

Universitatea  
Transilvania  
din Brașov  
FACULTATEA DE INGINERIE ELECTRICĂ  
ȘI ȘTIINȚA CALCULATOARELOR

*Solomon Vlad-George*

# PROIECT DE DIPLOMĂ

Conducător științific:  
Prof.Dr.Ing Ogruțan Petre-Lucian

Absolvent:  
2023-2024

BRAŞOV, 2023

**Departamentul Electronică și Calculatoare**  
**Programul de studii: Electronică Aplicată**

*Solomon Vlad-George*

# Sistem automat de prezență cu RFID și Raspberry PI

**Conducător științific:**  
Prof.Dr.Ing Ogruțan Petre-Lucian

## FIŞA PROIECTULUI DE DIPLOMĂ

Universitatea Transilvania din Brașov	Proiect de diplomă nr. .......
Facultatea Inginerie Electrică și Știința Calculatoarelor	Viza facultății
Departamentul Electronică și Calculatoare	Anul universitar <b>2023-2024</b>
Programul de studii: <b>Electronică Aplicată</b>	Promoția <b>2024</b>
Candidat <b>Vlad-George SOLOMON</b>	

Conducător științific: prof dr. ing. OGRUȚAN Petre-Lucian	
--	--

### PROIECT DE DIPLOMĂ

**Titlul lucrării: Sistem automat de prezență cu RFID și Raspberry PI**

Problemele principale tratate:

1. Analiza stadiului actual
2. Structura hardware, scheme electrice
3. Structura software, scheme logice, secvențe de program
4. Rezultate experimentale

Locul și durata practiciei: acasă și în laboratoarele departamentului

Bibliografie:

1. M. Romanca, P. Ogrutan, *Sisteme cu calculator incorporat. Aplicatii cu microcontrollere*, Ed. Universitatii Transilvania Brasov, 2011, ISBN 978-973-598-861-6
2. P. Ogrutan, *Interfatare si protocoale la nivelul fizic si nivelul legaturii de date*, Ed. Universitatii Transilvania Brasov, ISBN 978-606-19-0515-7, 2015
3. Dörte Müller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*, 3rd Edition, Wiley, ISBN: 978-0-470-69506-

Aspecte particulare: schema electrică a conexiunilor hardware, scheme logice simplificate ale programului software. Menționarea rezultatelor experimentale, fotografii ale montajelor.

Primit tema la data de: **20.06.2023**

Data predării lucrării: **27.06.2024**

Director departament,  
Şef lucr.dr.ing. STANCA Cornel Aurel.....

Conducător științific:  
Prof. dr. ing. .OGRUȚAN Petre Lucian

Candidat,  
**Vlad-George SOLOMON**

### PROIECT DE DIPLOMĂ - VIZE

Data vizei	Capitole/ problemele analizate	Semnătura cadrului didactic îndrumător
7.03.2023	Stadiul actual al cercetărilor	
4.11.2023	Concepție, proiectare, stabilire obiective	
20.03.2024	Realizare hardware și software.	 
16.05.2024	Verificare funcționalitate. Realizarea documentației	
10.06.2024	Punere la punct și rezultate experimentale	

### **APRECIEREA ȘI AVIZUL CONDUCĂTORULUI ȘTIINȚIFIC**

Criteriu evaluare (punctaj maxim)	Punctaj acordat	Argumentare
<b>1. Fond</b>		
1.1.Originalitate (10)	5	Nivel bun.
1.2.Nivel științific (25)	20	Nivel bun. A fost utilizat un modul Raspberry PI, programat cu un program propriu
1.3.Complexitate (25)	20	Complexitatea aplicației este medie

1.4. Nivel de implementare (30)	20	Nivel acceptabil. Proiectul este funcțional.
<b>Total fond (90)</b>	65	
<b>2. Formă (10)</b>	10	Forma are un nivel bun. <b>Similitudini Turnitin 5%</b>
<b>TOTAL (100)</b>	75	

Data: 27.06.2024	<b>ADMIS</b> pentru sustinere/ <b>RESPINS</b>	CADRU DIDACTIC ÎNDRUMĂTOR Prof. dr. ing. OGRUȚAN Petre Lucian
------------------	---	--

#### **AVIZUL DIRECTORULUI DE DEPARTAMENT**

Data: 27.06.2024	<b>ADMIS</b> pentru sustinere/ <b>RESPINS</b>	Director departament Şef lucr.dr.ing. STANCA Cornel Aurel ..... (semnatura)
---------------------	---	---

#### **SUSTINEREA PROIECTULUI DE DIPLOMĂ**

Sesiunea **iulie 2024**

Rezultatul sustinerii	PROMOVAT cu media:
	RESPINS <b>cu</b> refacerea lucrării
	RESPINS <b>fără</b> refacerea lucrării
PREȘEDINTE COMISIE Prof. dr. Petru Adrian COTFAS	..... (semnatura)

Cuprins

**Cuprins**

Cuprins .....	2
1 INTRODUCERE .....	8
1.1 SCOPUL PROIECTULUI .....	8
1.2 STRUCTURA LUCRĂRII .....	9
1.3 OBIECTIVE PRINCIPALE .....	10
1.3.1      Obiective hardware .....	10
1.3.2      Obiective software .....	10
2 STADIUL ACTUAL ÎN DOMENIU .....	10
3. Configurarea hardware .....	26
3.1 Componente necesare .....	26
3.2 SDA,SCK,MOSI,MISO,IRQ .....	27
4. <i>Raspberry PI 5</i> .....	33
5. Baza de date/MariaDB .....	34
6. Securitate .....	36
7. Implementare .....	38
8. Posibile implementări ulterioare .....	52
9. Concluzii .....	53
10. Bibliografie .....	56

*(figurile, tabelele și codurile sursă se vor grupa în două liste și vor fi numerotate așa cum se regăsesc în text)*

## FIGURI

- Fig.1 - Schema bloc RFID
- Fig.2 - Diagramă RC 522
- Fig.3 - Schema bloc funcționare RFID
- Fig.4 - Range-uri de frecvență RFID
- Fig.5 - Range-ul de citire în funcție de frecvență
- Fig.6 - Schemă protocol
- Fig.7 - HF system
- Fig.8 - Zebra FX7500
- Fig.9 - Specificații Zebra FX7500
- Fig.10 - Efectul unui filtru moving average
- Fig.11 - Simulare
- Fig.12 - Rezultatele simulării
- Fig.13 Conexiunile hardware
- Fig.14 Modul de citire a datelor utilizând protocolul SPI
- Fig.14.1 Modul de scriere a datelor utilizând protocolul SPI
- Fig.15 Diagrama de timp pentru SPI
- Fig.16 Adresa I2C
- Fig.17 Schema de Conexiuni a Expanderului I/O PCF8574 pentru Comunicare I2C
- Fig.18 Nivelele I2C
- Fig.19 Senzorul ultrasonic
- Fig.20 Formele de undă pentru funcționarea senzorului
- Fig.21 Specificații Raspberry
- Fig.21.1 Primele date stocate
- Fig.22 Reîncriptarea datelor pentru RFID
- Fig.23 Interfața pentru raspi-config
- Fig.24 Temperaturi Raspberry
- Fig.25 Soc error
- Fig.26 Temperaturile în momentul citirii tagurilor
- Fig.27 Grafic cu valorile monitorizării temperaturii Cpu la momente statice și la momente când tagul este detectat
- Fig.28 Timpul de citire și viteza de transfer RFID/implementare cod
- Fig.29 Grafice Timp de citire și viteza de transfer raportat la citiri
- Fig.30 Configurare hardware și teste
- Fig.31.1 Testare citire, lcd, senzor
- Fig.31.2 Testare citire, lcd, senzor
- Fig.31.3 Proiectare și testare
- Fig.32 Schemă bloc funcționare după citirea RFID

## **Lista de acronime**

*(se vor scrie toate prescurtările utilizate în ordine alfabetică)*

AES - Advanced Encryption Standard

ASK - Amplitude Shift Keying

BPSK - Binary Phase Shift Keying

FSK - Frequency Shift Keying

GPIO - General Purpose Input/Output

HF - High Frequency

IF - Intermediate Frequency

ISO/IEC - International Organization for Standardization / International Electrotechnical Commission

LCD - Liquid Crystal Display

LO - Local Oscillator

MIFARE - (Standard de carduri de proximitate RFID, fără denumire extinsă în document)

NRZ - Non-Return-to-Zero

PSK - Phase Shift Keying

RFID - Radio Frequency Identification

SPI - Serial Peripheral Interface

SSL - Secure Sockets Layer

TLS - Transport Layer Security

UART - Universal Asynchronous Receiver/Transmitter

# **1 INTRODUCERE**

---

Scopul proiectului

Obiective

principale

Structura lucrării

---

În era actuală, în care securitatea și gestionarea risurilor devin tot mai importante în fața amenințărilor globale, dezvoltarea unor soluții tehnologice eficiente pentru monitorizarea și controlul accesului devine esențială. Unul dintre aspectele semnificative ale acestui demers îl constituie securitatea spațiilor interioare, unde managementul accesului și monitorizarea prezenței reprezintă elemente cheie pentru asigurarea siguranței și protecției. Acest proiect propune dezvoltarea unui sistem avansat de control al prezenței, bazat pe tehnologia RFID și alte componente hardware, cu scopul de a asigura o gestionare eficientă a accesului într-o încăpere sau spațiu specific și folosirea eficientă a unei baze de date și stocarea informațiilor. Implementarea acestui sistem va contribui la întărirea securității și la îmbunătățirea eficienței în monitorizarea și gestionarea prezenței și se va ține cont de cătă lume este prezentă în acel moment în acel loc, reprezentând astfel o soluție viabilă în contextul actual al necesităților desecuritate și control al accesului.

## **1.1 SCOPUL PROIECTULUI**

În această lucrare sunt prezentate principalele aspecte legate de realizarea unui sistem demonitorizare a prezenței folosind tehnologia RFID și alte componente hardware. Scopul acestui proiect este să dezvolte un sistem eficient de control al accesului într-o încăpere sau spațiu specific, utilizând tehnologia RFID și alte componente hardware.

Partea practică a acestui sistem implică implementarea unui modul care poate efectua citirea și prelucrarea datelor de la tag-urile RFID și să le afișeze printr-un display LCD. Acest modul poate, de asemenea, să activeze un servomotor pentru a deschide sau închide o ușă în funcție de accesul

permis. Sistemul este conceput pentru a monitoriza și înregistra prezența în timp real a persoanelor în încăpere și poate emite semnale sonore pentru a indica autorizarea sau neautorizarea accesului.

Principalele funcții realizate de sistem sunt:

- Citirea și prelucrarea datelor de la tag-urile RFID;
- Afisarea informațiilor despre prezență pe un display LCD;
- Controlul accesului prin activarea unui servomotor pentru deschiderea/inchiderea ușii;
- Emiterea de semnale sonore pentru confirmarea sau respingerea accesului;
- Înregistrarea și transmiterea datelor despre prezență către alte dispozitive pentru monitorizare la distanță.

## **1.2 STRUCTURA LUCRĂRII**

Structura lucrării se împarte în următoarele capitole :

- Capitolul 1 prezintă tema, scopul și obiectivele proiectului.
- Capitolul 2 prezintă stadiul actual și comercial în domeniu dar și câteva informații despre proiect.
- Capitolul 3 prezintă configurarea hardware.
- Capitolul 4 prezintă informații despre Raspberry PI.
- Capitolul 5 prezintă baza de date.
- Capitolul 6 prezintă posibilități de sporire a securității.
- Capitolul 7 prezintă partea de implementare
- Capitolul 8 prezintă implementări pe viitor.
- Capitolul 9 prezintă concluziile.

### **1.3 OBIECTIVE PRINCIPALE**

Aici se va prezenta care sunt principalele obiective de la acest proiect atât hardware, cât și software.

#### **1.3.1 Obiective hardware**

Secțiunea aceasta abordează obiectivele fizice pe care modulele le vor îndeplini în cadrul proiectului de control al accesului cu tehnologie RFID:

- Găsirea și înțelegerea modului de funcționare al unui modul RFID potrivit pentru cerințe.
- Realizarea alimentării și comunicării cu modulul RFID, pentru a permite citirea și prelucrarea datelor de la tag-urile RFID și pentru a asigura funcționarea corespunzătoare asistemului.
- Implementarea unui display LCD pentru afișarea informațiilor despre prezență și accesibilitatea în încăpere, pentru a facilita interacțiunea utilizatorului cu sistemul.
- Integrarea unui buzzer pentru a emite semnale sonore în cazul autorizării sau neautorizării accesului, asigurând o notificare auditivă clară pentru utilizatori.

#### **1.3.2 Obiective software**

Secțiunea aceasta abordează algoritmii și metodele folosite pentru programarea în sine:

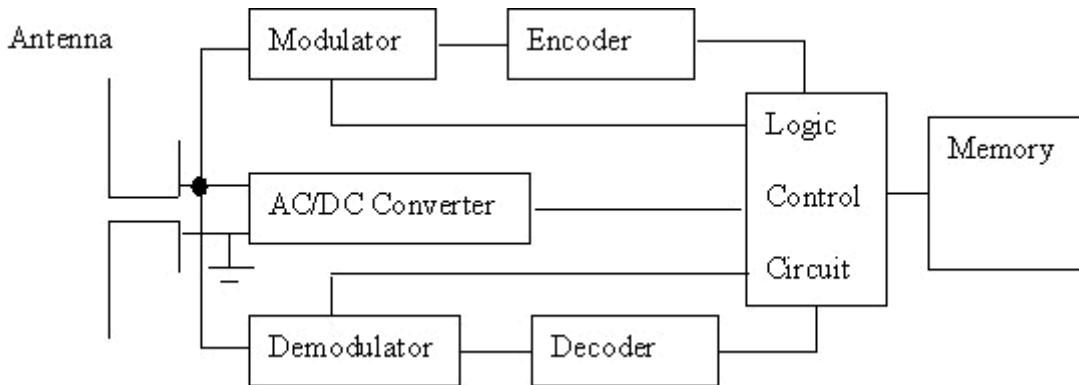
- Dezvoltarea unui algoritm pentru afișarea corectă a informațiilor despre prezență și accesibilitate pe display-ul LCD, asigurând o prezentare clară și ușor de înțeles a datelor pentru utilizatori.
- Implementarea unui algoritm de citire și prelucrare a datelor de la tag-urile RFID, pentru a identifica și valida utilizatorii autorizați și pentru a înregistra prezența lor în încăpere.
- Realizarea unui algoritm de comparare a datelor preluate cu parametrii setați, pentru a verifica conformitatea și validitatea accesului utilizatorilor.
- Programarea funcționalității de conexiune la distanță cu un dispozitiv, permitând monitorizarea și gestionarea sistemului de control al accesului de la distanță, pentru administrarea eficientă a accesului în încăpere.

## **2 STADIUL ACTUAL ÎN DOMENIU**

### **2.1 SISTEME RFID**

Un sistem de control RFID este o soluție tehnologică care utilizează cititoare și tag-uri RFID pentru monitorizarea și gestionarea accesului într-un anumit spațiu sau mediu. Funcționează prin citirea și înregistrarea identității tag-urilor RFID, care sunt atașate obiectelor sau persoanelor, pentru a permite sau restricționa accesul în diverse zone. Acest sistem are aplicații în securitatea clădirilor, gestionarea inventarului, urmărirea și localizarea activelor, autentificarea produselor și multe altele, contribuind la îmbunătățirea eficienței și securității în diverse industrii.

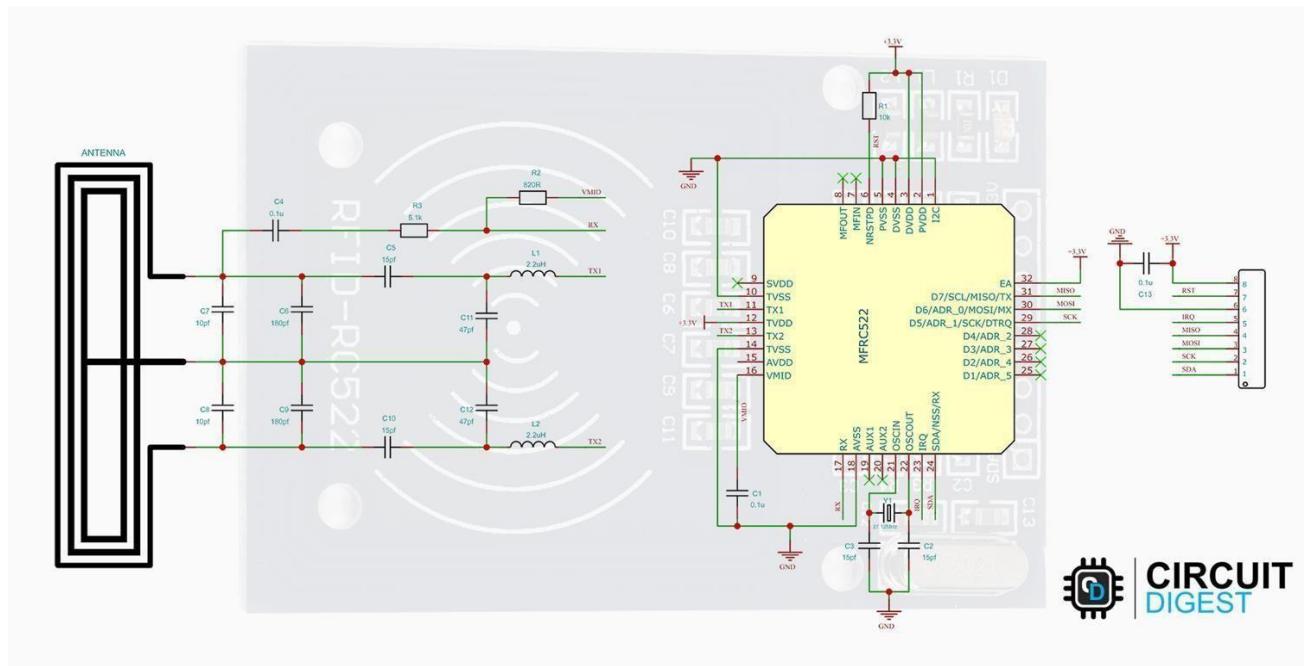
Cititorul emite un semnal radio către tag-ul RFID, care conține un chip electronic. Tag-ul RFID primește semnalul și răspunde înapoi cu informații stocate în chip, cum ar fi un cod unic de identificare. Cititorul primește apoi aceste informații și le procesează pentru a autentifica sau a identifica obiectul sau persoana asociată cu tag-ul RFID. În funcție de setările preconfigurate, sistemul poate sau restricționa accesul într-un anumit spațiu sau poate înregistra prezența într-o anumită locație.



**Fig 1.Schema bloc RFID [1]**

### Componentele principale:

- **Antenă:** Recepționează undele radio din aer și le transformă într-un semnal electric.
- **Filtru de bandă:** Elimină semnalele nedorite din afara benzii de frecvență a stației emisore.
- **Mixer:** Amestecă semnalul recepționat cu un semnal local oscilator (LO) pentru a converti frecvența semnalului recepționat la o frecvență intermediară (IF).
- **Filtru de frecvență intermediară (IF):** Selectează și amplifică semnalul IF dorit, eliminând altele semnale IF.
- **Demodulator:** Extragă informația audio sau video din semnalul IF modulat.
- **Decoder:** Decodifică informația audio sau video extrasă de demodulator.
- **Filtru de ieșire:** Elimină zgomotul rezidual și îmbunătățește calitatea semnalului audio sau video de ieșire.
- **Amplificator de ieșire:** Amplifică semnalul audio sau video de ieșire la un nivel adecvat pentru a fi redat prin difuzoare sau afișat pe un ecran.[1]



Fif 2.Diagramă RC 522 [2]

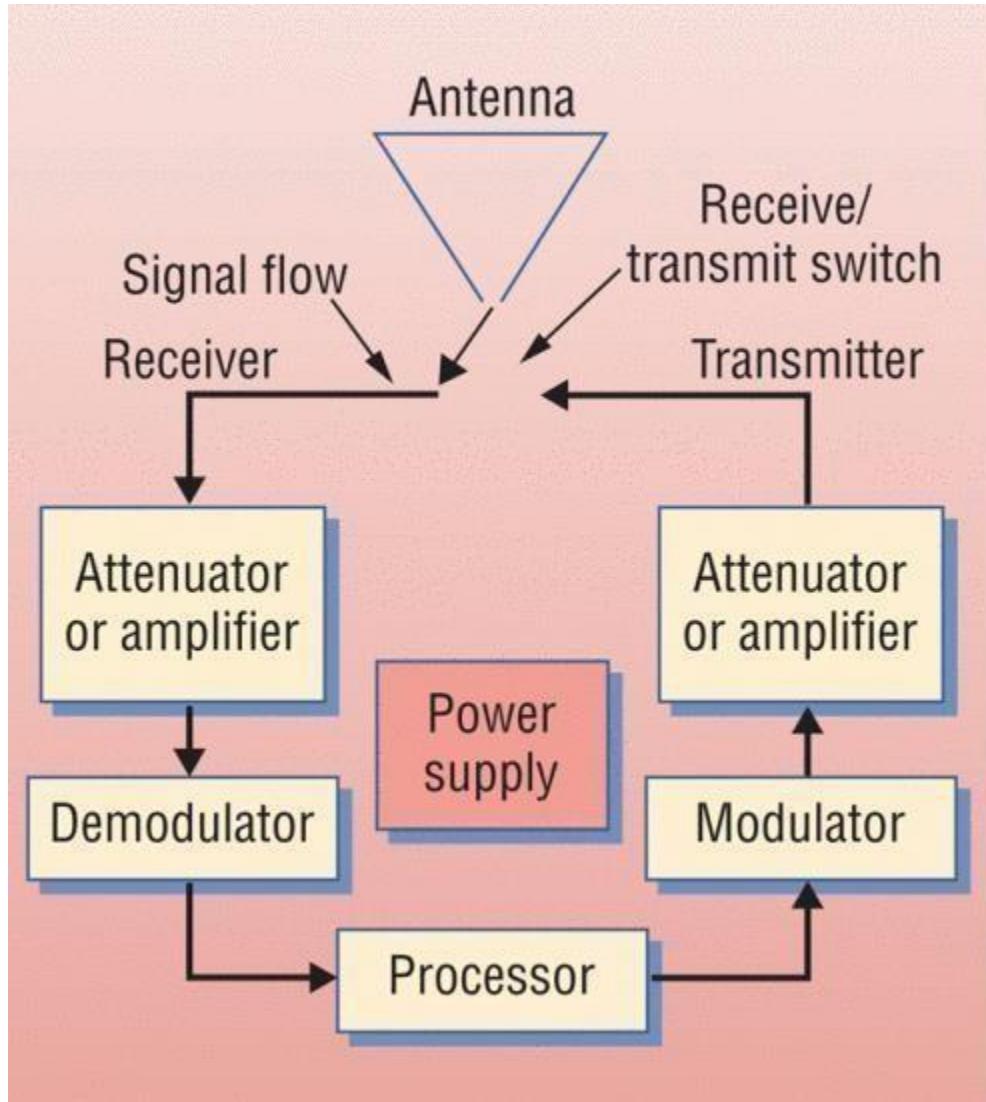


Fig.3 Schema bloc funcționare RFID[3]

Transmițătoarele RFID, sunt calculatoare mici și curesurse limitate. În sistemele pasive, ele detectează un semnal care sosește de la un cititor, activează eticheta, trimit un răspuns și stochează o cantitate mică de date. Cantitatea de stocare depinde de utilizare, variind de la câțiva biți pentru aplicații precum un sistem de control al inventarului unei mici magazine până la mai mulți kilobiți pentru aplicații precum un sistem mare de lanț de aprovizionare al unei afaceri.[3]

Name	Frequency range	ISM frequencies
LF	30300 kHz	< 135 kHz
HF	330 MHz	6.78 MHz, 13.56 MHz, 27.125 MHz, 40.680 MHz
UHF	300 MHz-3 GHz	433.920 MHz, 869 MHz, 915 MHz
Microwave	> 3 GHz	2.45 GHz, 5.8 GHz, 24.125 GHz

Fig 4.Raza de acțiune de frecvență RFID[4]

În acest caz se va folosi un dispozitiv cu frecvență de 13.56 MHz, acest detaliu este important pentru a știi în ce categorie ne încadrăm.

Name	Frequency range	ISM frequencies
LF	30300 kHz	< 135 kHz
HF	330 MHz	6.78 MHz, 13.56 MHz, 27.125 MHz, 40.680 MHz
UHF	300 MHz-3 GHz	433.920 MHz, 869 MHz, 915 MHz
Microwave	> 3 GHz	2.45 GHz, 5.8 GHz, 24.125 GHz

Fig.5 Raza de acțiune de citire în funcție de frecvență[4]

Modulul RC522 utilizează un protocol de comunicare specific numit Modulare Amplitudine/Demodulare Amplitudine (ASK/DEM) pentru a transmite și a recepționa semnalele între cipul RFID și cititorul RFID.

Acest protocol de comunicare este compatibil cu ISO/IEC 14443, care este un standard internațional pentru sistemele de identificare prin radiofrecvență (RFID). Înceea ce privește codificarea datelor, RC522 poate utiliza diferite formate de codificare, cum ar fi Manchester, BPSK (Binary Phase Shift Keying) sau NRZ (Non-Return-to-Zero), în funcție de cerințele aplicației și de modul de configurare a modulului.[4]

**Codarea Manchester** folosește o tranziție negativă în mijlocul celulei de bit pentru a însemna valoarea "1" și o tranziție pozitivă în mijlocul celulei de bit pentru a însemna valoarea "0". Tranzițiile care au loc la limita celulei de bit nu codifică o valoare și sunt folosite pentru a "reseta" codificarea pentru a trimite aceeași valoare din nou în mijlocul celulei. Codificarea este

independentă de modulație, deci aceasta ar putea fi o tranziție ASK, FSK sau PSK. Deoarece tranzițiile sunt sincronizate cu ceasul, se spune că această metodă are ceasul în fluxul de date. Unde destinatar poate deriva semnalul de ceas din această codificare, ceea ce face ca codificarea Manchester să fie foarte potrivită pentru transferul de mesaje mari. Etichetele de tip ISO 18000-6 tip B folosesc codificarea Manchester pentru a comunica cu un cititor.[4]

In **Codarea NRZ** fiecare bit de date este reprezentat prin unul dintre cele două nivele ale semnalului: nivel înalt (1) sau nivel scăzut (0). În timpul transmisiei, semnalul rămâne la nivelul său specificat pentru întreaga durată a fiecărui bit de date.

La receptor, semnalul este decodificat pentru a identifica biții individual. Nivelul semnalului este măsurat și comparat cu un prag specific pentru a determina valoarea binară a fiecărui bit.[4]

## 2.1.2 SECURITATE

### A. Probleme de securitate în sistemele de management al prezentei cu RFID

Sistemele de prezență bazate pe RFID se confruntă cu probleme de securitate, inclusiv accesul neautorizat la date sensibile și interceptarea semnalelor RFID. Algoritmi de criptare precum AES pot fi folosiți pentru a asigura transmisia securizată a datelor. Canalele de comunicare securizate precum SSL sau TLS pot fi, de asemenea, folosite pentru a proteja împotriva atacurilor.

B. Importanța criptării în securizarea sistemelor de management al prezentei bazate pe RFID este crucială pentru securizarea sistemelor de management al prezentei bazate pe RFID. Înălțarea popularității RFID-ului datoră avansului tehnologic, au apărut și preocupări privind confidențialitatea și securitatea datelor. Cercetătorii subliniază importanța criptării în securizarea sistemelor bazate pe RFID pentru a asigura confidențialitatea și securitatea datelor individuale. Implementarea algoritmilor de criptare poate îmbunătăți semnificativ securitatea sistemelor de management al prezentei bazate pe RFID, protejând datele sensibile împotriva atacurilor și prevenind accesul neautorizat.[5]

## 2.1.3 COMUNICAȚIE

Modulul RC522 utilizează mai multe protocoale de comunicație pentru a interacționa cu etichetele RFID. Printre cele mai comune protocoale folosite se numără:

1. ISO/IEC 14443: Acest protocol este utilizat pentru comunicarea între modulul RC522 și etichetele RFID de tip contactless. Protocolul definește formatele de date, comenzi și procedurile de comunicație necesare pentru interacțiunea corectă între dispozitive.
2. MIFARE: Acesta este un protocol specific dezvoltat de către NXP Semiconductors pentru etichetele lor RFID MIFARE. Protocolul MIFARE se bazează pe standardul ISO/IEC 14443 și adaugă funcționalități suplimentare, cum ar fi autentificarea și criptarea datelor.
3. Serial Peripheral Interface (SPI): Acest protocol este folosit pentru comunicarea între modulul RC522 și microcontrollerul sau dispozitivul gazdă, cum ar fi un Raspberry Pi. Protocolul SPI permite transferul rapid de date între dispozitive și este utilizat pentru configurarea modulului RC522 și pentru transmiterea și recepția de date.
4. Universal Asynchronous Receiver/Transmitter (UART): Acest protocol poate fi folosit pentru comunicația serială între modulul RC522 și un dispozitiv gazdă. Deși nu este atât de des utilizat ca SPI în aplicațiile RC522, UART poate fi o opțiune pentru unele configurații de comunicație.

Mai departe se va vorbi puțin despre protocolul SPI care face legătura dintre RFID și Raspberry, parte crucială în proiectul meu.

Interfața Serial Peripheral Interface (SPI) este un protocol de comunicare sincronă utilizat pentru transferul rapid de date între dispozitive, cum ar fi modulul RFID RC522 și Raspberry Pi.

Configurare pinilor:

- Interfața SPI utilizează patru linii pentru comunicare:
  - MISO (Master In Slave Out): Această linie transportă datele de la slave (RC522) către master (Raspberry Pi).
  - MOSI (Master Out Slave In): Această linie transportă datele de la Master(Raspberry Pi) către Slave (RC522).
  - SCK (Serial Clock): Această linie transportă semnalul de ceas generat de master (Raspberry Pi) pentru a sincroniza transmiterea datelor.
  - SS/CS (Slave Select/Chip Select): Această linie este folosită de master(Raspberry Pi) pentru a selecta slave (RC522) pentru comunicare.[10]

- **Inițializare:**

- Raspberry Pi-ul inițializează comunicarea SPI prin configurarea interfeței hardware SPI și configurația parametrilor necesari, cum ar fi viteza ceasului și formatul datelor.
  - Pinul SS/CS al modulului RC522 este conectat la un pin GPIO al Raspberry Pi-ului, care este folosit pentru a permite comunicarea cu RC522.

- **Schimbul de Date:**

- Pentru a trimite date la modulul RC522, Raspberry Pi-ul scrie date pe linia sa MOSI și generează simultan impulsuri de ceas pe linia SCK.
  - RC522-ul citește datele de pe linia sa MOSI și efectuează operațiile necesare.
  - Pentru a primi date de la modulul RC522, Raspberry Pi-ul generează impulsuri de ceas pe linia SCK în timp ce citește datele de pe linia sa MISO.
  - RC522-ul trimite date înapoi către Raspberry Pi prin linia sa MISO.

- **Selectarea Slave-ului:**

- Înainte de a iniția comunicarea cu modulul RC522, Raspberry Pi-ul selectează RC522-ul răgând pinul său SS/CS la nivel scăzut.
  - După ce comunicarea este completă, Raspberry Pi-ul eliberează RC522-ul setând din nou pinul SS/CS la nivel înalt.

- **Finalizare proces:**

- Odată ce schimbul de date este complet, comunicarea SPI este încheiată, iar Raspberry Pi-ul poate efectua alte sarcini sau poate iniția comunicarea cu alte dispozitive SPI, dacă este necesar.

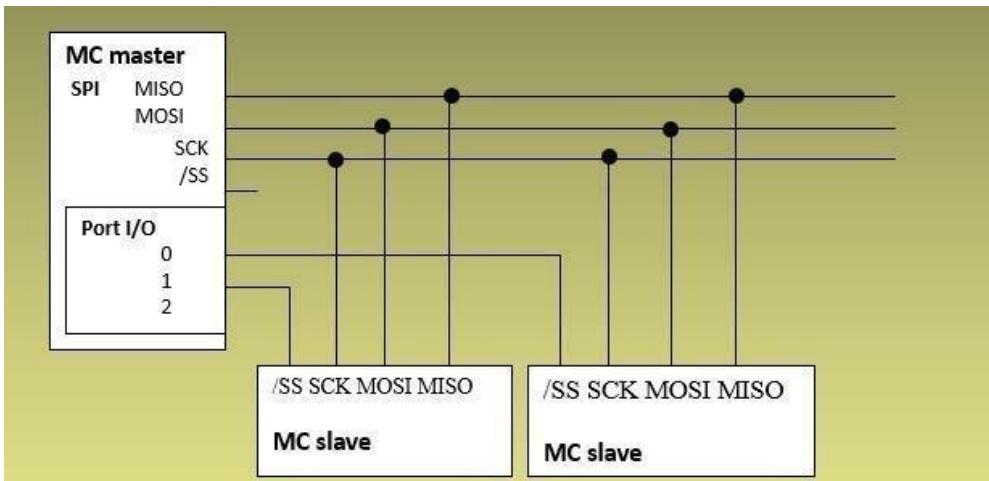


Fig.6 Schemă protocol[6]

### 2.1.3 STRUCTURĂ HF

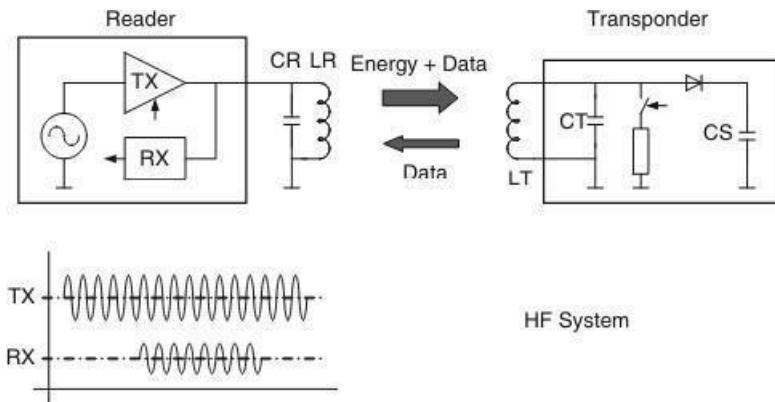


Fig 7. Hf system[7]

Într-un sistem RFID HF (Frecvență Înaltă), schimbul de date între cititor (reader) și transponder (tag) implică utilizarea energiei de alimentare și a datelor. În general, comunicarea începe cu cititorul, care transmite un semnal de energie către transponder pentru a-l activa și a-i permite să răspundă.

#### 1. Transmiterea (TX) de la cititor către transponder:

- Cititorul emite un semnal de radiofrecvență (RF) cu o anumită frecvență și putere.
- Acest semnal de RF este modulat pentru a include informații de control și date.
- În mod tipic, semnalul de energie este transmis utilizând modulație amplitudinală (ASK - Amplitude Shift Keying) sau modulație de frecvență (FSK - Frequency Shift Keying), unde modificările în semnalul de RF indică informațiile transmise.[7]

#### 2. Recepția (RX) de la transponder către cititor:

- Transponderul primește semnalul de energie RF și îl convertește în energie electrică pentru a se alimenta.

- După ce este activat, transponderul răspunde la cititor prin modularea înapoi asemnalului RF.
- Răspunsul transponderului poate include date și informații de control, care sunt modulate pe semnalul de RF.
- Cititorul detectează semnalul de răspuns și îl interpretează pentru a extrage datele transmise de transponder.[7]

## 2.2 IMPLEMENTĂRI

Lansat inițial de către Zebra Technologies în anul 2010, cititorul RFID Zebra FX7500 a reprezentat un punct de cotitură în evoluția tehnologiei RFID UHF. Prin introducerea unor caracteristici inovatoare și fiabile, acest cititor a devenit rapid un standard în industria urmăririi și identificării obiectelor. Cu o istorie solidă de performanță și fiabilitate, Zebra FX7500 continuă să fie alegerea preferată pentru companiile din întreaga lume.

### Zebra FX7500

Zebra FX7500 este un cititor RFID UHF cu performanțe excelente, capabil să detecteze și să identifice etichetele RFID la distanțe mari. Construit pentru utilizare intensivă, acest cititor este robust și durabil, funcționând în condiții extreme de temperatură și umiditate. Cu o interfață intuitivă, Zebra FX7500 oferă o experiență de utilizare simplă și eficientă. Dispune de opțiuni de conectivitate extinsă, inclusiv Ethernet și WiFi, pentru integrare în rețele existente și comunicare cu alte dispozitive. Beneficiază de protocoale de securitate avansate și funcții de criptare pentru protejarea datelor și informațiilor. Automatizează procesul de urmărire a prezenței, oferind evidențe precise și fiabile în diverse medii.



Fig.8. Zebra FX7500[8]

<b>Processor</b>	Texas Instruments AM3505 (600 Mhz)
<b>Memory</b>	Flash 512 MB; DRAM 256 MB
<b>Operating System</b>	Linux
<b>Firmware Upgrade</b>	Web-based and remote firmware upgrade capabilities
<b>Management Protocols</b>	RM 1.0.1 (with XML over HTTP/HTTPS and SNMP binding); RDMP
<b>Network Services</b>	DHCP, HTTPS, FTPS, SFPT, SSH, HTTP, FTP, SNMP and NTP
<b>Network Stack</b>	IPv4 and IPv6
<b>Security</b>	Transport Layer Security Ver 1.2, FIPS-140
<b>Air Protocols</b>	EPCglobal UHF Class 1 Gen2, ISO 18000-6C
<b>Frequency (UHF Band)</b>	Global Reader: 902 MHz–928 MHz (Maximum, supports countries that use a part of this band), 865 MHz–868 MHz US (only) Reader: 902 MHz–928 MHz

Fig. 9 Specificații[8]

## 2.3 SIMULARE SIMULINK

Un filtru de medie mobilă poate fi utilizat pentru a estompa fluctuațiile sau zgometul din semnalul recepționat de la tag-urile RFID. Acest tip de filtru este util pentru a îmbunătăți precizia și fiabilitatea datelor colectate de către cititorul RFID.

Funcționarea unui filtru de medie mobilă constă în calcularea mediei unui număr fix de puncte de date consecutive dintr-o serie de date. În cazul RFID, semnalul recepționat de la tag-uri poate fi afectat de interferențe sau fluctuații ale mediului, ceea ce poate duce la date inexacte sau eronate. Prin aplicarea unui filtru de medie mobilă, aceste fluctuații pot fi estompate, permitând cititorului RFID să detecteze și să interpreteze mai precis semnalul provenit de la tag-uri.

Filtrul de medie mobilă servește ca un instrument de prelucrare a semnalului, eliminând variațiile bruște și evidențind tendințele sau modelele subiacente din datele RFID. Prin utilizarea acestui filtru, se poate îmbunătăți calitatea datelor colectate, sporind astfel performanța sistemului RFID în diverse aplicații, cum ar fi monitorizarea prezenței sau urmărirea obiectelor în lanțul de aprovizionare.

M-am gândit că implementarea acestui filtru ar fi o idee foarte bună pentru a avea date mai stabile și eficiență mai bună.

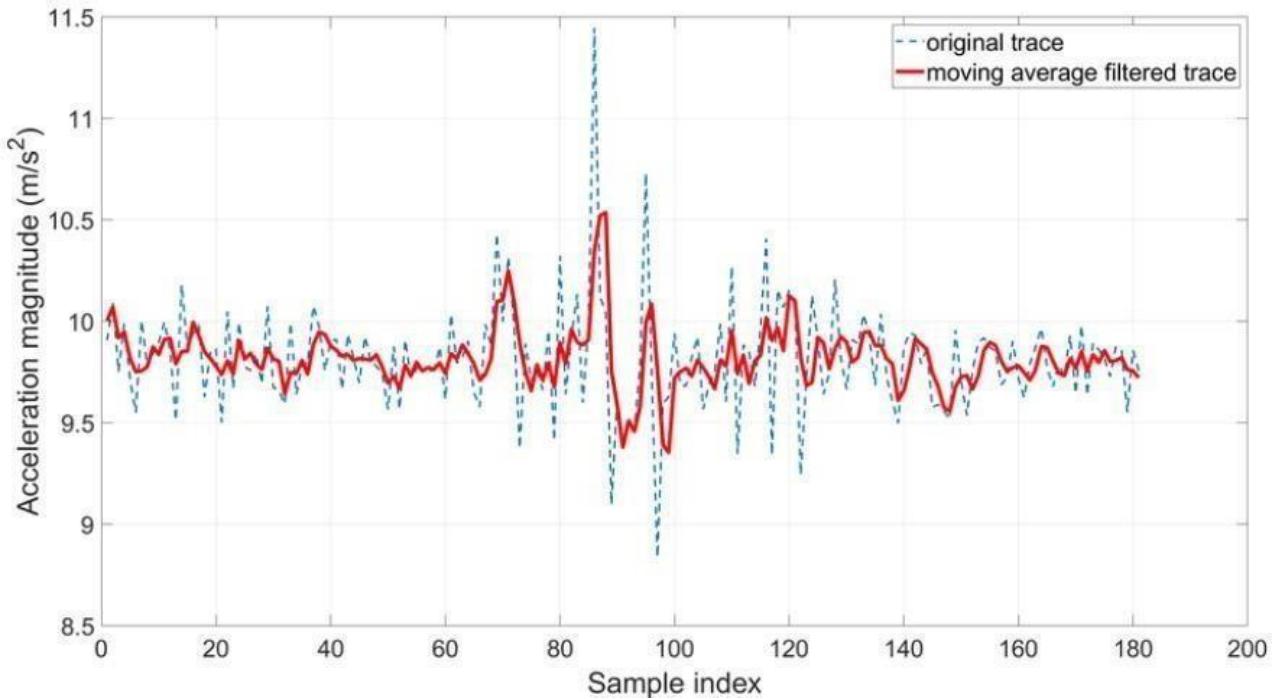


Fig.10 Efectul unui filtru moving average[9]

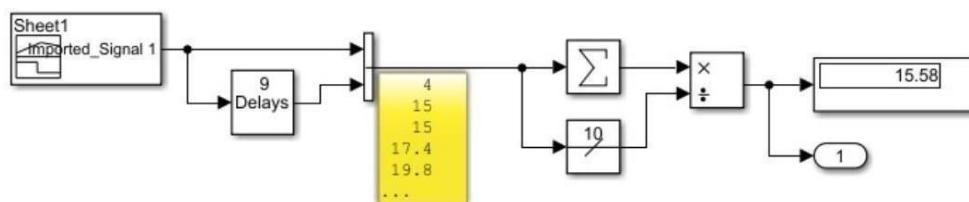


Fig.11 Simulare filtru

**Sheet1 (Imported\_Signal1):** Acest bloc importă semnalul de intrare dintr-o sursă externă, cum ar fi un fișier sau un workspace.

- **Rol:** Servește ca sursă de date pentru simulare, reprezentând semnalul RFID brut colectat de cititor.

**9 Delays:** Un bloc de întârziere care stochează un număr specific de probe ale semnalului de intrare.

- **Rol:** Stochează ultimele 9 valori ale semnalului de intrare pentru a le folosi în calculul mediei mobile.

**Blocul de Stocare (Memory Block):**Acest bloc păstrează valorile stocate în blocul de întârziere.

- **Rol:** Stochează un set de valori care vor fi utilizate pentru calculul mediei.

**$\Sigma$  (Sum):**Un bloc de sumare care adună toate valorile stocate în blocul de stocare.

- **Rol:** Calculează suma totală a valorilor stocate, care va fi folosită pentru a calcula media mobilă.

**Blocul de Împărțire (Division Block):**Un bloc care împarte suma totală a valorilor la numărul de probe (în acest caz, 10).

- **Rol:** Calculează media valorilor stocate, oferind rezultatul filtrului de medie mobilă.

**Blocul de Constante (Constant Block):**Un bloc care furnizează o valoare constantă pentru împărțirea sumelor.

- **Rol:** Definește numărul de valori care sunt luate în considerare pentru calculul mediei (în acest caz, 10).

**Blocul de Afisare (Display Block):**Un bloc de afişare care arată rezultatul final al calculului.

- **Rol:** Vizualizează media calculată a semnalului, reprezentând semnalul filtrat și prelucrat.

**Semnalul de Intrare:** Este importat prin blocul Sheet1 (Imported\_Signal1), care reprezintă datele brute colectate de la cititorul RFID. Aceste date pot conține zgomot și fluctuații care necesită filtrare.

**Blocul de Întârziere:** Stochează ultimele 9 valori ale semnalului, pregătindu-le pentru calculul mediei mobile. Acest bloc este esențial pentru a avea un set de date asupra căruia se va aplica filtrarea.

**Blocul de Stocare:** Menține aceste valori pentru a le putea utiliza în calculul ulterior.

**Blocul de Sumare:** Adună toate valorile stocate pentru a obține suma totală a valorilor semnalului în fereastra de timp selectată.

**Blocul de Împărțire:** Împarte suma totală la numărul de valori (10) pentru a calcula media mobilă. Aceasta reprezintă valoarea filtrată a semnalului, eliminând zgomotul și fluctuațiile.

**Blocul de Constante:** Asigură că suma este împărțită corect la numărul de probe, asigurând calculul corect al mediei.

**Blocul de Afisare:** Prezintă rezultatul final al calculului, care este semnalul filtrat și stabilizat.

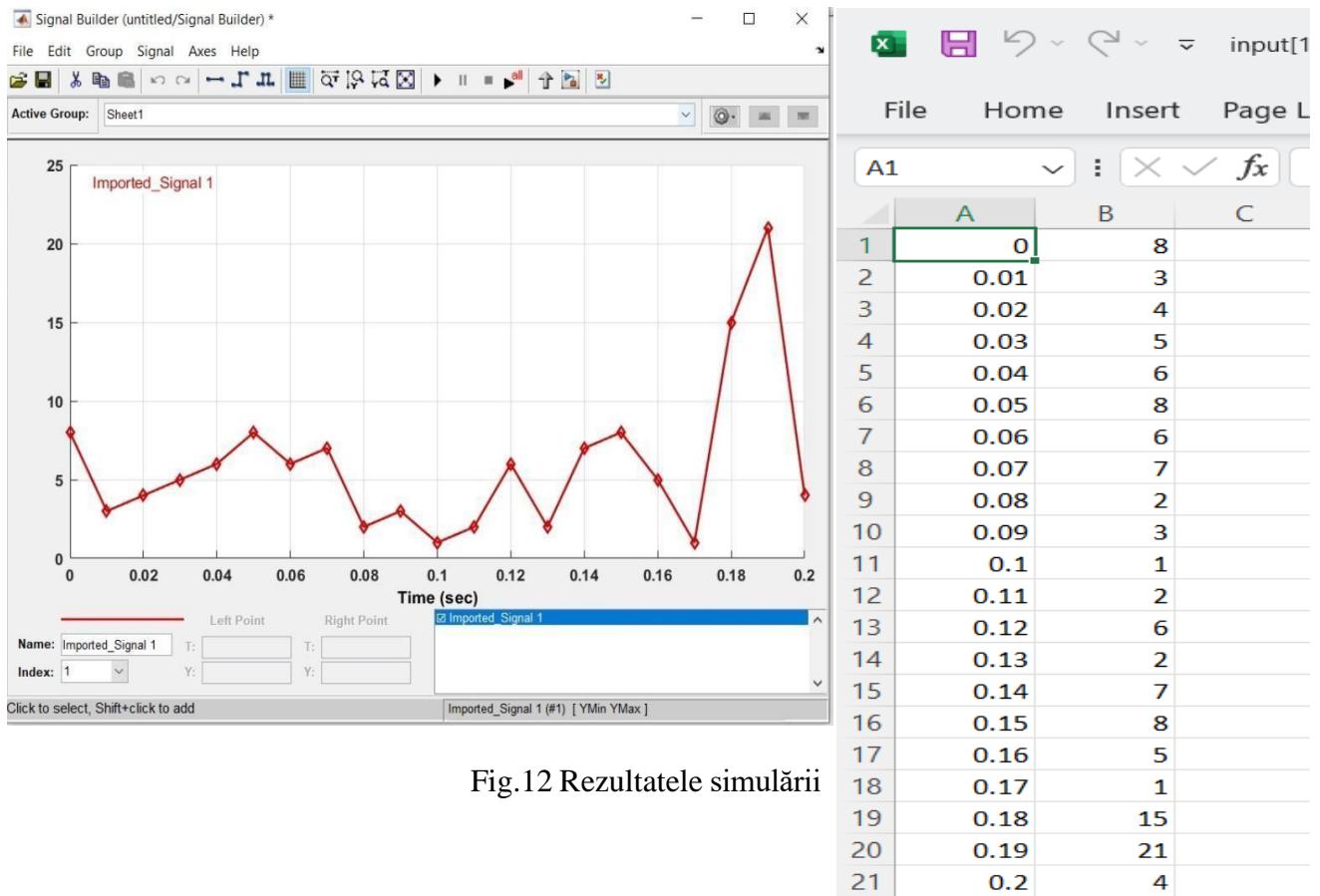


Fig.12 Rezultatele simulării

Am creat un fișier Excel (input.xlsx) care conține datele organizate în două coloane: Time (sec) și Input Signal. În SIMULINK, am utilizat blocul Signal Builder pentru a importa aceste date. Am deschis SIMULINK și am creat un nou model, apoi am adăugat blocul Signal Builder în modelul meu. În interiorul Signal Builder, am selectat opțiunea de importare a datelor din fișierul Excel (Import from File). Am navigat la fișierul Excel creat (input.xlsx) și am selectat foaia care conține datele (Sheet1). Am mapat coloana A la axa timpului și coloana B la axa valorilor semnalului.

După importarea datelor, Signal Builder a generat un grafic care arată evoluția semnalului de intrare în timp. Graficul rezultat prezintă fluctuațiile și zgomotul din semnalul importat. Am analizat graficul pentru a observa comportamentul semnalului, scopul fiind să identific fluctuațiile și să înțeleg cum se comportă semnalul în timp.

Acest proces de generare a datelor, importarea lor în SIMULINK și analiza graficului rezultat demonstrează cum valorile inițiale introduse manual în Excel pot fi utilizate pentru simulări și analize în SIMULINK. Această metodologie este esențială pentru a înțelege comportamentul semnalelor într-un mediu controlat și pentru a testa eficiența diferitelor algoritmi de prelucrare a semnalelor, cum ar fi filtrul moving average, fără a depinde de datele reale colectate de la senzori.

## Simulare proiect RFID si senzor

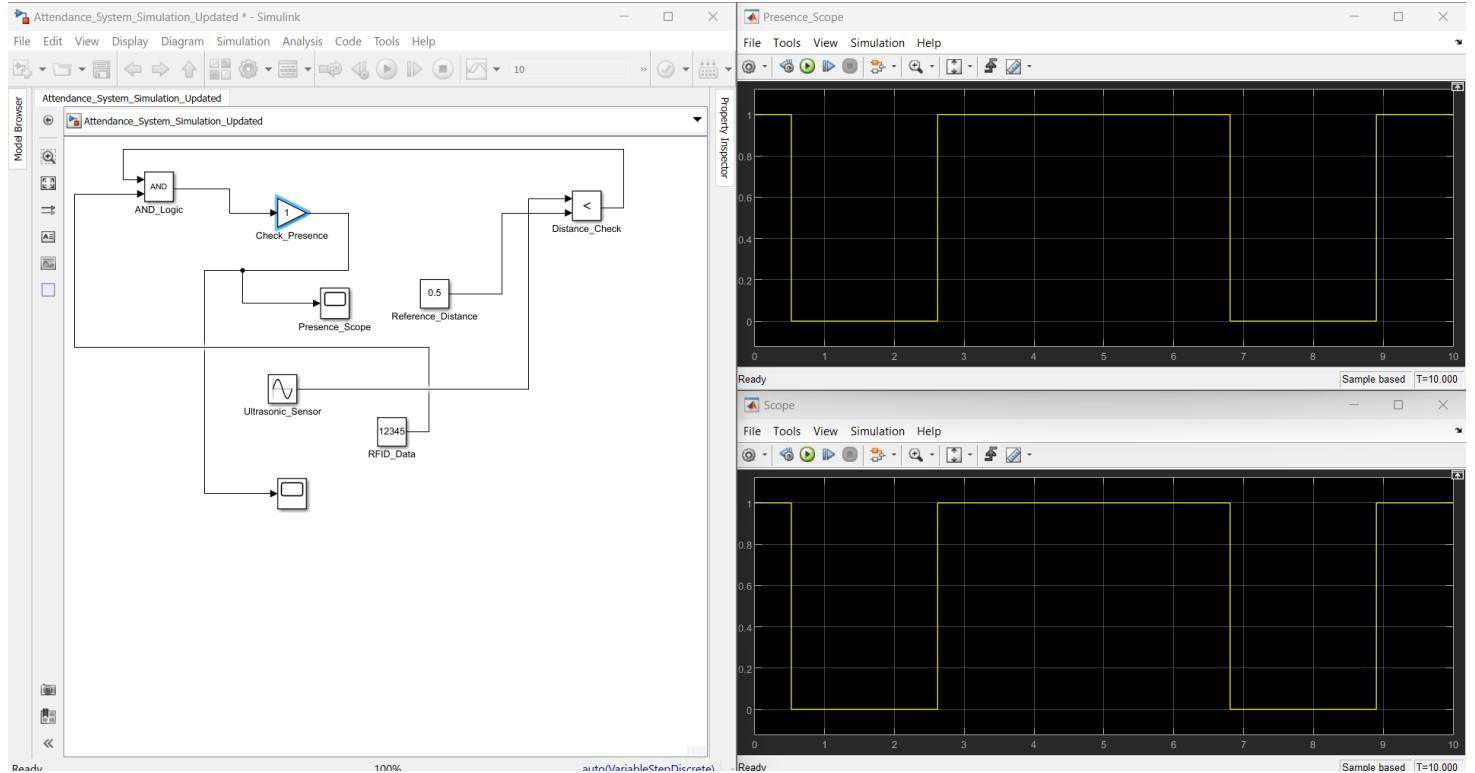


Fig 12.1 Rezultate simulare proiect pentru test

1. **RFID\_Data:** Blocul de date RFID conține un semnal constant care reprezintă un ID RFID, în acest caz, valoarea 12345. Acesta este folosit pentru a simula datele citite de la un tag RFID.

2. **Ultrasonic\_Sensor:** Blocul de senzor ultrasonic generează un semnal sinusoidal pentru a simula datele de la un senzor ultrasonic. Frecvența semnalului este setată la 1, ceea ce înseamnă că semnalul variază într-un interval de timp de 10 secunde.

3. **Reference\_Distance:** Acest bloc constant reprezintă o distanță de referință, setată la 0.5. Aceasta este utilizată pentru a compara distanțele măsurate de senzorul ultrasonic.

4. **Distance\_Check:** Acest bloc de operator relațional verifică dacă distanța măsurată de senzorul ultrasonic este mai mică decât distanța de referință (0.5). Dacă această condiție este adevărată, blocul produce un semnal 1; altfel, produce un semnal 0.

5. **AND\_Logic:** Blocul de logică AND combină semnalul de prezență și datele RFID. Acesta va produce un semnal 1 doar dacă ambele intrări sunt 1, ceea ce înseamnă că atât prezența este detectată, cât și un tag RFID valid este citit.

6. **Check\_Presence:** Blocul de câștig (Gain) aplică un câștig de 1 semnalului combinat. Practic, acesta trece semnalul mai departe fără modificări.

7. **Presence\_Scope și Scope:** Aceste blocuri de vizualizare (Scope) sunt utilizate pentru a vizualiza semnalele de prezență și rezultatul final al verificării distanței.

- Simularea detectează prezența utilizatorului folosind senzorul ultrasonic. Dacă distanța măsurată este mai mică decât 0.5, semnalul Distance\_Check devine 1.

• Datele RFID sunt constant 12345, ceea ce înseamnă că RFID-ul este întotdeauna valid.

• Blocul AND\_Logic combină aceste două semnale. Dacă ambele condiții sunt adevărate (prezența este detectată și RFID-ul este valid), produce un semnal 1.

- Semnalul combinat este trecut prin blocul Check\_Presence și este vizualizat în blocul Presence\_Scope.
- Semnalul de verificare a distanței este de asemenea vizualizat în blocul Scope.
- **Presence\_Scope:** Graficul arată starea prezenței detectate. Valorile sale variază între 0 și 1, indicând momentele în care prezența este detectată și datele RFID sunt valide.
- **Scope:** Graficul arată rezultatul verificării distanței. Valorile sale sunt 0 sau 1, indicând momentele în care distanța măsurată de senzorul ultrasonic este mai mică decât referința de 0.5.

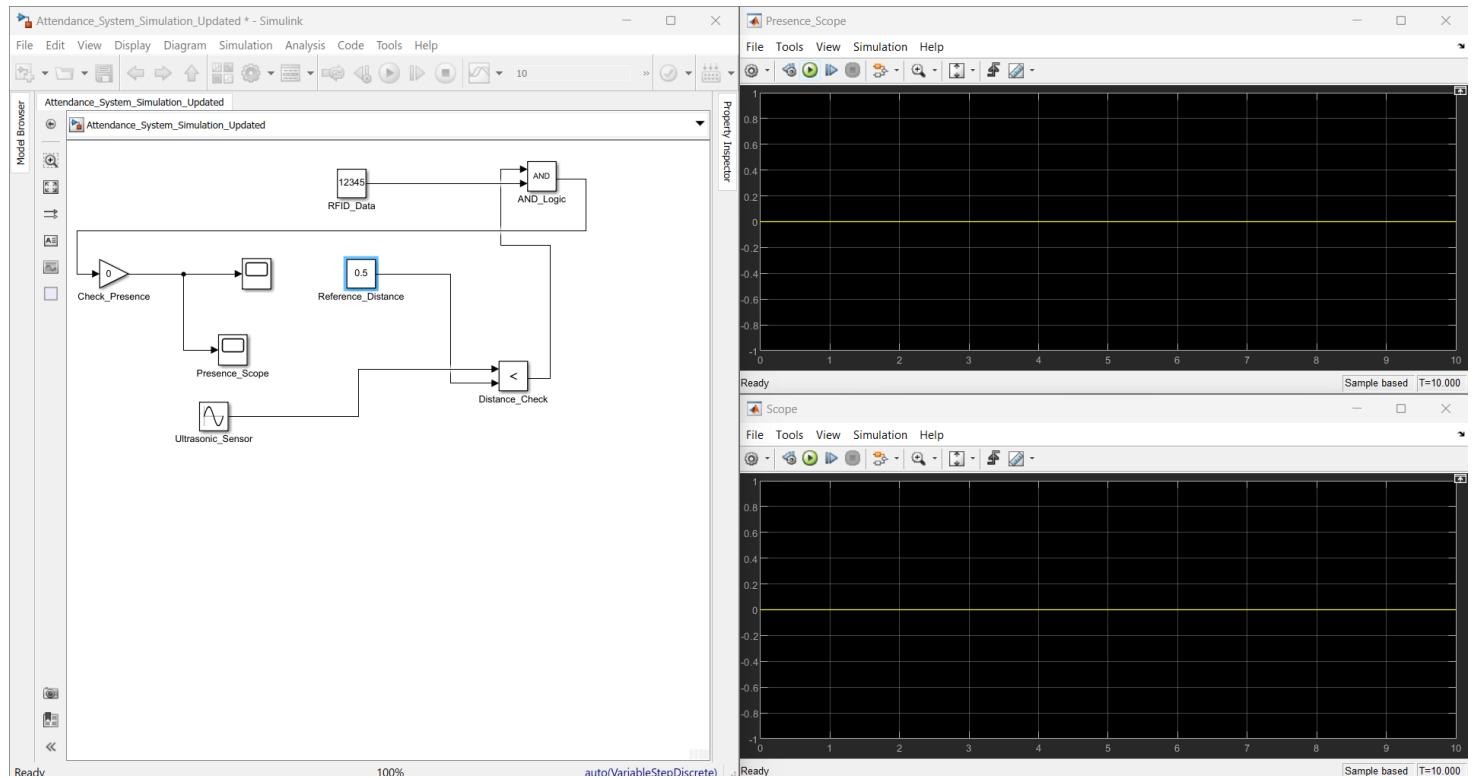


Fig.12.2 Rezultate simulare test 2 când una dintre condiții este falsă

Semnalul de pe **Presence\_Scope** este constant 0, deoarece valoarea setată pentru **Check\_Presence** este 0. Aceasta indică faptul că nu este detectat nimic.

Semnalul de pe **Scope** este, de asemenea, constant 0. Acest lucru se datorează faptului că operația logică AND dintre semnalul de prezență (0) și semnalul de distanță (oricare ar fi acesta) va fi întotdeauna 0.

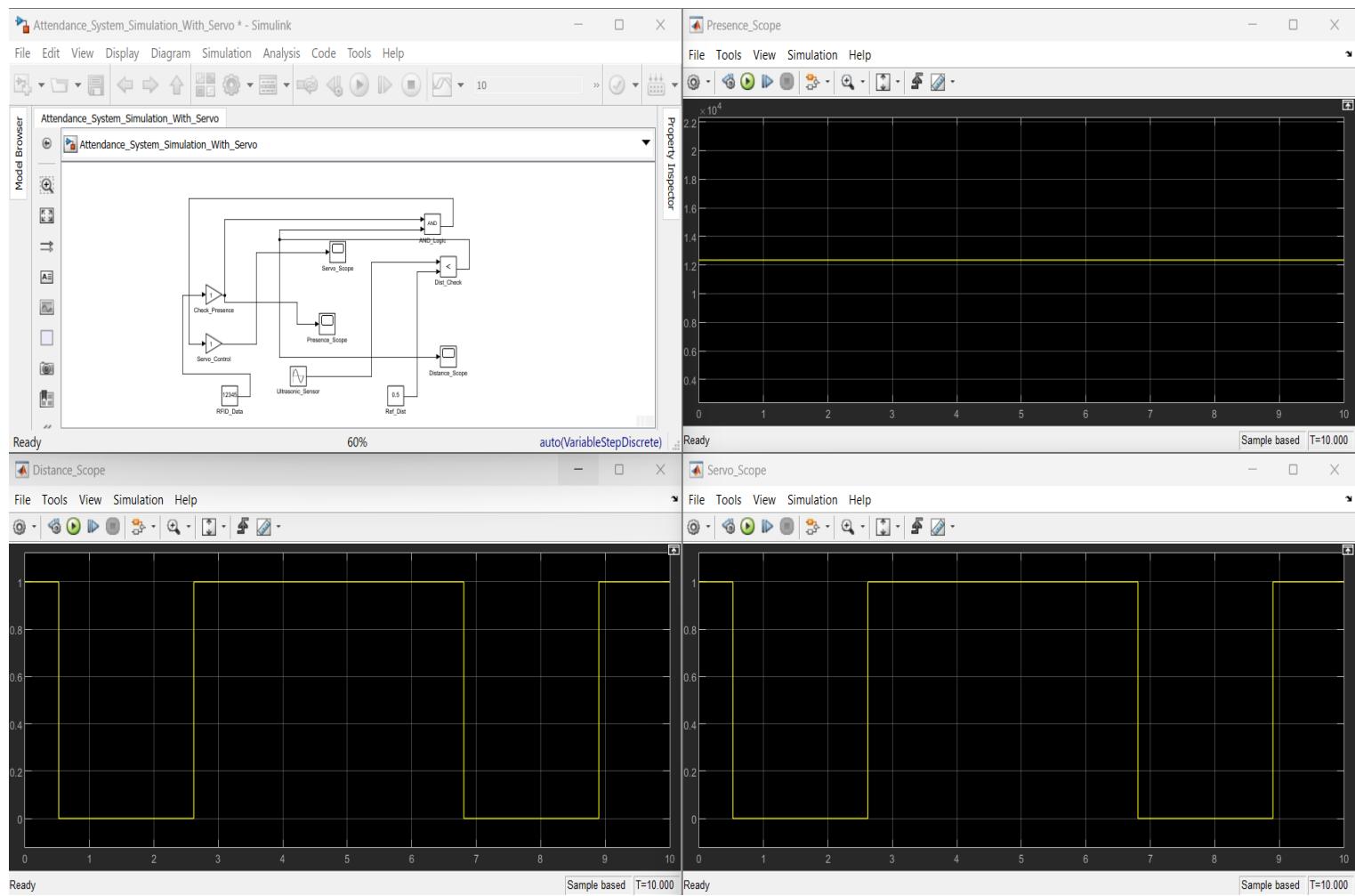


fig.12.3 Formele de undă simulare cu servo

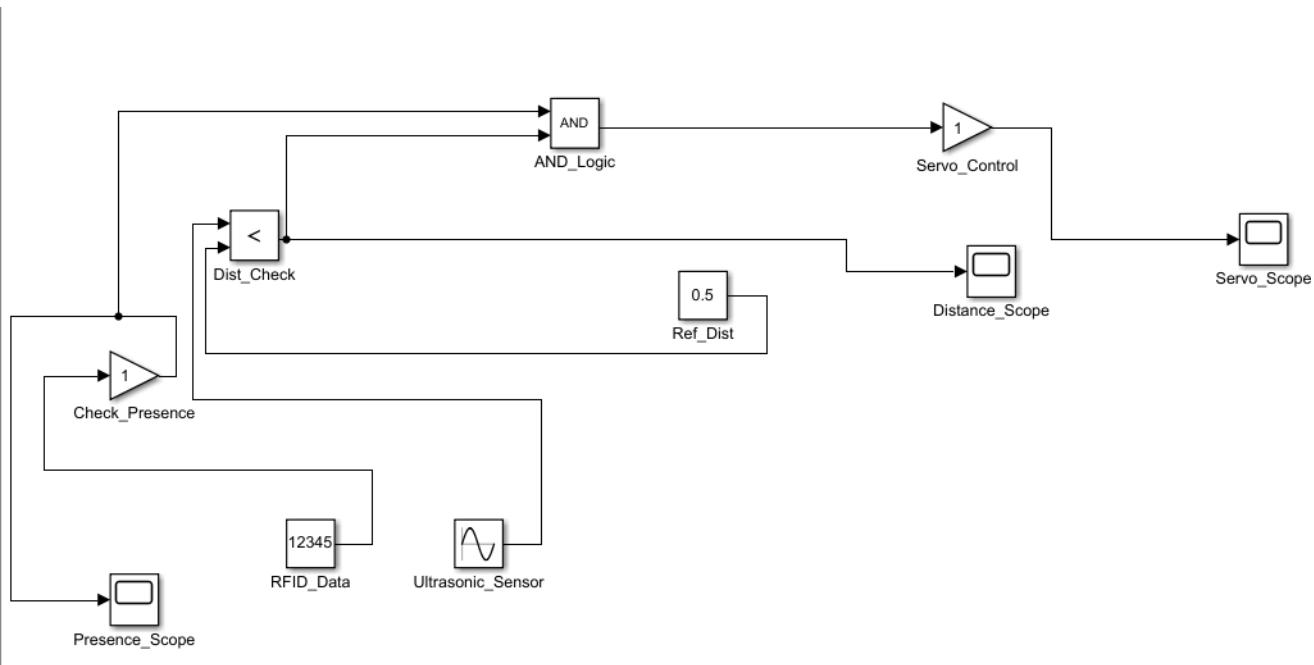


Fig 12.4 Simulare îmbunătățită

### **Distance\_Scope:**

- Graficul din Distance\_Scope arată un semnal binar care indică dacă distanța măsurată de senzorul ultrasonic este mai mică decât distanța de referință (0.5). Semnalul se schimbă între 0 și 1 în funcție de această condiție.
- Aceasta arată funcționarea corectă a blocului Dist\_Check, care compară distanța măsurată cu distanța de referință.

### **Servo\_Scope:**

- Graficul din Servo\_Scope arată semnalul de control pentru servomotor. Acesta arată un semnal binar care indică dacă toate condițiile de prezență și distanță sunt îndeplinite.
- În acest caz, graficul arată un semnal binar, indicând că servomotorul ar trebui să fie activat și dezactivat în funcție de condițiile definite de blocurile logice.

### **3. Configurarea hardware**

#### **3.1 Componente necesare**

Configurarea hardware reprezintă o etapă esențială în implementarea unui sistem de control al accesului bazat pe RFID. Acest capitol va detalia pașii necesari pentru a asambla și conecta componentele fizice necesare pentru proiectul de licență. Prin intermediul unei descrieri clare și concise, vom aborda atât conectarea modulului RFID RC522, cât și integrarea altor componente cum ar fi display-ul LCD, senzorul de distanță HC-SR04, și servo motorul SG90. Scopul acestei secțiuni este de a furniza o înțelegere aprofundată a modului în care componentele hardware interacționează între ele pentru a asigura funcționarea optimă a sistemului.

##### Componente necesare

1. Raspberry Pi 5-4gb - Va acționa ca unitate de procesare principală.
2. Modul RFID RC522 - Utilizat pentru citirea și scrierea tag-urilor RFID.
3. Display LCD I2C 16x2 - Pentru afișarea informațiilor relevante utilizatorilor.
4. Senzor de distanță HC-SR04 - Folosit pentru măsurarea distanței și detectarea prezenței.
5. Servo motor SG90 - Pentru controlul fizic al accesului.
6. Conexiuni și cabluri necesare - Pentru asigurarea conectivității între componente.
7. Alimentare specială servo-Vom avea nevoie de o alimentare separată pentru o componentă costisitoare

- **Conexiunea modulului RFID RC522:**

SDA la GPIO 8 (pin 24)  
SCK la GPIO 11 (pin 23)  
MOSI la GPIO 10 (pin 19)  
MISO la GPIO 9 (pin 21)  
IRQ la neconectat (optional)  
GND la GND (pin 6)  
RST la GPIO 25 (pin 22)  
VCC la 3.3V (pin 1)

- **Conexiunea display-ului LCD I2C:**

SDA la GPIO 2 (pin 3)  
SCL la GPIO 3 (pin 5)  
VCC la 5V (pin 4)  
GND la GND (pin 6)

- **Conexiunea senzorului de distanță HC-SR04:**

VCC la 5V (pin 2)  
Trig la GPIO 23 (pin 16)

Echo la GPIO 24 (pin 18)  
GND la GND (pin 9)

- **Conexiunea servo motorului SG90:**

PWM la GPIO 18 (pin 12)  
VCC la 5V (pin 4)  
GND la GND (pin 14)

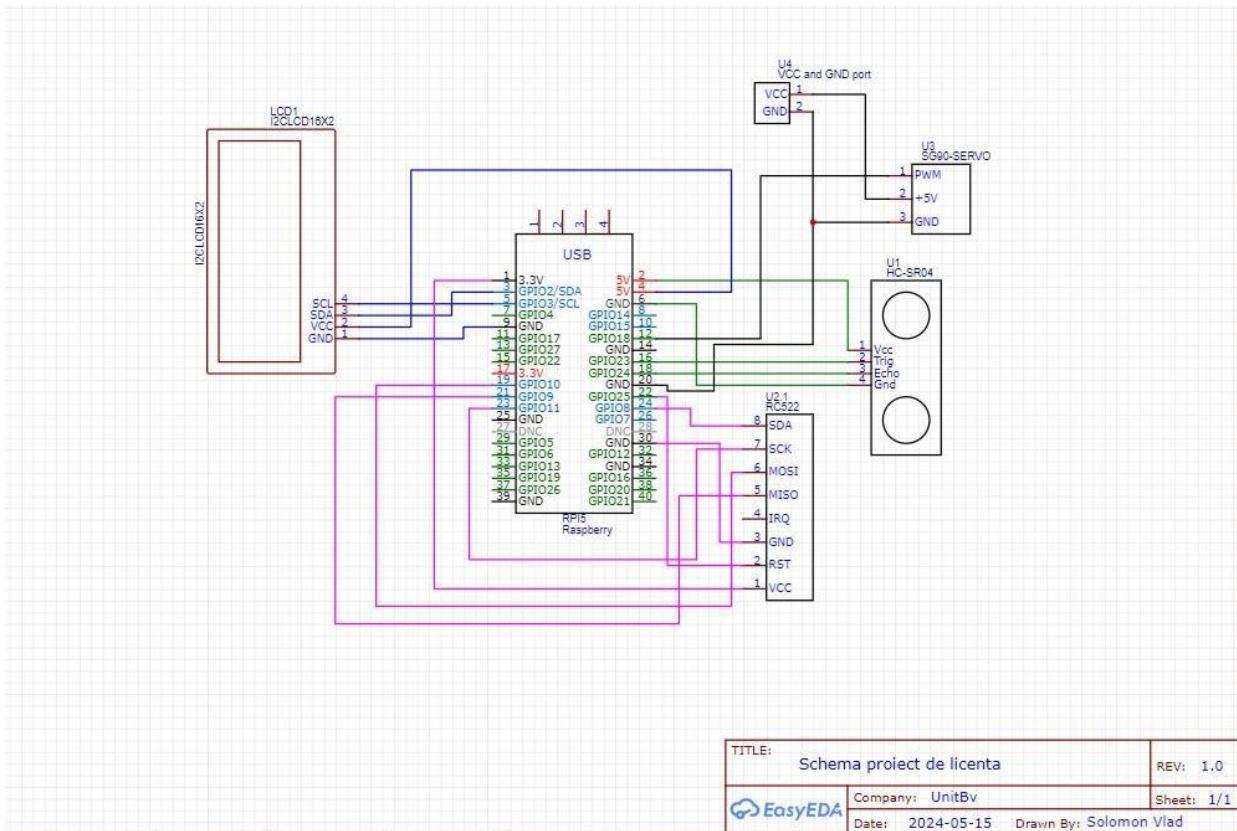


Fig.13 Conexiunile hardware

### 3.2 SDA,SCK,MOSI,MISO,IRQ

În contextul SPI, linia SDA este adesea utilizată pentru a se referi la linia SS (Slave Select). Aceasta linie este crucială pentru stabilirea și controlul comunicației între master și dispozitivele slave. Master-ul este responsabil pentru generarea ceasului și pentru selecția dispozitivului slave prin linia SS.

Master-ul generează un semnal de ceas (SCK), de obicei în intervalul MHz, pentru sincronizarea transferului de date.

În mod normal, linia SS este ridicată (high). Pentru a începe comunicația, master-ul trage linia SS în jos (low) pentru a selecta dispozitivul slave cu care dorește să comunice. Cu fiecare impuls de ceas pe linia SCLK, datele stocate în registrele MOSI și MISO sunt schimbată între master și slave.[10]

După transferul tuturor bițiilor de date (de obicei 8 biți), master-ul ridică din nou linia SS pentru a semnaliza sfârșitul comunicației. Astfel, datele inițiale din master sunt transferate în slave și viceversa, realizându-se o comunicare full-duplex.

Line	Byte 0	Byte 1	Byte 2	To	Byte n	Byte n + 1
MOSI	address 0	address 1	address 2	...	address n	00
MISO	X[1]	data 0	data 1	...	data n – 1	data n

Fig.14 Modul de citire a datelor utilizând protocolul SPI[10]

În tabelul prezentat, se arată ordinea în care byte-urile sunt trimise și primite prin intermediul liniilor MOSI (Master Out Slave In) și MISO (Master In Slave Out).

Primul byte trimis definește atât modul cât și adresa. Master-ul trebuie să trimită primul byte pentru a iniția comunicația. După aceasta, datele sunt transferate conform ordinii specificate în tabel.

Cel mai semnificativ bit trebuie să fie trimis primul, asigurând astfel că datele sunt corect sincronizate între master și slave.

Aceasta este linia pe care master-ul trimit date către slave. Ordinea byte-urilor trimise pe linia **MOSI** este următoarea:

**Byte 0:** Adresă 0 - Acest byte specifică adresa inițială a slave-ului cu care master-ul dorește să comunice. Este primul byte trimis și definește atât modul cât și adresa.

**Byte 1:** Adresă 1 - Al doilea byte trimis, continuând specificarea adresei slave-ului.

**Byte 2:** Adresă 2 - Al treilea byte care continuă specificarea adresei.

**Byte n:** Adresă n - Byte-ul n reprezintă adresa finală a slave-ului.

**Byte n + 1: 00** - Acest byte final indică încheierea secvenței de adrese. Este utilizat pentru a completa transferul de adrese.

#### MISO:

**Byte 0:** - Acest byte este ignorat și nu contează pentru master (de obicei folosit pentru sincronizare).

**Byte 1:** Date 0 - Primul byte de date primit de la slave.

**Byte 2:** Date 1 - Al doilea byte de date primit.

**Byte n:** Date n-1 - Byte-ul n-1 reprezintă penultimul byte de date primit.

**Byte n + 1:** Date n - Byte-ul final de date primit.

Line	Byte 0	Byte 1	Byte 2	To	Byte n	Byte n + 1
MOSI	address 0	data 0	data 1	...	data n – 1	data n
MISO	X[1]	X[1]	X[1]	...	X[1]	X[1]

Fig.14.1 Modul de scriere a datelor utilizând protocolul SPI[10]

**Byte 0:** Adresă 0 - Acest byte specifică adresa inițială a slave-ului cu care master-ul dorește să comunice. Este primul byte trimis și definește atât modul cât și adresa.

**Byte 1:** Date 0 - Primul byte de date ce urmează adresei. Conține prima parte a informației ce trebuie transmisă către slave.

**Byte 2:** Date 1 - Al doilea byte de date. Continuă transferul de informație către slave.

...

**Byte n-1:** Date n-1 - Penultimul byte de date transmis de master către slave.

**Byte n:** Date n - Byte-ul final de date transmis în această secvență.

Ordinea byte-urilor este crucială pentru asigurarea unei comunicări corecte între dispozitivele master și slave. Fiecare byte trimis și primit trebuie să respecte ordinea specificată pentru a preveni erorile de sincronizare și pentru a asigura integritatea

datelor transferate.

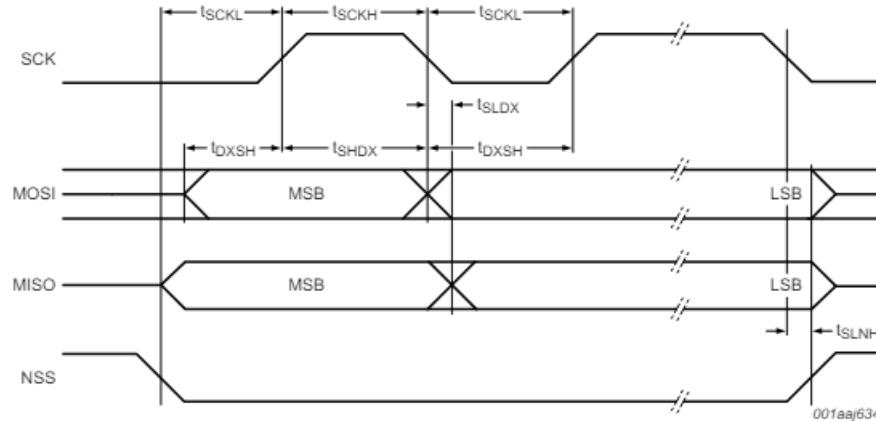


Fig.15 Diagrama de timp pentru SPI

**IRQ-** Pinul IRQ (Interrupt Request) este utilizat pentru a semnala procesorului sau controlerului principal că o anumită condiție a fost îndeplinită de modulul RFID.

Proiectul meu se concentrează pe o implementare simplă și ușor de înțeles a comunicării SPI între modulul RFID și Raspberry Pi. Utilizarea IRQ nu ar fi adus nimic important în plus și ar fi complicat utilizarea modulului.

Alimentarea modulului va fi la 3.3V.

### 3.3 Conectarea dintre Raspberry si Lcd-ul I2C

Un afișaj LCD1602 cu interfață I2C este un dispozitiv care poate prezenta text și caractere pe un ecran cu cristale lichide de 16x2 (16 coloane și 2 rânduri) folosind protocolul I2C. Modulul I2C include un cip PCF8574, care transformă datele seriale I2C în date paralele pentru afișajul LCD.[11]

Modulul I2C traduce semnalele primite de la Raspberry în comenzi pentru afișajul LCD. LCD-ul, cu 16x2 celule, este capabil să afișeze caractere și simboluri. Fiecare celulă este formată dintr-o matrice de 5x8 puncte care pot fi activate sau dezactivate prin aplicarea unei tensiuni electrice. Astfel, diferite combinații de puncte pot fi aprinse sau stinse pentru a crea diverse caractere și simboluri pe ecran.[13]

# Slave Address

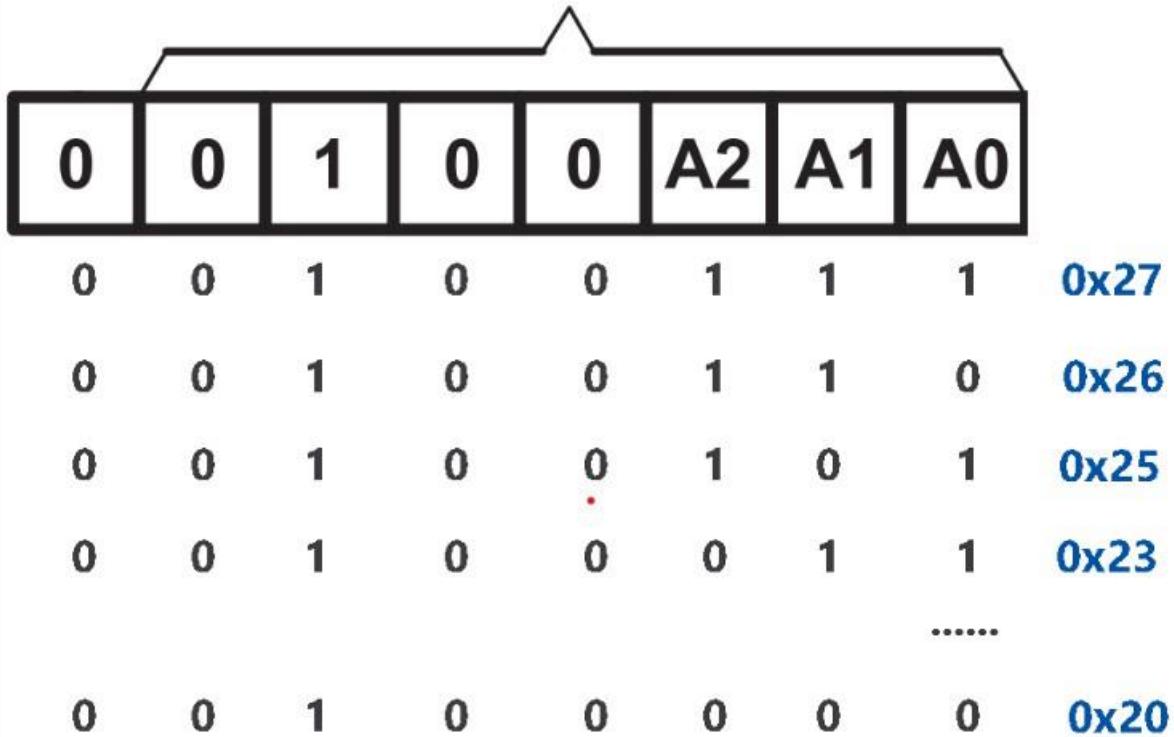


fig 16.Adresa I2C[11]

Adresa dispozitivului este compusă dintr-un prefix fix de 5 biți, urmat de trei biți configurabili (A2, A1, A0) care permit selectarea diferitelor dispozitive slave. Acest lucru este util pentru a avea mai multe dispozitive I2C conectate la același bus fără conflicte de adresare. În tabelul asociat, fiecare combinație a celor trei biți configuraibili corespunde unei adrese I2C unice (de exemplu, 0x27, 0x26, 0x25 etc.). Adresa completă este utilizată de master pentru a comunica cu un anumit dispozitiv sclav pe bus-ul I2C.

Lcd-ul dispune și de un potențiometru care permite să se regleze contrastul, este un detaliu foarte important deoarece la început probabil acesta nu va afișa nimic deoarece contrastul nu este reglat.

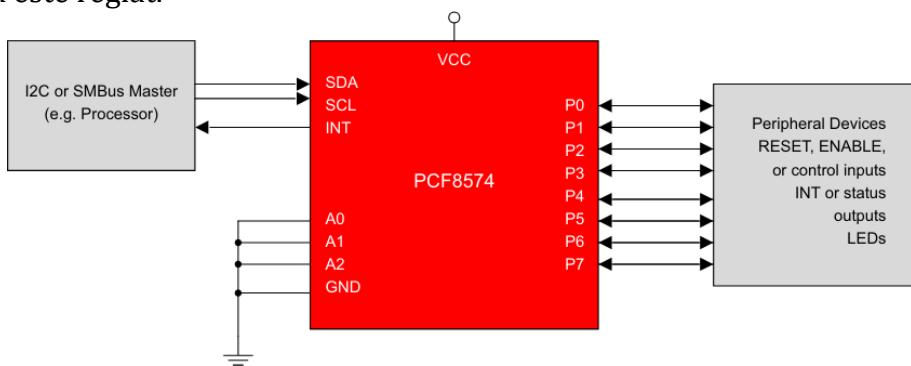


Fig.17 Schema de Conexiuni a Expanderului I/O PCF8574 pentru Comunicare I2C.[12]

Dispozitivul PCF8574 oferă o soluție de extindere a I/O-urilor la distanță pentru majoritatea familiilor de microcontrolere prin intermediul interfeței I2C, care utilizează liniile serial clock (SCL) și serial data (SDA). Dispozitivul dispune de un port I/O cvasi-bidirectional de 8 biți (P0-P7), inclusiv ieșiri cu blocare și capacitate de curent mare pentru a actiona direct LED-uri. Fiecare I/O cvasi-bidirectional poate fi utilizat ca intrare sau ieșire fără a necesita un

semnal de control al direcției datelor. La pornirea dispozitivului, I/O-urile sunt setate la high. În acest mod, doar o sursă de curent către VCC este activă.[12]

Designul protocolului I2C a fost împărțit în 3 niveluri în această metodă. După cum este descris în figura 18, aceste 3 niveluri, de la cel mai jos la cel mai înalt, sunt: nivelul de protocol (PRL), nivelul de semnal (SIL) și nivelul de interfață (INL).

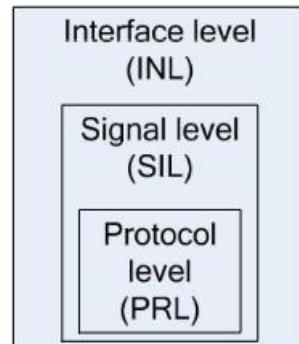


Fig.18 Nivelele I2C[14]

Nivelul de protocol (PRL) reprezintă baza și se ocupă de regulile fundamentale de comunicare. Nivelul de semnal (SIL) gestionează semnalele specifice și comunicarea efectivă pe magistrală. Nivelul de interfață (INL) reprezintă cel mai înalt nivel și se ocupă de interacțiunile între dispozitivele master și slave.[14]

### 3.4 Conectarea dintre Raspberry Pi și senzorul ultrasonic HC-SR 04

Senzorul ultrasonic HC-SR04 măsoară distanța până la obiecte utilizând tehnologia sonar, asemănător cu liliecii și delfinii. Aceasta permite detectarea distanțelor fără contact, oferind o precizie și stabilitate ridicată în citiri, într-un format ușor de folosit. Senzorul poate detecta distanțe între 2 cm și 400 cm (1 inch până la 13 picioare). Funcționează bine chiar și în condiții de lumina solară directă sau în prezența materialelor negre, spre deosebire de detectoarele Sharp, deși poate întâmpina dificultăți în detectarea materialelor moi, cum ar fi pânza. Acest dispozitiv include un modul complet de transmisie și recepție ultrasonică.[15]

Caracteristici:

- Sursa de alimentare: +5V DC
- Currentul de repaus: <2mA
- Currentul în timpul funcționării: 15mA
- Unghi eficient: <15°
- Interval de măsurare: 2cm – 400 cm (1" – 13ft)
- Precizie: 0,3 cm
- Unghi de măsurare: 30 grade
- Lățimea pulsului de intrare pentru declanșare: 10uS
- Dimensiuni: 45mm x 20mm x 15mm[15]



Fig.19 Senzorul ultrasonic[15]

**Funcționarea senzorului:** Pentru a începe măsurarea, pinul TRIG al senzorului HC-SR04 trebuie să primească un impuls de înaltă tensiune (5V) timp de cel puțin 10 microsecunde. Acest impuls inițiază trimiterea unui semnal ultrasonic format din 8 cicluri la frecvența de 40kHz. După ce senzorul detectează ecoul ultrasunetului reflectat, acesta setează pinul ECHO la înaltă tensiune (5V) și menține această stare pentru o perioadă de timp proporțională cu distanța măsurată. Pentru a determina distanța, se măsoară durata impulsului de pe pinul ECHO.

Formula pentru calcularea distanței este:

Time = Width of Echo pulse, in uS (micro second)

- Distance in centimeters = Time / 58
- Distance in inches = Time / 148
- Or you can utilize the speed of sound, which is 340m/s

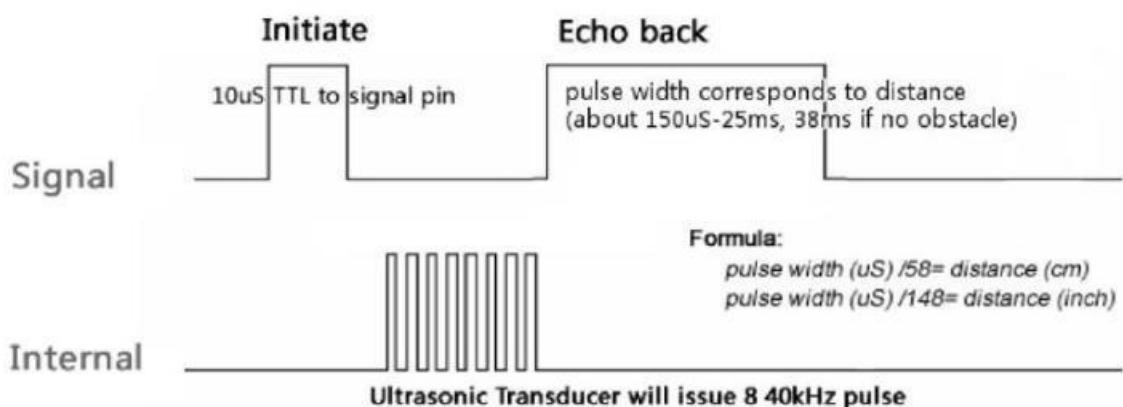


Fig 20.Formele de undă pentru funcționarea senzorului[15]

Se trimit un impuls de 10 microsecunde pe pinul TRIG. După trimiterea impulsului, senzorul așteaptă ecoul ultrasunetului. Durata impulsului primit pe pinul ECHO este proporțională cu distanța măsurată (aproximativ 150 $\mu$ s - 25ms, 38ms dacă nu este obstacol).

## **4. Raspberry PI 5**

 Raspberry Pi 5	BCM2712	4GB 8GB	40-pin GPIO header	<ul style="list-style-type: none"><li>• 2 × micro HDMI</li><li>• 2 × USB 2.0</li><li>• 2 × USB 3.0</li><li>• 2 × CSI camera/DSI display ports</li><li>• single-lane PCIe FFC connector</li><li>• UART connector</li><li>• RTC battery connector</li><li>• four-pin JST-SH PWM fan connector</li><li>• PoE+-capable Gigabit Ethernet (1Gb/s)</li><li>• 2.4/5GHz dual-band 802.11ac Wi-Fi 5 (300Mb/s)</li><li>• Bluetooth 5, Bluetooth Low Energy (BLE)</li><li>• microSD card slot</li><li>• USB-C power (5V, 5A (25W) or 5V, 3A (15W) with a 600mA peripheral limit)</li></ul>
---	---------	------------	--------------------	--

fig.21 Specificații Raspberry

Raspberry Pi 5 4GB reprezintă un avans semnificativ în lumea microcomputerelor, fiind conceput atât pentru entuziaștii de tehnologie, cât și pentru profesioniști. Acest dispozitiv compact și accesibil oferă performanțe impresionante și o gamă variată de funcționalități, făcându-l ideal pentru o multitudine de proiecte și aplicații.

Echipat cu un procesor ARM Cortex-A76 quad-core de 1.8 GHz și 4GB de memorie RAM LPDDR4x-3200, Raspberry Pi 5 4GB asigură o putere de procesare robustă și eficientă. Acest hardware puternic permite rularea fluidă a aplicațiilor complexe și suportă multitasking-ul fără probleme, fiind ideal pentru dezvoltarea de prototipuri, proiecte de automatizare și aplicații IoT.[16][17]

În ceea ce privește conectivitatea, Raspberry Pi 5 4GB este dotat cu o varietate de opțiuni care îl fac extrem de versatil. Include un port Ethernet Gigabit pentru conexiuni rapide la rețea, dual-band Wi-Fi (2.4GHz și 5GHz) pentru conectivitate wireless stabilă, și Bluetooth 5.0 pentru comunicare cu dispozitive periferice. Cele patru porturi USB (două USB 3.0 și două USB 2.0) permit conectarea unei game largi de dispozitive externe, de la tastaturi și mouse-uri la stocare externă și module de expansiune.[16][17]

Suportul pentru grafică este asigurat de GPU-ul VideoCore VI, care permite redarea video 4K la 60Hz prin intermediul celor două porturi micro HDMI. Acest lucru face din Raspberry Pi 5 4GB o soluție excelentă pentru proiecte multimedia și aplicații care necesită ieșiri video de înaltă calitate. De asemenea, jack-ul audio-video de 3.5mm oferă posibilități suplimentare pentru conexiuni analogice.

O caracteristică esențială a Raspberry Pi 5 4GB este prezența celor 40 de pini GPIO, care oferă posibilități extinse de conectare și control pentru diverse senzori și module. Aceste pini suportă protocoale de comunicație precum I2C, SPI și UART, facilitând integrarea cu o varietate de dispozitive externe.[18]

Alimentarea dispozitivului se realizează printr-un conector USB-C, necesită o sursă de alimentare de 5V/3A, asigurând un consum eficient de energie, ideal pentru aplicații mobile și integrate. Suportul pentru alimentarea prin Ethernet (PoE) adaugă o flexibilitate suplimentară în configurarea și utilizarea dispozitivului.[18]

În concluzie, Raspberry Pi 5 4GB este o platformă puternică și versatilă, perfectă pentru o gamă largă de aplicații, de la educație și prototipare până la proiecte IoT și automatizări. Performanțele sale ridicate, împreună cu conectivitatea extinsă și suportul pentru numeroase extensii, îl fac un instrument indispensabil pentru orice dezvoltator sau entuziașt al tehnologiei.[18]

## 5. Baza de date/MariaDB

MariaDB este un sistem de gestionare a bazelor de date relaționale (RDBMS) derivat din MySQL, dezvoltat pentru a oferi performanțe și securitate îmbunătățite, precum și compatibilitate deplină cu MySQL. Este cunoscut pentru capacitatea să de a gestioneaza volume mari de date și tranzacții complexe într-un mod eficient și fiabil.

MariaDB a fost creată de Michael "Monty" Widenius, unul dintre fondatorii MySQL, în 2009. Denumirea "MariaDB" provine de la numele fiicei sale, Maria. După achiziția MySQL de către Oracle Corporation, MariaDB a fost dezvoltată ca o versiune open-source pentru a menține independență și transparentă în dezvoltarea bazelor de date.[20]

- MariaDB este compatibilă cu MySQL la nivel de API și protocol, ceea ce înseamnă că aplicațiile și instrumentele care funcționează cu MySQL pot fi utilizate și cu MariaDB fără modificări majore. Oferă optimizări semnificative pentru stocarea și interogarea datelor, suportând replicarea master-master și master-slave pentru configurații de înaltă disponibilitate și echilibrare a încărcării.[19]
- Include stocarea de date în coloane (ColumnStore), stocarea temporară (Temporary Tables) și suport pentru motoare de stocare multiple, cum ar fi InnoDB, MyISAM și Aria, fiecare având avantaje specifice în funcție de tipul de aplicație. MariaDB oferă îmbunătățiri semnificative de securitate, inclusiv autentificarea plugin-urilor, criptarea datelor în tranzit și la rest, și controale granulare ale permisiunilor utilizatorilor.[19]

MariaDB permite crearea de proceduri stocate și funcții definite de utilizator, care pot executa operații complexe direct pe serverul de baze de date. Acest lucru reduce traficul de rețea și îmbunătățește performanța aplicațiilor. Procedurile stocate sunt segmente de cod SQL care pot fi salvate și reutilizate, în timp ce UDF-urile sunt funcții personalizate create pentru a extinde funcționalitățile standard SQL.[21]

Partiționarea permite distribuirea tabelelor mari pe mai multe locații fizice, îmbunătățind astfel performanța și gestionarea datelor. MariaDB suportă diverse metode de partiționare, inclusiv partiționarea pe intervale, pe valori hash și pe valori list. Aceasta ajută la optimizarea interogărilor și la administrarea eficientă a datelor.

Optimizatorul de interogări din MariaDB este proiectat pentru a îmbunătăți performanța interogărilor SQL. Acesta utilizează diverse tehnici de optimizare, cum ar fi utilizarea de indexi, analizarea planurilor de execuție și optimizarea subinterogărilor. Acest lucru ajută la executarea mai rapidă a interogărilor complexe și la reducerea timpului de răspuns.

MariaDB include motorul de stocare Aria, care este proiectat pentru a fi un substitut modern pentru MyISAM. Aria oferă performanțe îmbunătățite, recuperare automată în caz de eroare și suport pentru tranzacții ACID (atomicitate, coherență, izolare, durabilitate). Este ideal pentru aplicații care necesită performanță ridicată și fiabilitate.

MariaDB oferă numeroase îmbunătățiri de securitate, inclusiv autentificare bazată pe plugin-uri, criptare a datelor atât în tranzit, cât și la rest, și controale granulare ale permisiunilor utilizatorilor. Aceste caracteristici asigură că datele sunt protejate împotriva accesului neautorizat și a atacurilor cibernetice.

MariaDB include un set extensiv de funcții matematice și statistice care permit utilizatorilor să efectueze calcule complexe direct în interogările SQL. Aceste funcții sunt utile pentru aplicații de analiză a datelor, raportare și business intelligence.[21]

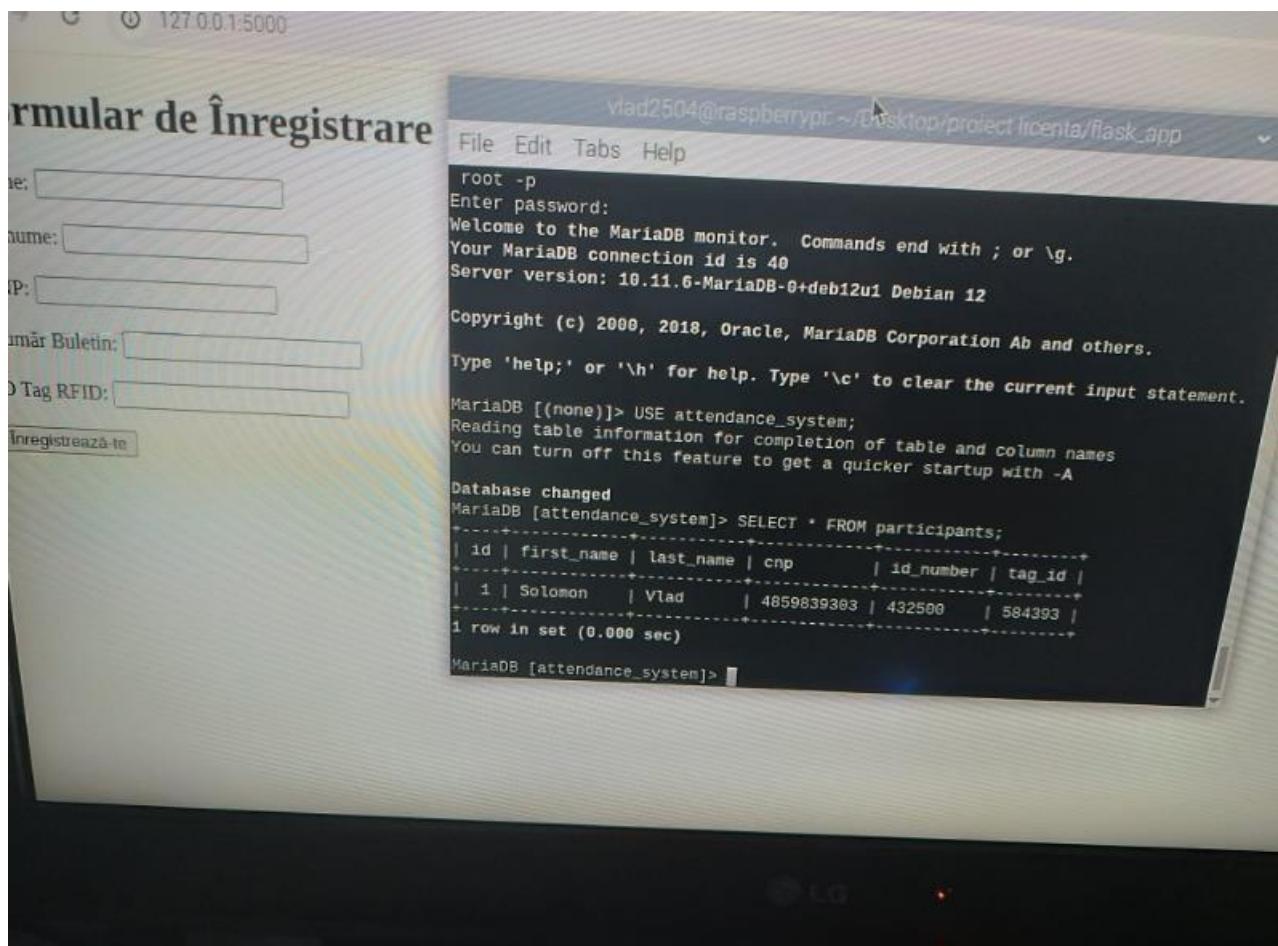


Fig 21.1 Primele date stocate

## 6. Securitate

Securitatea în sistemele RFID este esențială pentru protejarea datelor sensibile și prevenirea atacurilor cibernetice. Această secțiune va explora tehnici avansate și practici de securitate care pot fi implementate în proiectul tău folosind tehnologia RFID. Aceste tehnici includ autentificarea, criptarea datelor, controlul accesului, auditul și monitorizarea activităților, precum și alte măsuri de protecție.

### Autentificarea Utilizatorilor

Autentificarea utilizatorilor reprezintă un pas crucial în asigurarea securității unui sistem RFID. Aceasta poate fi realizată prin diverse metode, cum ar fi autentificarea bazată pe parolă, unde utilizatorii trebuie să furnizeze un nume de utilizator și o parolă pentru a accesa sistemul. De asemenea, se poate utiliza autentificarea prin certificate digitale, care asigură o conexiune securizată și autentificare bidirectională. Autentificarea multifactor (MFA), care combină mai multe metode de autentificare, cum ar fi parole, token-uri de securitate și biometrie, oferă un nivel suplimentar de securitate.

### Criptarea Datelor

Criptarea este vitală pentru protejarea datelor transmise și stocate de sistemele RFID. Criptarea în tranzit folosește SSL/TLS pentru a proteja datele transmise între tag-urile RFID și cititoare, prevenind interceptarea și manipularea datelor. Criptarea la rest asigură protecția datelor stocate în bazele de date și pe tag-urile RFID. Algoritmi de criptare puternici, cum ar fi AES și RSA, sunt utilizati pentru a asigura confidențialitatea și integritatea datelor.

### Controlul Accesului

Controlul accesului garantează că doar utilizatorii autorizați pot accesa și modifica datele RFID. Acest lucru este realizat prin permisiuni bazate pe roluri (RBAC), care definesc și gestionez permisiunile în funcție de rolurile utilizatorilor în sistem. De asemenea, accesul fizic la cititoarele RFID și alte echipamente critice trebuie restricționat. În plus, se poate limita accesul la sistemul RFID doar pentru adrese IP de încredere printr-o listă de acces bazată pe IP.

### Audit și Monitorizare

Auditul și monitorizarea activităților sunt esențiale pentru detectarea și prevenirea accesului neautorizat. Logurile de securitate înregistrează toate activitățile și accesările sistemului RFID, facilitând detectarea și investigarea incidentelor de securitate. Monitorizarea în timp real utilizează instrumente pentru a urmări activitățile și a detecta comportamente suspecte. Auditul periodic verifică conformitatea cu politicile de securitate și identifică eventualele vulnerabilități.

### Protecția Datelor la Nivelul Middleware

Middleware-ul joacă un rol crucial în securizarea sistemelor RFID, gestionând corect datele între cititoare și backend. Aceasta poate filtra și valida datele înainte de a le transmite către backend, asigurând că doar datele legitime sunt procesate. Pentru a preveni atacurile de

tip „replay”, middleware-ul utilizează nonce-uri și timestamp-uri. De asemenea, criptarea comunicațiilor middleware asigură că toate datele transmise între cititor și backend sunt protejate împotriva interceptării.

## Bune Practici de Configurare

Configurarea corectă a echipamentelor și a sistemelor software este esențială pentru securitate. Menținerea sistemului RFID la zi cu ultimele patch-uri și actualizări de securitate este vitală. Utilizarea de parole complexe și schimbarea periodică a acestora contribuie la securitatea sistemului. Implementarea unui firewall pentru a controla traficul de rețea și protejarea sistemului RFID împotriva accesului neautorizat sunt esențiale. Realizarea de backup-uri regulate și testarea procedurilor de recuperare asigură continuitatea datelor în caz de incident de securitate.

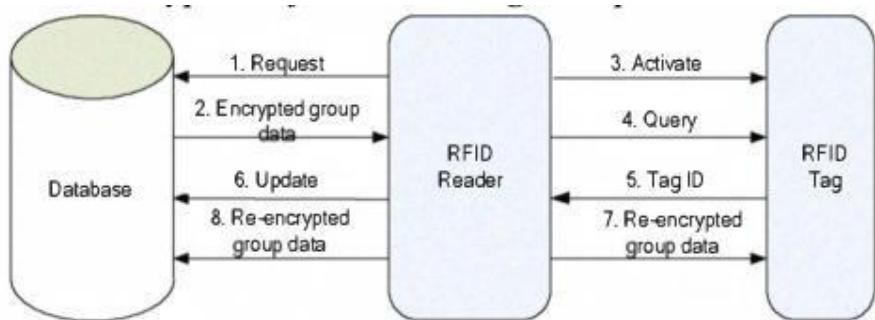


Fig.22 Reîncriptarea datelor pentru RFID[25]

Re-encriptarea este o tehnică esențială pentru asigurarea securității datelor în sistemele RFID. Aceasta metodă permite prevenirea urmăririi și identificării neautorizate a tagurilor RFID, oferind un nivel suplimentar de protecție împotriva atacurilor. Iată cum poate fi implementată re-encriptarea în proiectul tău RFID, bazându-ne pe studiile și protocolele de securitate existente.

În aplicațiile RFID, datele transmise între cititor și taguri sunt vulnerabile la interceptare și atacuri de urmărire. Pentru a rezolva aceste probleme de securitate, se poate utiliza un protocol de securitate bazat pe re-encriptare. Acest protocol re-encriptează ID-ul tagului de fiecare dată când este citit, prevenind astfel atacatorii să identifice și să urmărească tagurile RFID.

### 1. Re-encriptarea Datelor Grupului de Taguri RFID:

- Cititorul solicită datele criptate ale grupului specific de la baza de date.
- Cititorul primește datele criptate ale grupului.
- Cititorul activează toate tagurile din grupul specific.
- Cititorul interoghează tagurile active.
- Se folosește un algoritm de anti-coliziune bazat pe grup pentru a identifica toate tagurile răspunzătoare.
  - Se actualizează informațiile legate de taguri în baza de date.
  - Cititorul generează un număr întâmplător și re-encriptează datele.
  - Noul cifru este scris în tagurile identificate și datele grupului sunt actualizate în baza de date.

## 2. Analiza Securității:

- Protocolul permite utilizarea de taguri RFID de cost redus, deoarece operațiunile criptografice sunt realizate de cititor, iar tagurile nu necesită capacitați de calcul.
- Capacitate anti-urmărire: Chiar dacă ID-urile tagurilor sunt interceptate, atacatorul nu poate determina asocierea între mărfuri și expeditori specifici, prevenind astfel urmărirea mărfurilor.
- Capacitate anti-tamper: Datele grupului sunt criptate și foarte greu de decriptat fără cheia corectă. Selectarea unei frecvențe adecvate de re-criptare poate preveni eficient modificarea ilegală a datelor grupului.
  - Capacitate anti-contrafacere: Rezultatele re-criptării sunt unice pentru fiecare grup de taguri. Dacă datele grupului interceptate au fost re-criptate, un cititor neautorizat nu poate activa tagurile corespunzătoare.

## 7. Implementare

Pentru proiectul de sistem de prezență bazat pe Raspberry Pi, am ales să folosesc un card de memorie MicroSD Kingston Canvas Select Plus de 128GB. Acest card SD este esențial pentru stocarea sistemului de operare, a aplicațiilor și a datelor colectate de sistem. Alegerea acestui card SD specific a fost motivată de mai multe considerente tehnice și de performanță. Principalele criterii au fost viteza și capacitatea de stocare suficient de mare care să-mi permită mai multe îmbunătățiri sau proiecte viitoare.

Am ales Raspberry Pi OS deoarece este o distribuție bazată pe Debian optimizată special pentru hardware-ul Raspberry Pi. Acest sistem de operare asigură o compatibilitate excelentă cu toate componente și librăriile necesare pentru implementarea proiectului.

**Versiunea** pe 32 de biți (x32) a fost aleasă pentru a asigura o compatibilitate mai largă cu diverse librării și drivere, precum și pentru a beneficia de un consum de memorie mai redus. În plus a fost modificată de la 64 la 32 după ce am sesizat că versiunea de 32 de biți se mișcă mai bine pe varianta mea de 4gb Ram și este compatibilă cu mai multe librării.

Python a fost ales datorită simplității sale și a suportului extensiv pentru diverse module și librării. Este foarte bine suportat pe Raspberry Pi și are numeroase librării pentru gestionarea componentelor hardware. În plus am vrut să lucrez cu el pentru aprofunda cunoștințele în acest limbaj.

Primul lucru pe care l-am făcut după ce am intrat în sistemul de operare al Raspberry Pi a fost să mă asigur că sistemul este actualizat. Acest pas este esențial pentru a beneficia de cele mai recente pachete și patch-uri de securitate. Iată comenzi folosite:

```
sudo apt-get update  
sudo apt-get upgrade
```

Pentru a utiliza corect modulul RFID și ecranul LCD, a fost necesar să activez interfețele SPI și I2C ale Raspberry Pi. Iată comanda utilizată.

```
sudo raspi-config
```

În meniul de configurare, am selectat opțiunile de interfețe (Interfacing Options) și am activat SPI și I2C.

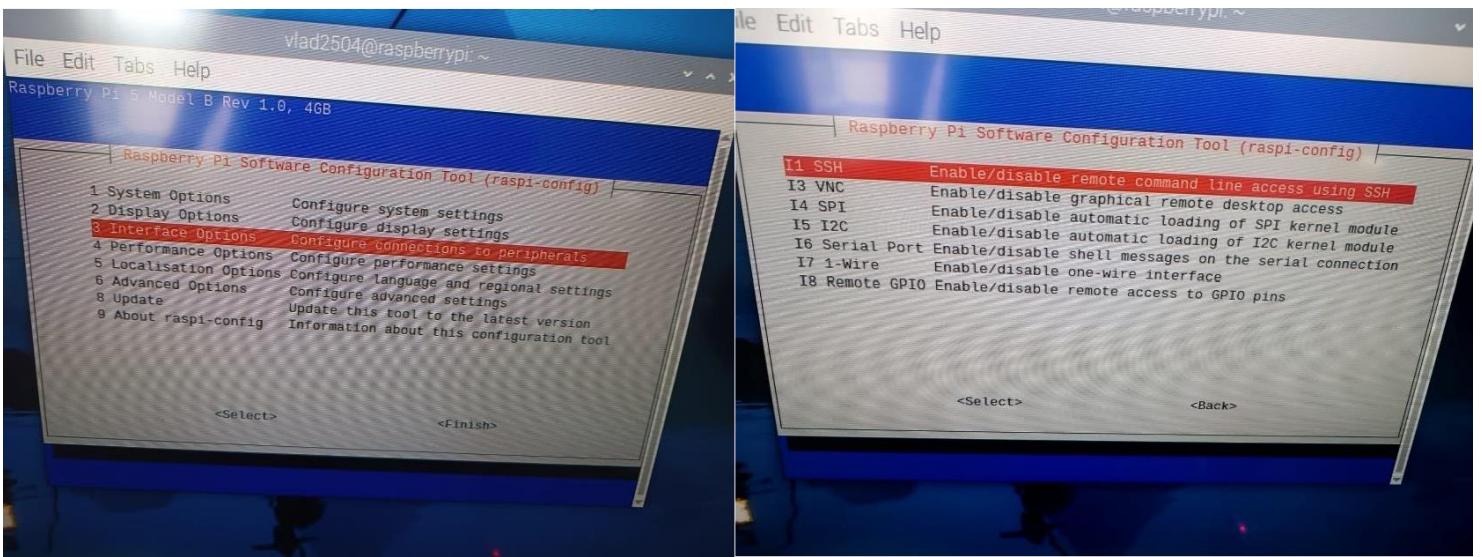


fig.23 Interfață pentru raspi-config

A urmat să instalez diverse biblioteci în mediul global pentru a mă asigura că orice biblioteca de care am nevoie este instalată, ulterior mi-am creat un mediu virtual în care am făcut aceleași configurații. Însă înainte de asta, am vrut să mă asigur că placa este la temperaturi decente.

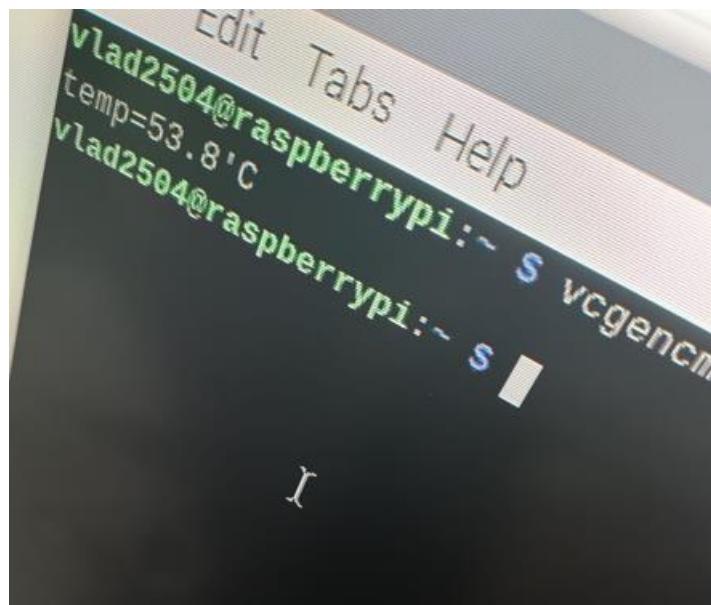


Fig.24 Temperaturi Raspberry

Prin crearea unui mediu virtual, am izolat dependențele proiectului de cele ale sistemului global Python. Acest lucru este important pentru a evita conflictele dintre pachete.

Temperaturile au fost mai mult decât decente pe parcursul la toată utilizarea pe parcursul proiectului. Bineînțeles asta s-a întâmplat și datorită faptului că am adăugat și un cooler.

După ce am avut probleme referitoare la librăria Rpi.Gpio, anume, am întâmpinat probleme în legătură cu librăria modulului deoarece pe lângă problemele de conflict care se întâmplau dacă lucrăm cu ambele librării în paralel deoarece gpio-urile erau folosite, la un moment dat aveam și eroarea "Cannot determine SOC peripheral base address". Aceasta a fost cea mai întâlnită eroare pe parcursul proiectului sub diferite forme, în primele două ture am reușit să o rezolv făcând niște simple update-uri la sistem. Dar nu a fost o soluție permanentă, la urmatoarele update-uri revenind problemele.

Am fost nevoie să găsesc o soluție permanentă, aceasta venind din partea librăriilor, după mai multe research-uri am descoperit că problema de fapt era legată de librăria GPIO care este curpinsă și în librăria pentru RC522. Clasica librărie care era compatibilă cu Raspberry 4 nu era compatibilă și cu 5, deci m-am orientat către GPIO zero, care este o librărie compatibilă, așa că am căutat și pe github până am găsit o librărie pentru R255, care folosește GPIO zero iar codurile pentru RC au început din nou să ruleze.

```
[myenv] vlad2504@raspberrypi:~/Desktop/project licenta $ python3 test_rfid.py
Traceback (most recent call last):
File "/home/vlad2504/Desktop/project licenta/test_rfid.py", line 5, in <module>
    reader = SimpleMFRC522()
              ^^^^^^^^^^^^^^
File "/home/vlad2504/Desktop/project licenta/myenv/lib/python3.11/site-packages/
mfrc522/SimpleMFRC522.py", line 14, in __init__
    self.READER = MFRC522()
                  ^^^^^^^^
File "/home/vlad2504/Desktop/project licenta/myenv/lib/python3.11/site-packages/
mfrc522/MFRC522.py", line 151, in __init__
    GPIO.setup(pin_rst, GPIO.OUT)
RuntimeError: Cannot determine SOC peripheral base address
[myenv] vlad2504@raspberrypi:~/Desktop/project licenta $
```

Fig.25 Soc error

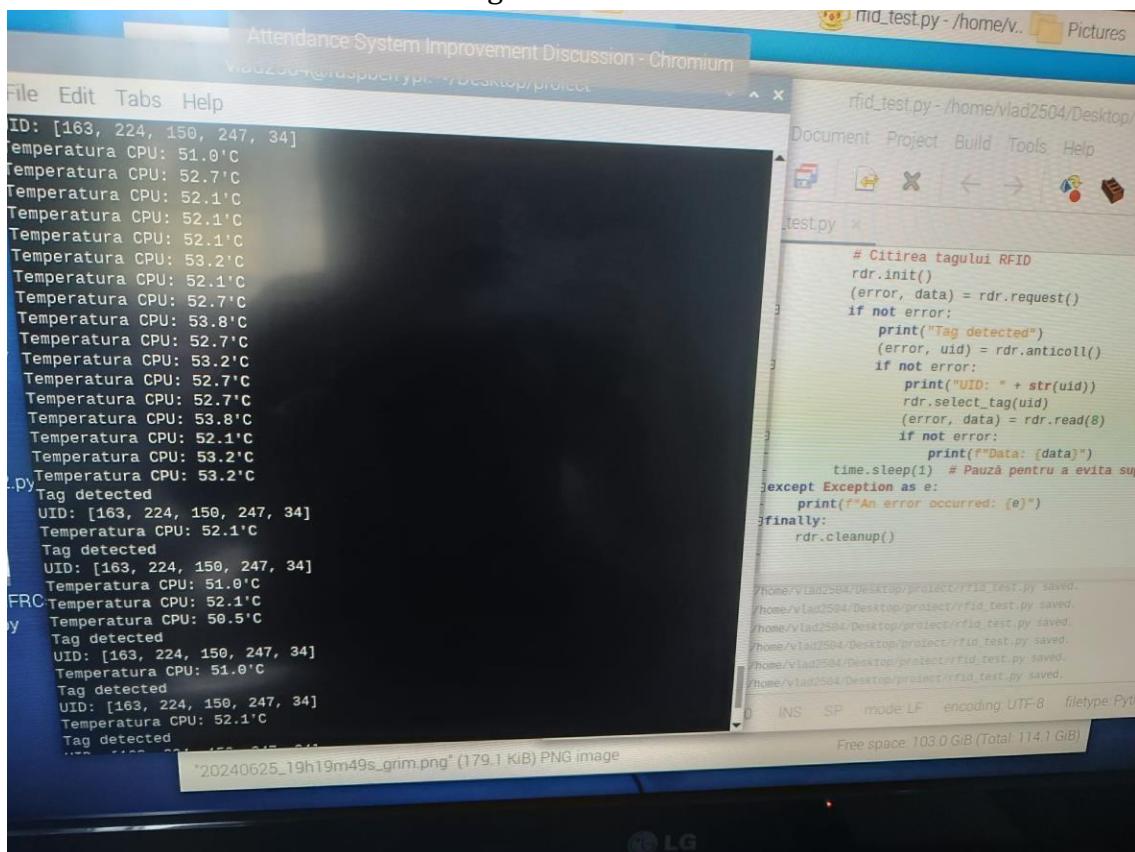


Fig.26 Temperaturile în momentul citirii tagurilor

Figure 1

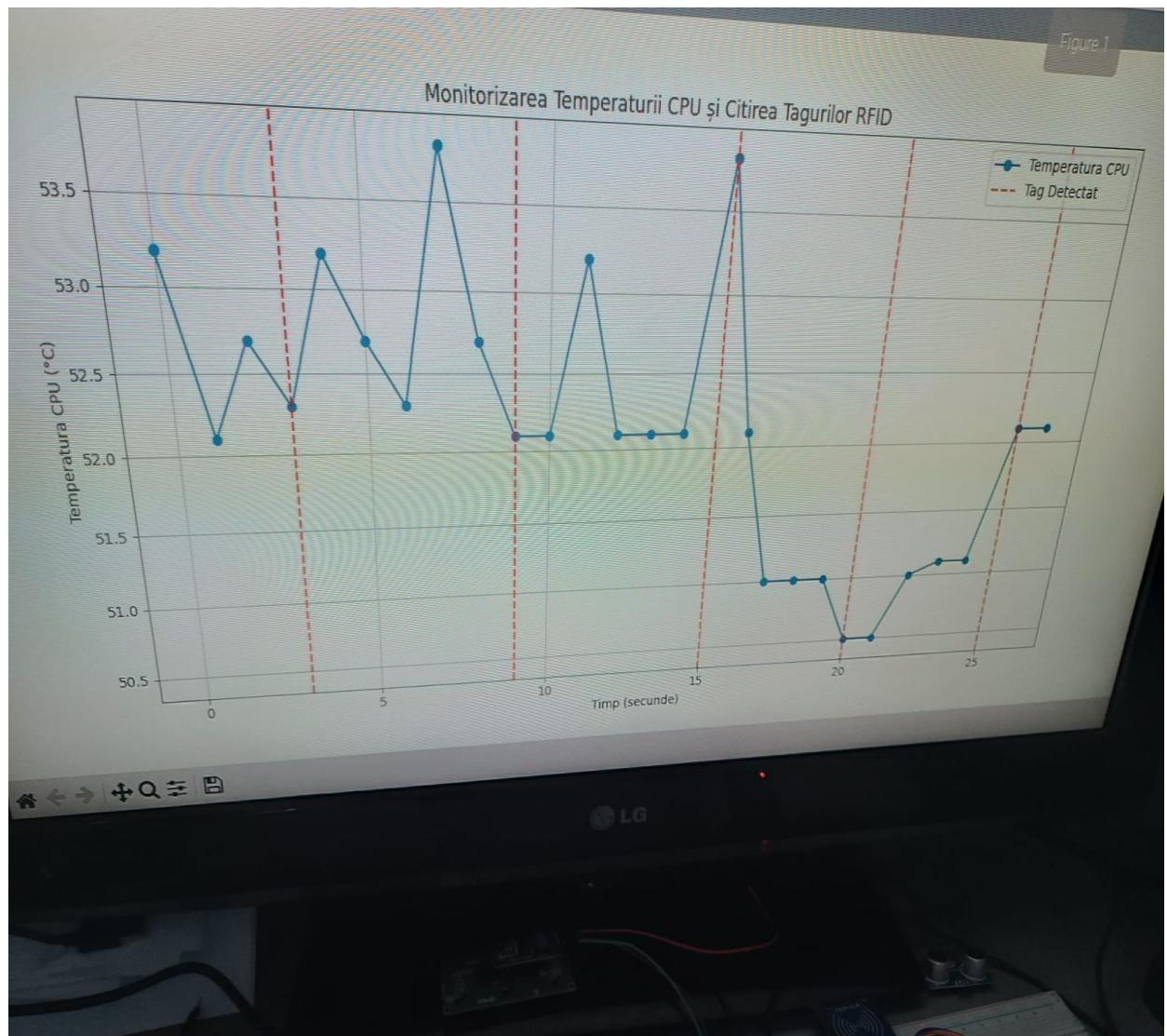


Fig.27 Grafic cu valorile monitorizării temperaturii Cpu la la momente statice si la momente când tagul este detectat

```

g detected
D: [35, 158, 154, 247, 208]
Timp de citire: 0.007918 secunde
Viteza de transfer: 2020.62 bytes/secundă
Tag detected
ID: [35, 158, 154, 247, 208]
Timp de citire: 0.008397 secunde
Viteza de transfer: 1905.47 bytes/secundă
Tag detected
UID: [35, 158, 154, 247, 208]
Timp de citire: 0.007724 secunde
Viteza de transfer: 2071.58 bytes/secundă
Tag detected
UID: [35, 158, 154, 247, 208]
Timp de citire: 0.008232 secunde
Viteza de transfer: 1943.72 bytes/secundă
Tag detected
UID: [35, 158, 154, 247, 208]
Timp de citire: 0.008277 secunde
Viteza de transfer: 1933.08 bytes/secundă
Tag detected
UID: [35, 158, 154, 247, 208]
Timp de citire: 0.008315 secunde
Viteza de transfer: 1924.16 bytes/secundă
Tag detected
UID: [35, 158, 154, 247, 208]
Timp de citire: 0.008438 secunde
Viteza de transfer: 1896.21 bytes/secundă

```

```

24     def measure_transfer_speed(data_length):
25         return data_length / transfer_time
26
27     try:
28         while True:
29             read_time = measure_read_speed()
30             if read_time > 0:
31                 print(f"Timp de citire: {read_time} secunde")
32             data_length = 16 # bytes
33             transfer_speed = measure_transfer_speed(data_length)
34             print(f"Viteza de transfer: {transfer_speed} bytes/secundă")
35             time.sleep(1) # Pauză pentru a evita să se scrie în terminal prea mult
36
37     except KeyboardInterrupt:
38         print("\nProgram interrupted")
39
40     finally:
41         rdr.cleanup()

```

line: 31 / 41 col: 12 sel: 0 INS SP MOD mode: LF encoding: UTF-8

Fig.28 Timpul de citire si viteza de transfer RFID/implementare cod

Temperatura CPU variază între aproximativ 51.5°C și 53.5°C pe parcursul perioadei de monitorizare. Aceste fluctuații sunt normale și sunt influențate de sarcinile procesorului, în special în timpul citirilor RFID.

Citirile RFID sunt indicate prin linii punctate roșii. Acestea arată că citirile au loc la intervale regulate și sunt relativ frecvente. Se observă că după fiecare citire a unui tag RFID, există o mică variație în temperatura CPU. Acest lucru sugerează că citirea tagurilor implică un consum suplimentar de resurse CPU, ceea ce duce la creșteri temporare ale temperaturii.

În jurul timpului de 14-15 secunde, se observă un spike semnificativ în temperatura CPU, urmat de o scădere abruptă. Acest spike ar putea fi cauzat de o sarcină intensă temporară pe procesor, posibil legată de procesarea citirii RFID.

În ciuda fluctuațiilor și a spike-urilor ocazionale, temperatura CPU rămâne în intervale acceptabile pentru funcționarea normală a Raspberry Pi.

Nu se observă supraîncălzire prelungită, ceea ce sugerează că sistemul gestionează bine sarcina citirilor RFID fără a compromite stabilitatea generală.

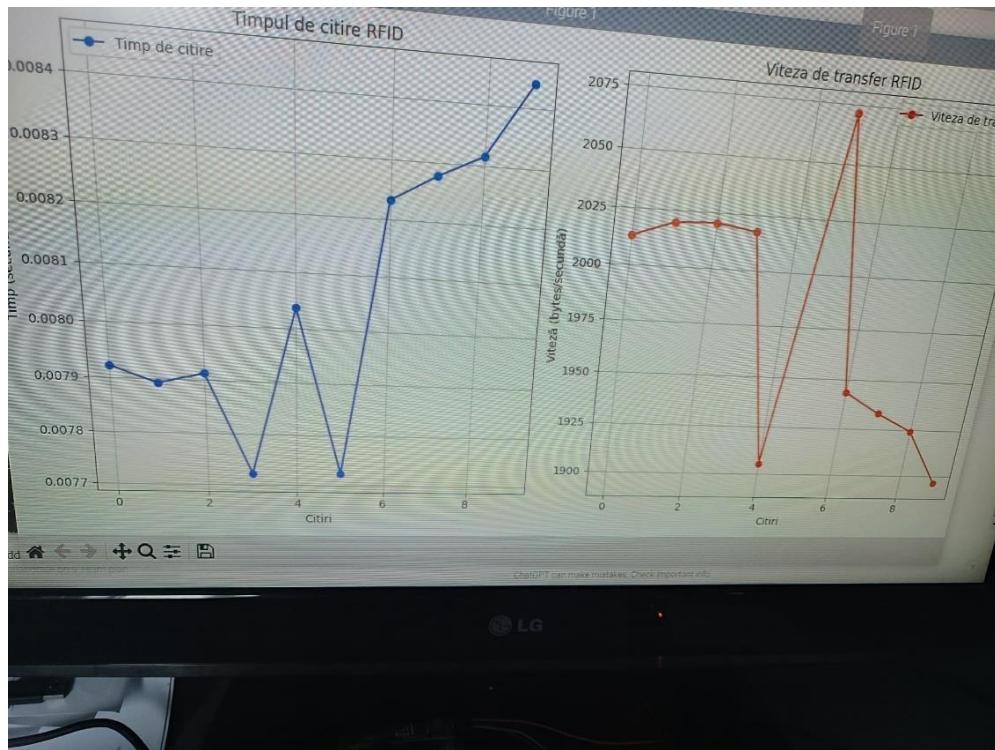


Fig.29 Grafice Timp de citire si viteza de transfer raportat la citiri

### **Timpul de Citire RFID**

Se observă o tendință de creștere a timpului de citire pe măsură ce se efectuează mai multe citiri, este de menționat că testele au fost făcute și cu accesări succesive de tag-uri, fără a se lăsa pauză între citiri.

Există fluctuații între citiri. Aceste variații pot fi cauzate de interferențe electromagnetice, variații ale semnalului RFID sau probleme temporare de procesare.

De observat faptul ca citirile sunt mai stabile la început.

Testele de mai sus au fost facute în primă fază doar cât modulul a fost conectat cu Raspberry-ul, ulterior au fost efectuate teste și cât timp am avut componentele conectate iar rezultatele au fost foarte asemănătoare.

### **Viteza de Transfer RFID**

După cum se observă în fig.28 se observă un spike în viteza de transfer la un anumit punct, urmat de o scădere. Acest comportament ar putea fi rezultatul unei perturbări temporare în semnalul RFID sau a unui spike în activitatea procesorului. Tagurile au fost la fel folosite, în mod succesiv. Tagul a fost ținut lângă modul.

Viteza de transfer este stabilă la început, similar cu timpul de citire, ceea ce sugerează o funcționare normală inițială. Instabilitatea apare ușor pe parcurs, iar ulterior vine din nou o perioadă de ușoara stabilizare, spike-urile se pot întâmpla și din motive externe cum ar fi modul în care tagul este apropiat sau ținut lângă modul, însă acestea nu afectează atât de tare funcționalitatea acestuia.

**Ecranizarea unor elemente sensibile** pentru a reduce interferențele electromagnetice. Acest lucru poate fi realizat prin utilizarea de cabluri ecranate și poziționarea strategică a componentelor pentru a minimiza interferențele.

A urmat adăugarea lcd-ului unde la fel am avut probleme legate de biblioteci. În cele din urmă am optat pentru RPLCD.

Biblioteca RPLCD.i2c este specializată în controlul afișajelor LCD care folosesc interfața I2C, cum este LCD-ul 1602 utilizat în proiectul tău. Acest tip de afișaj este ideal pentru afișarea mesajelor de confirmare și instrucțiuni pentru utilizatori, necesare în sistemul de prezență.

Proiectul include multiple componente hardware care necesită pini GPIO, cum ar fi senzorul ultrasonic HC-SR04 și servomotorul SG90. Utilizarea interfeței I2C pentru afișajul LCD permite economisirea pinilor GPIO, necesari pentru aceste alte componente.

Biblioteca RPLCD.i2c permite controlul contrastului și al iluminării de fundal, ceea ce îmbunătățește vizibilitatea mesajelor în diverse condiții de iluminare. Acest lucru este util în contextul unui eveniment unde condițiile de lumină pot varia.

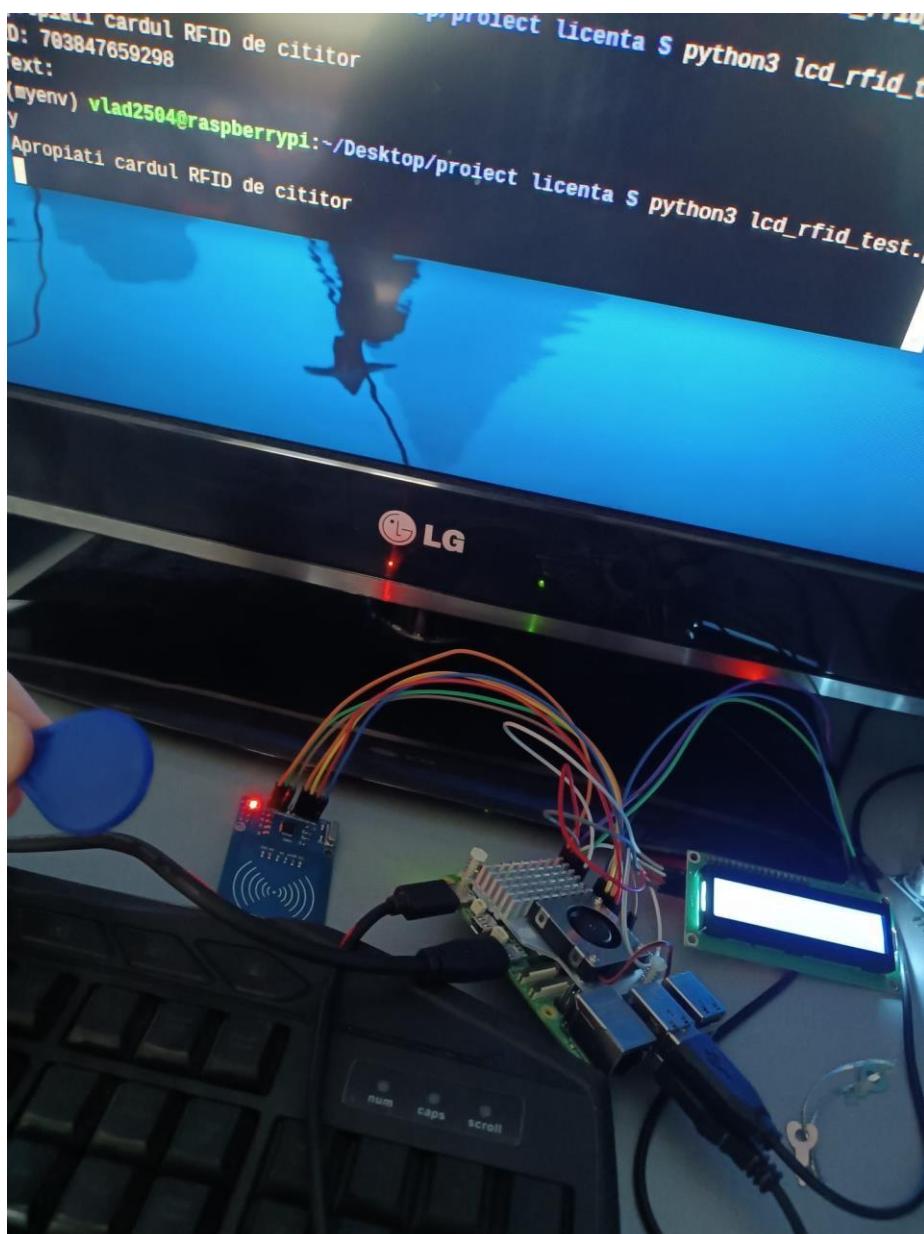


Fig.30 Configurare hardware si teste



Fig.31.1 Testare citire,lcd, senzor



fig.31.2 Testare citire,lcd,senzor

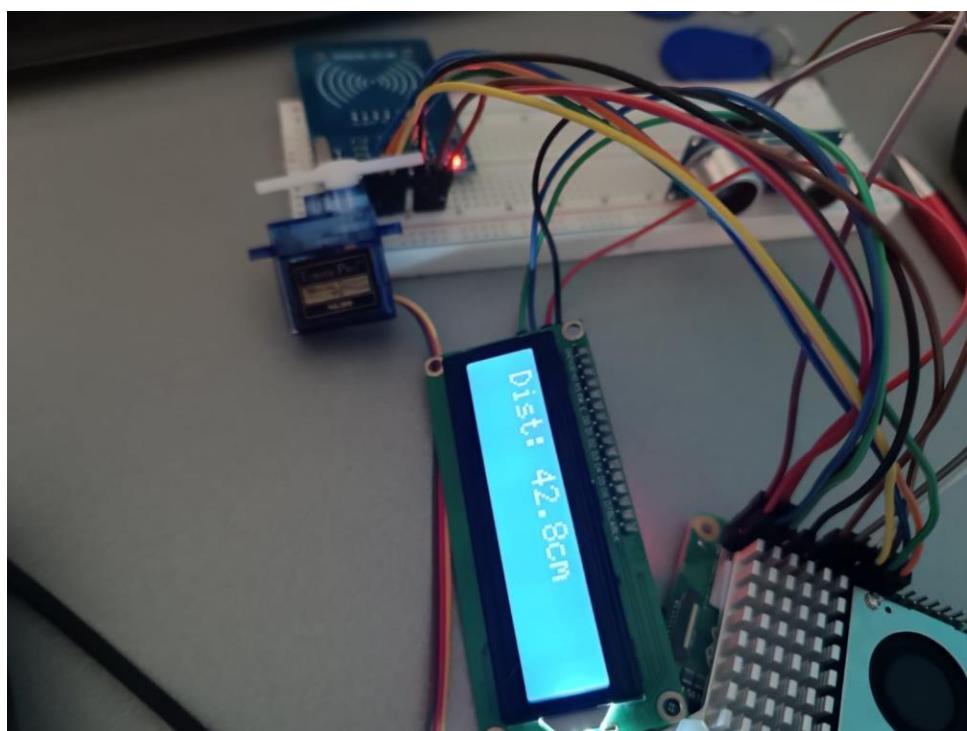


Fig.31.3 Proiectare si testare

Când a fost vorba de testele pentru senzor, acesta a oferit deviații medii de maxim 0.10 cm, aşadar o valoare destul de neglijabilă.



Fig.32 Schemă bloc funcționare după citirea RFID

Procesul începe cu sistemul pregătit să detecteze și să citească un tag RFID.

Sistemul citește tagul RFID prezentat de participant. Tagul conține un ID unic pentru identificarea participantului.

D-ul citit de pe tag este comparat cu înregistrările din baza de date.

Se va verifica dacă participantul este găsit în baza de date.

Dacă participantul este găsit, afișajul LCD arată un mesaj de bun venit. Senzorul ultrasonic măsoară distanța pentru a verifica poziționarea participantului.

Dacă distanța este optimă, servomotorul deschide bariera.

După acces, servomotorul închide bariera.

Procesul se încheie și sistemul revine la starea de aşteptare pentru următorul participant.

## **Portiuni din codul principal**

```
from flask import Flask, request, render_template
from gpiozero import DistanceSensor, Servo
from RPLCD.i2c import CharLCD
import mariadb
from pirc522 import RFID
import os
import time
import uuid # Import pentru generarea ID-urilor unice
```

Flask este un micro-framework web pentru Python. Este folosit pentru a crea aplicații web ușoare și flexibile.

Flask creează o aplicație Flask.

Permite accesul la datele din cererile HTTP (cum ar fi datele trimise prin formulare).

render\_template: Randează şabloane HTML folosind Jinja2 pentru a genera pagini web dinamice.

DistanceSensor: O clasă din gpiozero pentru a controla senzorii de distanță, cum ar fi HC-SR04.

Servo: O clasă din gpiozero pentru a controla servomotoarele.

os: Biblioteca standard Python pentru interacțiunea cu sistemul de operare. Este folosită aici pentru a rula comenzi ale sistemului de operare.

time: Biblioteca standard Python pentru manipularea timpului, inclusiv pentru întârzieri și măsurarea duratelor.

uuid: O bibliotecă standard Python pentru generarea identificatorilor unici universali (UUID). UUID-urile sunt folosite pentru a genera ID-uri unice pentru participanți.

Această secțiune de cod importă toate bibliotecile necesare pentru funcționarea sistemului. Flask este folosit pentru a crea aplicația web, gpiozero pentru a controla componente hardware (senzorul de distanță și servomotorul), RPLCD pentru a controla afișajul LCD, mariadb pentru interacțiunea cu baza de date, pirc522 pentru gestionarea cititorului RFID, iar bibliotecile standard os, time și uuid sunt folosite pentru diverse funcționalități de sistem, manipularea timpului și generarea ID-urilor unice.

```

rfid = RFID()
sensor = DistanceSensor(echo=24, trigger=23)
servo = Servo(18)
lcd = CharLCD(i2c_expander='PCF8574', address=0x27, port=1, cols=16, rows=2,
dotsize=8)

```

Aici am avut instanțierile. Adresa folosită pentru Lcd am aflat-o la configurarea lcd-ului.

```

db_connection = mariadb.connect(
    host="localhost",
    user="your_username",
    password="your_password",
    database="your_database"
)
db_cursor = db_connection.cursor()

```

Această secțiune de cod stabilește conexiunea la baza de date MariaDB și creează un cursor pentru executarea interogărilor SQL. Parametrii de conexiune specifică locația serverului de baze de date (în acest caz, local), precum și acreditările de autentificare și baza de date utilizată. Cursorul creat este utilizat ulterior pentru a executa operațiuni SQL, cum ar fi inserarea, actualizarea și selectarea datelor.

```

def read_tag():
    while True:
        (error, tag_type) = rfid.request()
        if not error:
            (error, uid) = rfid.anticoll()
            if not error:
                rfid.select_tag(uid)
                (error, data) = rfid.read(8)
                if not error:
                    return uid, data
    time.sleep(0.1) # Așteaptă 100ms înainte de a verifica din nou
    return None, None

```

Funcția `read_tag` din cod a fost actualizată pentru a utiliza o buclă de așteptare în loc de `irq`, eliminând astfel necesitatea de a utiliza întreruperi pentru detectarea tagurilor RFID. În loc de aceasta, funcția verifică periodic prezența unui tag RFID la intervale de 100ms. Procesul începe prin așteptarea detectării unui tag folosind metoda `rfid.request()`. Dacă un tag este detectat cu succes, se continuă cu o operațiune anti-coliziune prin metoda `rfid.anticoll()` pentru a obține ID-ul unic (UID) al tagului. Odată ce UID-ul este obținut, tagul este selectat folosind `rfid.select_tag(uid)`, permitând astfel operațiuni de citire sau scriere ulterioare.

Datele sunt apoi citite de la tag utilizând `rfid.read(8)`, care returnează datele stocate în blocul 8 al tagului RFID. Dacă toate operațiunile sunt reușite, funcția returnează UID-ul și datele citite. În cazul în care apare o eroare în oricare dintre pași, funcția se întoarce în bucla de așteptare, continuând să verifice periodic prezența unui tag până când citirea reușește. Această abordare simplifică gestionarea tagurilor RFID și reduce complexitatea necesară pentru implementarea întreruperilor.

```

def write_tag(tag_id):
    while True:
        (error, tag_type) = rfid.request()
        if not error:
            (error, uid) = rfid.anticoll()
            if not error:
                rfid.select_tag(uid)
                rfid.write(8, str(tag_id))
            return
        time.sleep(0.1) # Așteaptă 100ms înainte de a verifica din nou

```

Funcționează aproximativ ca funcția de citire dar aceasta are rol de a pune baza funcționalității pentru următoarea funcție care se ocupă cu algoritmul selecției a cee ace scriem pe tag.

```

def is_user_present():
    # Măsurarea distanței folosind senzorul ultrasonic
    echo_time = sensor.distance # Timpul de ecou în secunde
    distance = (echo_time * 34300) / 2 # Calculul distanței în cm
    print(f"Distanța măsurată: {distance} cm")
    return distance < 50 # Prag de 50 cm pentru detectarea prezenței

```

Funcția is\_user\_present măsoară distanța față de un obiect folosind senzorul ultrasonic HC-SR04. Timpul de ecou este obținut și convertit în distanță în centimetri. Dacă distanța măsurată este mai mică de 50 cm, funcția consideră că un utilizator este prezent și returnează True, altfel returnează False. Aceasta permite sistemului să detecteze dacă un utilizator se află într-o poziție optimă pentru a interacționa cu cititorul RFID și alte componente ale sistemului. Formula pentru distanță a devenit optională odată cu utilizarea bibliotecii GPIOzero.

```

@app.route('/register', methods=['POST'])
def register():
    nume = request.form['nume']
    prenume = request.form['prenume']
    cnp = request.form['cnp']
    numar_buletin = request.form['numar_buletin']

    # Generarea unui ID unic
    unique_id = str(uuid.uuid4())
    participant_data = (unique_id, nume, prenume, cnp, numar_buletin)

    # Inserarea datelor participantului în baza de date și obținerea ID-ului
    query = "INSERT INTO participants (id, nume, prenume, cnp, numar_buletin) VALUES (?, ?, ?, ?, ?)"
    db_cursor.execute(query, participant_data)
    db_connection.commit()

```

Funcția register gestionează înregistrarea unui participant nou. Aceasta extrage datele trimise prin formular, generează un ID unic pentru participant, inserează datele în baza de date și afișează o pagină. Acest proces asigură că fiecare participant are un ID unic și că datele acestuia sunt stocate corect în baza de date.

```
def write_tag_endpoint():
    participant_id = request.form['participant_id']
    manual_id = request.form.get('manual_id')

    if manual_id:
        write_tag(manual_id)
        return f"ID manual {manual_id} a fost scris pe tagul RFID."
    else:
        write_tag(participant_id)
        return f"ID automat {participant_id} a fost scris pe tagul RFID."
```

Funcția write\_tag\_endpoint gestionează scrierea unui ID pe un tag RFID. Preia ID-ul participantului din formular și, dacă un ID manual este furnizat, scrie acel ID pe tag. Dacă nu, scrie ID-ul generat automat. Returnează un mesaj de confirmare în funcție de tipul ID-ului scris.

```
def main():
    try:
        print("Sistemul este gata. Așteptarea citirii tagurilor RFID...")
        while True:
            # Monitorizarea temperaturii CPU
            cpu_temp = get_cpu_temp()
            print("Temperatura CPU: " + str(cpu_temp))
```

Funcția main începe prin afișarea unui mesaj în consolă pentru a indica faptul că sistemul este pregătit și așteaptă citirea tagurilor RFID. Apoi intră într-o buclă infinită care rulează continuu, monitorizând temperatura CPU-ului. Funcția get\_cpu\_temp este apelată pentru a obține temperatura curentă a CPU-ului, iar această valoare este afișată în consolă pentru monitorizare. Această secțiune asigură că temperatura CPU-ului este urmărită în timp real, ceea ce poate fi util pentru diagnosticare și pentru a preveni supraîncălzirea dispozitivului.

```
# Detectarea prezenței utilizatorului
if is_user_present():
    lcd.clear()
    lcd.write_string("Poziționează-te corect")
    time.sleep(2)
```

În bucla principală, sistemul verifică prezența unui utilizator folosind funcția is\_user\_present. Această funcție utilizează senzorul ultrasonic pentru a măsura distanța și a determina dacă un utilizator se află în apropiere. Dacă un utilizator este detectat, afișajul LCD este șters și se afișează mesajul "Poziționează-te corect", spunându-i utilizatorului să se poziționeze corect în fața senzorului. Sistemul așteaptă apoi 2 secunde pentru a permite

utilizatorului să se poziționeze corespunzător. Această secțiune asigură că utilizatorul este în poziția corectă pentru citirea tagului RFID.

```
# Citirea unui tag RFID
tag_id, tag_data = read_tag()
if tag_id:
    print(f"Tag detectat: ID={tag_id}, Data={tag_data}")

# Verificarea și afișarea datelor participantului
query = "SELECT * FROM participants WHERE id = ?"
db_cursor.execute(query, (tag_id,))
participant = db_cursor.fetchone()
if participant:
    lcd.clear()
    lcd.write_string(f"Bun venit, {participant['nume']}")
    {participant['prenume']}")
    servo.max() # Deschiderea barierei
    time.sleep(5) # Timp de așteptare pentru acces
    servo.min() # Închiderea barierei
else:
    lcd.clear()
    lcd.write_string("Tag necunoscut. Înregistrare necesară.")
else:
    lcd.clear()
    lcd.write_string("Eroare la citirea tagului.")

time.sleep(1)
except KeyboardInterrupt:
    print("Programul a fost oprit manual.")
```

Sistemul încearcă să citească un tag RFID folosind funcția `read_tag()`. Aceasta returnează ID-ul și datele tagului. Dacă un tag este citit cu succes (tag\_id nu este `None`), ID-ul și datele tagului sunt afișate în consolă pentru monitorizare.

Se execută o interogare SQL pentru a verifica dacă ID-ul citit există în baza de date a participanților. Dacă participantul este găsit (participant nu este `None`), afișajul LCD este șters și se afișează mesajul "Bun venit" urmat de numele și prenumele participantului. Servomotorul este acționat pentru a deschide bariera (`servo.max()`), iar sistemul așteaptă 5 secunde pentru a permite accesul, după care bariera este închisă (`servo.min()`).

Dacă participantul nu este găsit în baza de date, afișajul LCD este șters și se afișează mesajul "Tag necunoscut. Înregistrare necesară". În cazul în care citirea tagului eșuează, afișajul LCD este șters și se afișează mesajul "Eroare la citirea tagului".

După fiecare iterație a buclei, sistemul așteaptă 1 secundă (`time.sleep(1)`) înainte de a relua procesul, asigurând o funcționare stabilă și periodică a verificărilor.

```
finally:
    rfid.cleanup()
    sensor.close()
```

```

lcd.close(clear=True)
db_cursor.close()
db_connection.close()

if __name__ == "__main__":
    app.run(debug=True)
    main()

```

Blocul finally din funcția main asigură curățarea resurselor utilizate de sistem. Acesta curăță cititorul RFID, închide senzorul ultrasonic și afișajul LCD, și închide cursorul și conexiunea la baza de date. Dacă scriptul este rulat direct, aplicația Flask este pornită în modul debug și funcția main este apelată pentru a începe execuția buclei principale. Această structură asigură că toate resursele sunt eliberate corespunzător și că aplicația web rulează corect.

Pentru baza de date, am folosit Maria DB iar principalele linii de cod din terminal pentru a începe au fost:

```

CREATE USER 'admin'@'localhost' IDENTIFIED BY 'parola_admin';
GRANT ALL PRIVILEGES ON *.* TO 'admin'@'localhost' WITH GRANT OPTION;
mysql -u admin -p
CREATE DATABASE attendance_system;
USE attendance_system;
CREATE TABLE participants (
    id VARCHAR(36) PRIMARY KEY,
    nume VARCHAR(100),
    prenume VARCHAR(100),
    cnp VARCHAR(13),
    numar_buletin VARCHAR(20)
);

```

Practic eu în aceste linii de cod am creat un admin cu permișii complete, am creat baza de date attendance\_system și tabela participants pentru stocarea informațiilor despre participanți.

## 8. Posibile implementări ulterioare

Posibile implementări ulterioare pentru acest sistem includ mai multe opțiuni pentru îmbunătățirea funcționalității și securității. Prima opțiune este adăugarea unui modul de autentificare biometrică, cum ar fi un scanner de amprente sau recunoașterea facială, care ar spori semnificativ securitatea sistemului și ar elimina riscul folosirii neautorizate a tagurilor RFID.

O altă opțiune este integrarea sistemului cu platforme de notificare pentru a trimite notificări prin email sau SMS atunci când un participant se înregistrează sau accesează evenimentul. Aceasta ar îmbunătăți comunicarea și managementul evenimentelor, oferind organizatorilor informații în timp real despre participare.

Dezvoltarea unui modul de analiză a datelor și generare de rapoarte ar permite monitorizarea și analizarea statisticilor de participare și a altor metrii relevante. Aceasta ar ajuta organizatorii să îmbunătățească experiența participanților printr-o înțelegere mai detaliată a comportamentului acestora la evenimente.

Un sistem de plăti integrat ar simplifica procesul de înregistrare și plată pentru participanți, folosind platforme precum PayPal sau Stripe. Aceasta ar oferi o soluție completă de management al evenimentelor, făcând întregul proces mai convenabil pentru utilizatori.

Dezvoltarea unei aplicații mobile complementare ar permite participanților să se înregistreze, să acceseze informații despre eveniment și să primească notificări direct pe dispozitivele lor mobile. Aceasta ar îmbunătăți considerabil experiența utilizatorilor și ar facilita comunicarea între organizatori și participanți.

Extinderea funcționalității sistemului pentru a se integra cu alte sisteme de control al accesului, cum ar fi porți turnichet sau bariere automate, ar crește eficiența și securitatea la evenimente. Aceasta ar permite un control mai bun al accesului în diverse zone ale locației evenimentului.

Suportul pentru multiple evenimente simultane ar fi, de asemenea, o îmbunătățire semnificativă. Adaptarea sistemului pentru a gestiona mai multe evenimente în același timp, cu baze de date și interfețe dedicate pentru fiecare eveniment, ar sprijini organizarea simultană a mai multor evenimente, facilitând managementul centralizat.

În final, dezvoltarea unei interfețe de administrare web mai avansate, cu funcționalități suplimentare pentru organizatori, cum ar fi managementul utilizatorilor și vizualizarea în timp real a datelor despre eveniment, ar ușura considerabil managementul evenimentelor. Aceasta ar oferi organizatorilor instrumente avansate pentru monitorizare și administrare, făcând procesul de organizare mai eficient și mai transparent.

Aceste implementări ulterioare pot aduce îmbunătățiri semnificative sistemului actual, sporindu-i funcționalitatea, securitatea și ușurința de utilizare, atât pentru organizatori, cât și pentru participanți.

## 9. Concluzii

Proiectul demonstrează viabilitatea utilizării tehnologiei RFID și a platformei Raspberry Pi pentru dezvoltarea unui sistem de înregistrare și acces automatizat. Sistemul integrează cu succes mai multe componente hardware și software, oferind o soluție eficientă și sigură pentru gestionarea participării la evenimente.

Testele efectuate pe diverse componente ale sistemului au confirmat stabilitatea și fiabilitatea acestuia. Monitorizarea temperaturii CPU și performanțele senzorului ultrasonic

au fost esențiale pentru a asigura funcționarea optimă a sistemului. Graficele generate în timpul testelor au evidențiat performanțele constante ale sistemului și au permis identificarea și optimizarea potențialelor instabilități.

Rezultatele testelor au arătat că senzorul ultrasonic HC-SR04 a oferit măsurători precise ale distanței, ceea ce a permis detectarea corectă a prezenței utilizatorilor. Afişajul LCD 1602 IIC/I2C a furnizat informații clare și concise, îmbunătățind interacțiunea utilizatorilor cu sistemul. Modulul RFID RC522 a demonstrat capacitatea de a citi și scrie taguri în mod eficient, asigurând unicitatea și securitatea datelor stocate.

Implementarea unui algoritm de generare a ID-urilor unice a fost crucială pentru a preveni conflictele și pentru a menține integritatea bazei de date. Conexiunea cu MariaDB a permis stocarea și gestionarea eficientă a datelor participanților, iar utilizarea Flask pentru dezvoltarea interfeței web a facilitat o experiență de utilizator intuitivă.

Pe plan software, utilizarea librăriilor specifice pentru fiecare componentă hardware, cum ar fi gpizero și RPLCD.i2c, a simplificat semnificativ procesul de dezvoltare și integrare. Alegerea limbajului Python a oferit flexibilitate și ușurință în scrierea și întreținerea codului, iar Flask a oferit un cadru robust și ușor de utilizat pentru dezvoltarea aplicațiilor web.

Din punct de vedere al utilizatorului final, sistemul oferă o experiență intuitivă și eficientă, permitând înregistrarea rapidă și accesul facil la evenimente. Mesajele afișate pe ecranul LCD și controlul automatizat al barierelor asigură o interacțiune simplă și clară pentru utilizatori.

În concluzie, proiectul demonstrează viabilitatea utilizării tehnologiei RFID și a platformei Raspberry Pi pentru dezvoltarea unui sistem de înregistrare și acces automatizat. Implementările viitoare pot extinde funcționalitățile actuale, aducând îmbunătățiri în securitate, analiză de date și experiență utilizatorului, făcând acest sistem și mai robust și versatil. Proiectul servește ca o bază solidă pentru dezvoltări ulterioare și poate fi adaptat pentru diverse aplicații și scenarii de utilizare. Testele și graficele obținute în timpul implementării au evidențiat stabilitatea și performanțele sistemului, oferind încredere în aplicabilitatea și extensibilitatea acestuia în medii reale.

## 10 BIBLIOGRAFIE

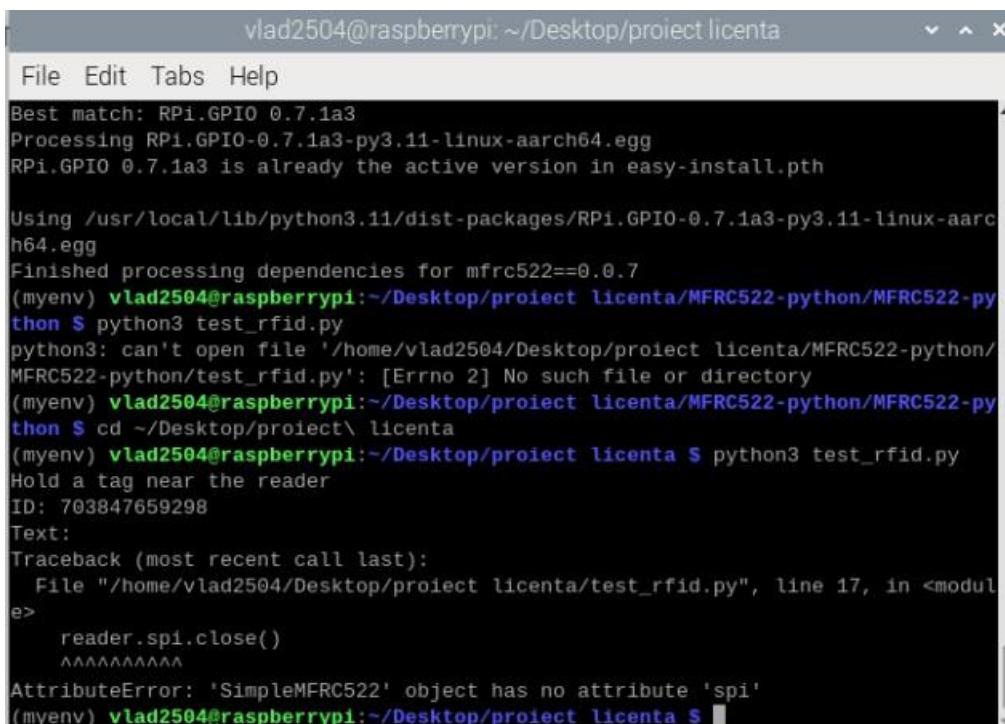
---

1. TutorialsWeb. "Operation of RFID Systems." Accessed at: <https://www.tutorialsweb.com/rfid/operation-of-rfid-systems.htm>.
2. CircuitDigest. "Interfacing RFID Reader Module with Arduino." Accessed at: <https://circuitdigest.com/microcontroller-projects/interfacing-rfid-reader-module-with-arduino>.
3. Ortiz, S. "How secure is RFID?" in Computer, vol. 39, no. 7, pp. 17-19, July 2006. doi: 10.1109/MC.2006.232.
4. Lahiri, S. *RFID Sourcebook*. IBM Press, 2005.
5. Ayodele, F., Singh, H., & AbdAllah, E. G. "Securing RFID-Based Attendance Management Systems: An Implementation of the AES Block Cipher Algorithm." 2023 IEEE 13th International Conference on RFID Technology and Applications (RFID-TA), Aveiro, Portugal, 2023, pp. 99-102. doi: 10.1109.
6. Ogruțan, Petre. *Microcontroller: Privire generală*, 2020.
7. Hagl, Andreas. *RFID Security: Techniques, Protocols, and System-On-Chip Design*. 1st ed., Springer US, 2009.
8. Zebra Technologies. "FX7500 Fixed RFID Reader Specification Sheet." Accessed at: <https://www.zebra.com/us/en/products/spec-sheets/rfid/rfid-readers/fx7500.html>.
9. ResearchGate. "A demonstration of the moving average filter and low-pass filter in action." Accessed at: [https://www.researchgate.net/figure/A-demonstration-of-the-moving-average-filter-and-low-pass-filter-in-action-They fig5\\_336091851](https://www.researchgate.net/figure/A-demonstration-of-the-moving-average-filter-and-low-pass-filter-in-action-They fig5_336091851).
10. Microchip Technology Inc. "AN1301 - Data Synchronization: Manchester Encoding and Its Uses." Accessed at: <https://ww1.microchip.com/downloads/en/AppNotes/01301A.pdf>.
11. Texas Instruments. "SLAA539–June 2012 - Manchester Code Basics." Accessed at: <https://www.ti.com/lit/an/slaa539/slaa539.pdf>.
12. International Electrotechnical Commission (IEC). "ISO/IEC 14443-1:2018 Identification cards - Contactless integrated circuit cards - Proximity cards - Part 1: Physical characteristics." Accessed at: <https://www.iso.org/standard/73568.html>.
13. NXP Semiconductors. "MIFARE Classic EV1." Accessed at: <https://www.nxp.com/products/rfid-nfc/mifare-ics/mifare-classic/mifare-classic-ev1-1k-and-4k>.
14. Zebra Technologies. "FX7500 Fixed RFID Reader Support." Accessed at: <https://www.zebra.com/us/en/support-downloads/rfid/fixed-readers/fx7500.html>.
15. Analog Devices. "Understanding and Minimizing RF Interference in RFID Systems." Accessed at: <https://www.analog.com/en/analog-dialogue/articles/understanding-minimizing-rf-interference-rfid.html>.
16. Texas Instruments. "AN-1060 Application Note: Antenna Basics for UHF RFID Readers." Accessed at: <https://www.ti.com/lit/an/sloa132/sloa132.pdf>.
17. IEEE. "IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 15.1a: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs)." Accessed at: <https://ieeexplore.ieee.org/document/4478902>.
18. Peter, D. (2007). "RFID and contactless technology." VDE Verlag GmbH.
19. ISO. "ISO 18000-6:2013 Information technology - Radio frequency identification for item management - Part 6: Parameters for air interface communications at 860 MHz to 960 MHz." Accessed at: <https://www.iso.org/standard/46145.html>.
20. Karygiannis, T., & Eydt, B. (2007). "Guidelines for Securing Radio Frequency Identification (RFID) Systems." National Institute of Standards and Technology (NIST). Accessed at: <https://csrc.nist.gov/publications/detail/sp/800-98/archive/2007-05-15>.

21. ResearchGate. "Securing RFID Systems using Advanced Encryption Standard (AES)." Accessed at:  
[https://www.researchgate.net/publication/328053892\\_Securing\\_RFID\\_Systems\\_using\\_Advanced\\_Encryption\\_Standard\\_AES](https://www.researchgate.net/publication/328053892_Securing_RFID_Systems_using_Advanced_Encryption_Standard_AES).
22. Wiley Online Library. "RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification." Accessed at:  
<https://onlinelibrary.wiley.com/doi/book/10.1002/0470854647>.
23. Elsevier. "A Survey of Security and Privacy Issues in the Internet of Things: Solutions and Future Directions." Accessed at:  
<https://www.sciencedirect.com/science/article/pii/S1570870518304242>.
24. SpringerLink. "RFID Security: Techniques, Protocols, and System-On-Chip Design." Accessed at: <https://link.springer.com/book/10.1007/978-1-4419-8216-2>.
25. IEEE Xplore. "An Analysis of RFID Security Techniques." Accessed at:  
<https://ieeexplore.ieee.org/document/8443097>.

## Exemple de erori întâmpinate

- Problemele legate de alimentare, cum ar fi alimentarea servomotorului, care are nevoie de măcar 700mA și 5V pentru funcționare în regim maxim, în cele din urmă am încercat integrarea unui acumulator de 2A și 5V pentru a avea o alimentare constantă.
- Erorile legate de GPIO care au fost prezente în mare parte din implementarea proiectului, fie eroarea legată de SOC, fie eroarea legată de ocuparea pinilor GPIO, aplicația crezând că alte procese de pe fundal se desfășoară. Soluțiile fie au fost schimbarea bibliotecii fie o aplicație de a elibera procesele de pe GPIO.



The screenshot shows a terminal window titled "vlad2504@raspberrypi: ~/Desktop/proiect licenta". The terminal output is as follows:

```
File Edit Tabs Help
Best match: RPi.GPIO 0.7.1a3
Processing RPi.GPIO-0.7.1a3-py3.11-linux-aarch64.egg
RPi.GPIO 0.7.1a3 is already the active version in easy-install.pth

Using /usr/local/lib/python3.11/dist-packages/RPi.GPIO-0.7.1a3-py3.11-linux-aarch64.egg
Finished processing dependencies for mfrc522==0.0.7
(myenv) vlad2504@raspberrypi:~/Desktop/proiect licenta/MFRC522-python/MFRC522-python$ python3 test_rfid.py
python3: can't open file '/home/vlad2504/Desktop/proiect licenta/MFRC522-python/MFRC522-python/test_rfid.py': [Errno 2] No such file or directory
(myenv) vlad2504@raspberrypi:~/Desktop/proiect licenta/MFRC522-python/MFRC522-python$ cd ~/Desktop/proiect\ licenta
(myenv) vlad2504@raspberrypi:~/Desktop/proiect licenta$ python3 test_rfid.py
Hold a tag near the reader
ID: 703847659298
Text:
Traceback (most recent call last):
  File "/home/vlad2504/Desktop/proiect licenta/test_rfid.py", line 17, in <module>
    reader.spi.close()
    ^^^^^^^^^^
AttributeError: 'SimpleMFRC522' object has no attribute 'spi'
(myenv) vlad2504@raspberrypi:~/Desktop/proiect licenta$
```

Fig E1 problemă legată de biblioteca RC522

O problemă care s-a rezolvat prin găsirea unei biblioteci pentru raspberry care pentru Rc522 care să cuprindă GPIOzero.

```

setup.py - /home/vlad2504/Desktop/proiect licenta/MFRC522-python3 - Geany
File Edit Search View Document Project Build Tools Help
vlad2504@raspberrypi: ~/Desktop/project licenta/MFRC522-python3
Symbols File Edit Tabs Help
Variables
  • PLATFORM
  • author [28]
  • author_email [28]
  • dependency_links [25]
  • description [25]
  • install_requires [25]
  • name [25]
  • packages [25]
  • url [30]
  • version [26]
Imports
  • distro [5]
  • os [2]
  • setup [4]
  • subprocess [2]
Status
  17:23:28: File /home/vlad2504/Desktop/project licenta/MFRC522-python3/setup.py saved.

Traceback (most recent call last):
  File "/home/vlad2504/Desktop/proiect licenta/MFRC522-python3/setup.py", line 2
2, in <module>
    PLATFORM = platform.linux_distribution()[0].lower()
               ^^^^^^^^^^^^^^^^^^^^^^^^^^
AttributeError: module 'platform' has no attribute 'linux_distribution'
(myenv) vlad2504@raspberrypi:~/Desktop/project licenta/MFRC522-python3 $ pip ins
tall distro
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting distro
  Downloading https://www.piwheels.org/simple/distro/distro-1.9.0-py3-none-any.w
hl (20 kB)
Installing collected packages: distro
Successfully installed distro-1.9.0
(myenv) vlad2504@raspberrypi:~/Desktop/project licenta/MFRC522-python3 $ sudo py
thon3 setup.py install
/home/vlad2504/Desktop/proiect licenta/MFRC522-python3/setup.py:10: DeprecationW
arning: distro.linux_distribution() is deprecated. It should only be used as a c
ompatibility shim with Python's platform.linux_distribution(). Please use distro
.id(), distro.version() and distro.name() instead.
    PLATFORM = distro.linux_distribution()[0].lower()
17:23:28: File /home/vlad2504/Desktop/project licenta/MFRC522-python3/setup.py saved.

```

Fig.E2 Una dintre problemele apărute la instalări

De cele mai multe ori aceste probleme apăreau fie din cauza că nu foloseam mediul virtual, fie din cauza unei comenzi folosite care necesita anumite permisiuni.

```

File Edit Tabs Help
finally:
    reader.spi.close() # Închidem SPI după citire
    lgpio.gpiochip_close(0^C)
(myenv) vlad2504@raspberrypi:~/Desktop/proiect licenta $ python test_rfid.py
old a tag near the reader
D: 703847659298
ext:
Traceback (most recent call last):
  File "/home/vlad2504/Desktop/proiect licenta/test_rfid.py", line 12, in <mod
>
    reader.spi.close() # Închidem SPI după citire
               ^^^^^^^^
AttributeError: 'SimpleMFRC522' object has no attribute 'spi'
(myenv) vlad2504@raspberrypi:~/Desktop/proiect licenta $ python test_rfid.py
old a tag near the reader
D: 703847659298
ext:
Traceback (most recent call last):
  File "/home/vlad2504/Desktop/proiect licenta/test_rfid.py", line 16, in <mod
>
    reader.cleanup()
               ^^^^^^
AttributeError: 'SimpleMFRC522' object has no attribute 'cleanup'
(myenv) vlad2504@raspberrypi:~/Desktop/proiect licenta $ 

```

Fig.E3 Eroare apărută din cauza funcției de cleanup

Eroarea asta a mai apărut când am încercat să folosesc diverse funcții incompatibile cu RC522.

Câteodată s-a mai întâmplat să nu fie activat SPI sau pur și simplu librăria sa nu fie compatibilă și aparea că modulul RC nu conține atributul acesta.

```
[root@mfrc522] (3.6)
(myenv) vlad2504@raspberrypi:~/Desktop/proiect_licenta $ ls /dev/spidev*
/dev/spidev0.0  /dev/spidev0.1  /dev/spidev10.0
(myenv) vlad2504@raspberrypi:~/Desktop/proiect_licenta $
```

Aceasta era una din modalitățile prin care am testat dacă SPI este activat.  
Alte erori au mai fost bazate pe comunicația dintre componente, alimentarea acestora,etc.

---

UNIVERSITATEA TRANSILVANIA DIN BRAŞOV  
FACULTATEA \_\_\_\_\_

CERERE DE ÎNSCRIERE LA EXAMENUL DE \_\_\_\_\_

**I. Date personale ale candidatului/ candidatei comunicate în scopul prelucrării necesare pentru organizarea examenului de finalizare studii**

1. Date privind identitatea persoanei

Numele de naștere: \_\_\_\_\_ Numele (dacă este cazul): \_\_\_\_\_  
Prenumele: \_\_\_\_\_ CNP \_\_\_\_\_

2. Sexul:  Feminin  Masculin

3. Data și locul nașterii:

Ziua / luna / anul \_\_\_\_\_ / \_\_\_\_\_ / \_\_\_\_\_

Locul (localitate, județ, țara) \_\_\_\_\_ / \_\_\_\_\_ / \_\_\_\_\_

4. Prenumele părinților:

Tata: \_\_\_\_\_ Mama: \_\_\_\_\_

5. Domiciliul stabil: Localitatea \_\_\_\_\_, jud. \_\_\_\_\_ Cod poștal \_\_\_\_\_  
str. \_\_\_\_\_ nr. \_\_\_, bloc \_\_\_, sc. \_\_\_, et. \_\_\_, ap. \_\_\_,

Telefon \_\_\_\_\_, mail \_\_\_\_\_

**II. Date privind școlarizarea**

6. Sunt absolvent(ă) promoția: \_\_\_\_\_ / \_\_\_\_\_ (anul înmatriculării / anul absolvirii)

7. Mențiuni privind școlarizarea: \_\_\_\_\_

8. Programul de studii \_\_\_\_\_

9. Durata studiilor \_\_\_\_\_

10. Forma de învățământ absolvită:  IF  IFR  ID

Fără taxă  Cu taxă

11. Solicit înscrierea la examenul de \_\_\_\_\_, sesiunea \_\_\_\_\_ anul \_\_\_\_\_

12. Lucrarea/ Proiectul de \_\_\_\_\_ pe care o susțin are următorul titlu: \_\_\_\_\_

13. Conducător științific: \_\_\_\_\_

14. Susțin examenul de \_\_\_\_\_ (pentru prima oară, a doua oară - dupăcaz) \_\_\_\_\_.

15. Menționez că sunt de acord cu afișarea rezultatelor examenului conform art.15 alin.9/art.18 alin.9 din OMENCS nr.6125/2016 modificat prin OMEN nr.5643/2017.

SEMNĂTURA,  
Secretar facultate

(numele și prenumele, semnătura)

VERIFICAT,

F05-PS 7.6-01/ed.2,rev.2

## Dosar de înscriere la examenul de diplomă/disertație

Pentru înscrierea la examenul de licență/ diplomă/ disertație, absolvenții trebuie să depună la secretariat următoarele acte:

1. Cerere de înscriere la examen
2. Declarație pe proprie răspundere privind prelucrarea datelor cu caracter personal în cadrul procedurii de organizare a examenului de licență/diplomă/disertație
3. Certificat de naștere, în copie legalizată sau în copie simplă care a fost certificată „Conform cu originalul” de către persoana autorizată din secretariatul facultății, în baza prezentării actului în original;
4. Certificat de căsătorie (dacă este cazul), în copie legalizată sau în copie simplă care a fost certificată „Conform cu originalul” de către persoana autorizată din secretariatul facultății, în baza prezentării actului în original;
5. Ordin al rectorului de schimbare a numelui absolventului (dacă este cazul);
6. Diplomă de bacalaureat sau echivalentă cu aceasta, în copie legalizată sau în copie simplă, certificată „Conform cu originalul” de către persoana autorizată din secretariatul facultății, în baza prezentării actului în original – pentru examenul de licență/diplomă;
7. Diploma de licență sau diplomă de inginer și anexa la diplomă, în copie legalizată sau în copie simplă care a fost certificată „Conform cu originalul” de către persoana autorizată din secretariatul facultății, în baza prezentării actului în original – pentru examenul de disertație;
8. Certificat de competență lingvistică (numai pentru examenul de licență sau de diplomă), eliberat de instituția organizatoare sau de o altă instituție specializată, națională sau internațională, recunoscută de instituția organizatoare.

Pentru absolvenții proprii, competențele lingvistice certificate prin notele din registrul matricol la o limbă străină de largă comunicare internațională sunt recunoscute de Departamentul de Lingvistică teoretică și aplicată, fără a mai fi necesar un certificat de competență lingvistică atașat la dosar.

Pentru absolvenții proprii care susțin examenele de licență/diplomă la alte instituții de învățământ superior, precum și pentru absolvenți ai altor instituții de învățământ superior care susțin examenele de licență/diplomă la UTBv, existența în dosar a certificatului de competență lingvistică este obligatorie.

9. 2 fotografii color, recente, dimensiunea ¾ cm, pe hârtie fotografică;
10. Carte de identitate sau pașaport, în copie;
11. Copie a Scrisorii de acceptare la studii / Ordinului MEN sau Atestatului de echivalare (dacă este cazul);
12. Chitanța de plată a taxei de examen (dacă este cazul);
13. Declarație pe proprie răspundere privind originalitatea lucrării de licență/proiectului de diplomă/disertației;

Absolvenții proveniți de la alte instituții de învățământ superior vor depune documentele prevăzute la pt.1-13, la care se adaugă:

14. Suplimentul la diplomă, eliberat de instituția de învățământ de stat sau particular superior absolvită, din care să rezulte, pentru fiecare semestru și an de studii, disciplinele promovate, numărul de ore prevăzut pentru fiecare curs, aplicații, lucrări practice – separat, forma de verificare (examen, colocviu, proiect, verificare), creditele și notele obținute. și o copie a Suplimentului la diplomă, certificată „conform cu originalul” de către facultatea care o eliberează;

15. Adeverință eliberată de instituția de învățământ de stat sau particular superior absolvită, din care să rezulte calitatea de absolvent, întocmită în conformitate cu Ordinul

.....

Documentele se depun la secretariatul facultății într-un dosar plic de carton, pe care se înscriu:

-Numele și prenumele absolventului;

- Programul de studii
- Facultatea
- Sesiunea
- Promoția

*Notă: Certificarea conformității cu originalul a copiilor după actele de identitate/de stare civilă și a actelor de studii se face de către angajații desemnați din cadrul facultății, în baza prezentării documentului în original.*

**DECLARAȚIE PRIVIND ORIGINALITATEA  
LUCRĂRII DE LICENȚĂ / PROIECTULUI DE DIPLOMĂ /  
DISERTAȚIEI**

UNIVERSITATEA TRANSILVANIA DIN BRAȘOV  
FACULTATEA INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR  
PROGRAMUL DE STUDII ELECTRONICĂ APLICATĂ

NUMELE ȘI PRENUMELE: Solomon Vlad-George

PROMOȚIA: 2020-2024

SESIUNEA: IULIE 2024

TEMA LUCRĂRII / PROIECTULUI / DISERTAȚIEI: Sistem automat de prezență cu RFID și Raspberry PI

CONDUCĂTOR ȘTIINȚIFIC: Prof. Dr. Ing. Ogruțan Petre-Lucian

Declar pe propria răspundere că lucrarea de față este rezultatul muncii proprii, pe baza cercetărilor proprii și pe baza informațiilor obținute din surse care au fost citate și indicate conform normelor etice, în textul lucrării/proiectului, în note și în bibliografie.

Declar că nu s-a folosit în mod tacit sau ilegal munca altora și că nici o parte din teză/proiect nu încalcă drepturile de proprietate intelectuală ale altcuiu, persoană fizică sau juridică.

Declar că lucrarea/ proiectul nu a mai fost prezentat(ă) sub această formă vreunei instituții de învățământ superior în vederea obținerii unui grad sau titlu științific ori didactic.

În cazul constatării ulterioare a unor declarații false, voi suporta rigorile legii.

Data: 25.06.2024

Absolvent

