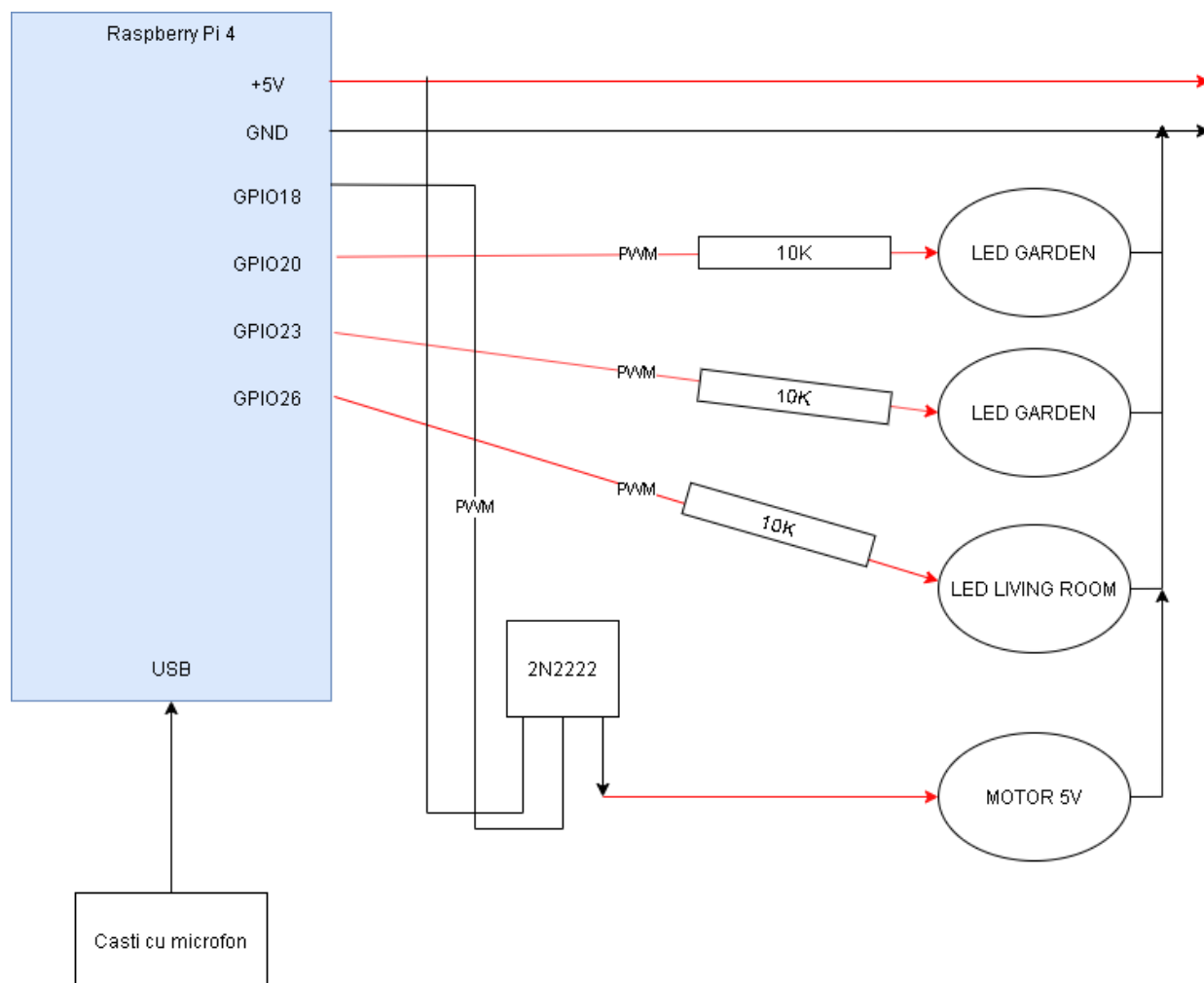
Led-ul este o diode semiconductoare care emite lumina atunci cand i se aplica o tensiune de alimentare. Este folosit in industrie pentru a confirma starile unei parti a unui circuit. De exemplu led-urile unu router care semnaleaza faptul ca exista conexiune la internet.

4. Diagrama concept



5. Schema bloc





6. Costurile proiectului

Dispozitiv	Pret	
Raspberry Pi 4 4GB	430 Lei	
MQ-2	12.5 Lei	
DHT11	10 Lei	
3x Led	1 Leu	
Breadboard + fire	10 Lei	
WeMos D1 Mini	50 Lei	
Total		513.5 Lei

7. Prezentarea implementarii software

7.1 Limbajul Python

Python este un limbaj scriptic care are nevoie de un interpretor pentru a functiona. Acesta este un limbaj de uz general folosit de la aplicatii simple pana la aplicatii de uz industrial. Este un limbaj favorit in randul programatorilor care lucreaza in domeniul inteligentei artificiale.

7.2 Biblioteca GPIO-zero

Biblioteca GPIO-zero reprezinta o simpla interfata intre dispozitivele GPIO, conectate la Raspberry Pi, si cod. Aceasta biblioteca ajuta programatorii sa scrie un cod intr-un mod mai usor de inteles, venind cu diferite clase si metode foarte usor de inteles. De exemplu pentru a aprinde un led atunci cand un buton este apasat se va executa urmatoarea secventa de cod:

```
from gpiozero import LED, Button
from signal import pause

led = LED(17)
button = Button(3)

button.when_pressed = led.on
button.when_released = led.off
```

7.3 Voice Recognition

Pentru procesare si recunoastere a comenzilor a fost folosita o biblioteca de Python numita SpeechRecognition. Aceasta are mai multe modele folosibile pentru recunoastere vocala. Modelul folosit in cadrul proiectului se numeste Google Speech Recognition.

Functionarea lui are la baza calibrarea sensibilitatii la sunetul de fundal, iar dupa acest proces, asculta continuu si inregistreaza orice si dupa o mica pauza acesta incepe sa proceseze si sa afiseze ceea ce s-a spus pana in momentul acela. Pentru a da comenzi catre Raspberry Pi s-a utilizat limba engleza pentru o acuratete mai mare a acestei biblioteci.

7.4 Framework-ul Angular

Angular este un framework al limbajului de programare TypeScript si este folosit in industrie ca o metoda de creare a aplicatiilor de mobil, dar si a paginilor web, atat cele statice cat si cele dinamice. Acest framework a fost dezvoltat de catre o echipa din cadrul companiei Google. Cu acest framework a fost creata interfata din cadrul paginii web a proiectului.

7.5 Framework-ul Flask

Flask este un "micro-web" framework al limbajului de programare Python. Este un framework light si usor de folosit deoarece nu are foarte multe functionalitati precum legaturi cu baza de date folosind modele sau validari de date. Prin intermediul acestui framework s-a creat API-ul proiectului care asculta si preia comenzile predefinite pe care i le trimitem.

7.6 InfluxDB

InfluxDB este o baza de date open-source folosita pentru stocarea datelor de tip "time-series". Este folosita in industrie pentru foarte multe proiecte de IoT. In cadrul acestui proiect, InfluxDB a fost folosita pentru stocarea datelor.

8. Performante

Performantele aplicatiei sunt destul de comparabile cu implementarile existente pe piata insa are niste probleme datorita platformei Web pe care o hosteaza, consumandu-i o mare parte a memoriei RAM. Astfel ca un model de Raspberry Pi 4 cu 8GB RAM ar fi mult mai potrivita pentru o astfel de implementare. Server-ul Python care comunica cu placuta Arduino si cu pagina Web se comporta normal, cu performante inalte si timp minim de raspuns.

Pe de alta parte, pagina Web poate fi pornita pe un alt server si se poate comunica cu placuta Raspberry pentru a diminua memoria RAM si adaugandu-. mai multe resurse disponibile comunicarii intre servere.

O problema este faptul ca modulul de comunicare prin voce necesita o conexiune la internet pentru a se putea accesa API-ul de la Google pentru Voice Recognition. Timpul de raspuns pentru acest modul este in medie 3-4 secunde.

9. Concluzii

Acest proiect a dus la o concluzie foarte importanta, mai exact faptul ca se poate crea un astfel de home assistant cu costuri minime si cu performante bune. Astfel ca, incadrând toate modulele de Voice Recognition, platforma Web de control si vizionare a datelor senzorilor si controlul a altor module, se poate concluziona ideea ca proiectul, desi complex, a fost usor de implementat si a constituit un exercitiu foarte bun pentru a integra diferite module si a le face sa comunice intre ele.

10. Dezvoltari ulterioare

Acest proiect doreste sa ofere posibilitatea fiecărei persoane sa vina cu senzori si sa ii monteze si sa ii foloseasca, de aceea in viitor se doreste a se crea un framework care sa genereze automat API-ul pentru controlul dispozitivelor dar si o lista de cuvinte cheie ce trebuiesc rostite pentru a le controla si prin comanda vocala.

11. Bibliografie

<https://www.macrumors.com/guide/homekit/>

<https://medium.com/google-developers/iot-google-assistant-f0908f354681>

<https://www.the-ambient.com/how-to/add-devices-google-assistant-441>

https://en.wikipedia.org/wiki/Raspberry_Pi

<https://learn.adafruit.com/dht>

<https://www.pololu.com/file/0J309/MQ2.pdf>

<https://gpiozero.readthedocs.io/en/stable/>

<https://pypi.org/project/SpeechRecognition/>

12. Anexa

Implementarea software: <https://github.com/VladutDinu/PAOC-HomeAssistant>