

UNIVERSITATEA TEHNICĂ „Gheorghe Asachi” din IAȘI  
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DOMENIUL: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI  
SPECIALIZAREA: CALCULATOARE

## **SISTEM DE CONTROL AL ACCESULUI SECURIZAT CU AUTENTIFICARE MULTIPLĂ**

Coordonator științific  
Ş. l. dr. Mihai Timiș

Absolvent  
Roberta-Georgiana Popa

Iași, 2023

- **Scopul temei alese:**

Scopul temei alese este de a dezvolta un sistem de control al accesului pentru încuietorile electrice cu ajutorul placuței Raspberry Pi Pico W, care să asigure securitatea într-un mod eficient și practic într-o locație restricționată. Sistemul va permite utilizatorilor să acceseze zona restricționată prin introducerea unui cod pe un keypad sau prin utilizarea unui card RFID. De asemenea, sistemul va afișa starea ușii pe un ecran LCD și va fi capabil să controleze sistemul de blocare al încuietorii.

- **Referințe la teme/subiecte similare:**

Această temă se încadrează în domeniul securității și controlului accesului. Subiectele similare includ dezvoltarea de sisteme de securitate și control acces (într-o locuință/firmă/instituție), utilizând tehnologia RFID și sistemele de control al accesului bazate pe coduri PIN, cu ajutorul platformelor hardware precum Arduino, Raspberry Pi, ESP8266 etc. Cu toate acestea, această lucrare se va concentra pe dezvoltarea unei soluții personalizate utilizând placuța Raspberry Pi Pico W.

- **Resursele hardware/software utilizate:**

- Raspberry Pi Pico W
- Modul RFID (Reader/Writer) RC522
- Sistem de blocare a ușii (sistem electromecanic de blocare)
- Keypad
- Ecran LCD
- Breadboard
- Jumper wires
- MicroPython
- Editor de cod (Thonny sau Visual Studio Code)

- **Algoritmi sau metode alese:**

- Algoritmul de citire a datelor de la modulul RFID și verificarea autenticității utilizatorului
- Algoritmul de citire a datelor de pe keypad și verificarea codului PIN introdus de utilizator
- Algoritmul de control al sistemului de blocare a ușii și a semnalizării stării acestaia
- Algoritmul de interfațare între modulul RFID, keypad, ecran LCD și Raspberry Pi Pico W pentru a permite comunicația între cele patru dispozitive

- **Rezultate așteptate:**

Se așteaptă ca această lucrare de licență să demonstreze modul în care Raspberry Pi Pico W poate fi utilizat pentru dezvoltarea și testarea cu succes a unui sistem de control al accesului eficient și practic, care asigură securitatea încuietorilor electrice, folosind un sistem de autentificare prin card RFID și cod PIN. Sistemul de blocare dezvoltat va oferi un nivel ridicat de securitate, o autentificare multiplă, ceea ce va permite utilizatorilor să controleze sistemul cu ușurință.

- **Proiectarea hardware/software a lucrării:**

**Hardware-ul sistemului este format din:**

1. Placa Raspberry Pi Pico W: este placa principală a sistemului și va fi utilizată pentru a controla toate celelalte componente. Are un procesor ARM Cortex-M0+ și este compatibilă cu majoritatea senzorilor și modulelor hardware. Plăcuța Raspberry Pi Pico W va fi conectată la internet prin Wi-Fi și va putea să se conecteze la serverul web.
2. Modulul RFID: Acesta va fi utilizat pentru autentificarea utilizatorului și va permite accesul doar utilizatorilor autorizați. Este conectat la plăcuță prin intermediul unei interfețe SPI.

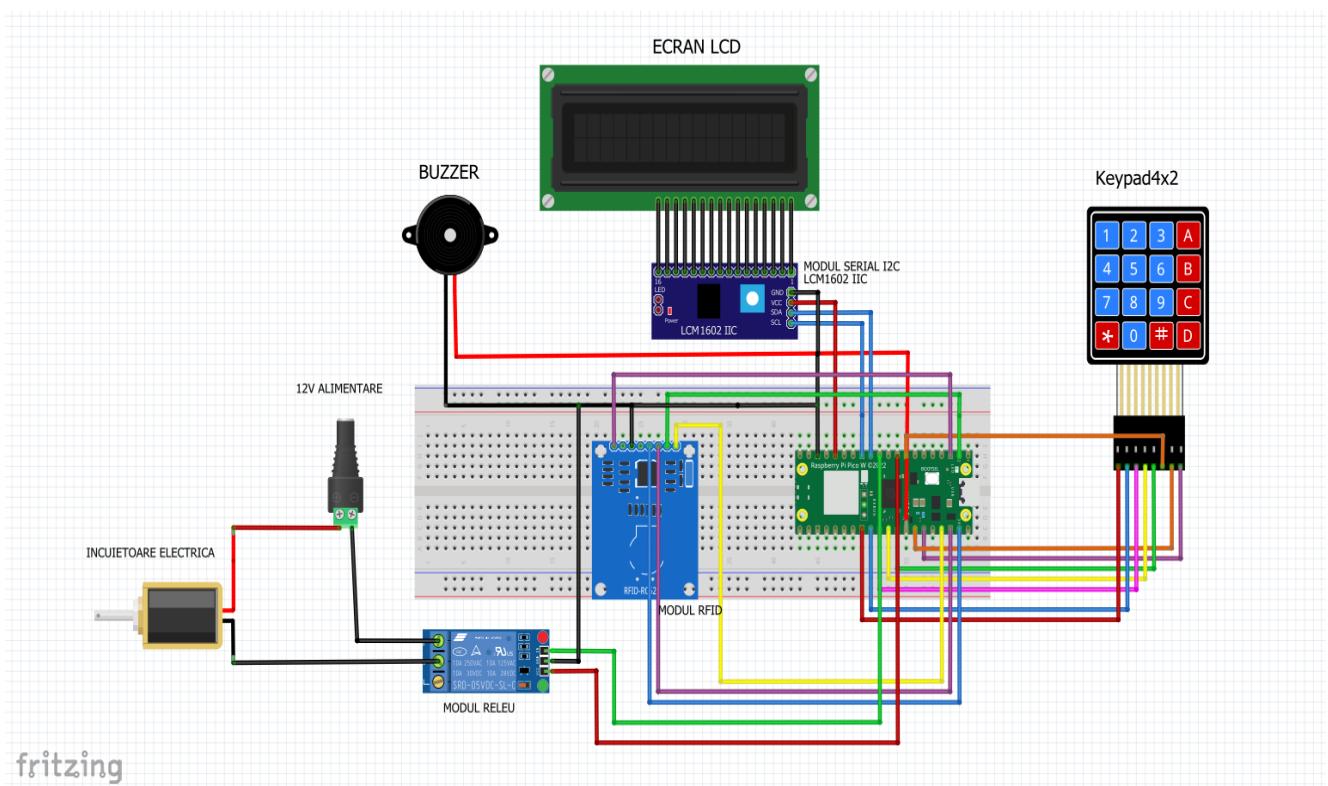
3. Keypad: Utilizatorii neautorizați vor putea accesa sistemul prin intermediul unui keypad care va fi conectat la plăcuță. Acesta va fi utilizat pentru a introduce un cod PIN, în cazul în care utilizatorii nu au un tag RFID valid.

4. Ecran LCD: Un ecran LCD va fi utilizat pentru afișarea mesajelor și informațiilor referitoare la sistem, precum: la deblocarea încuietorii se va afișa "pin incorrect/corect" sau "deschis/inchis". Este conectat la plăcuță cu ajutorul unui modul interfață I2C.

5. Încuietoare electrică: O încuietoare electrică va fi utilizată pentru a bloca/debloca accesul. Aceasta va fi conectată la Raspberry Pi Pico W cu ajutorul unui modul releu și va fi controlată prin intermediul acestuia.

6. Alimentare: Sistemul va fi alimentat printr-un cablu micro-USB conectat la o sursă de alimentare.

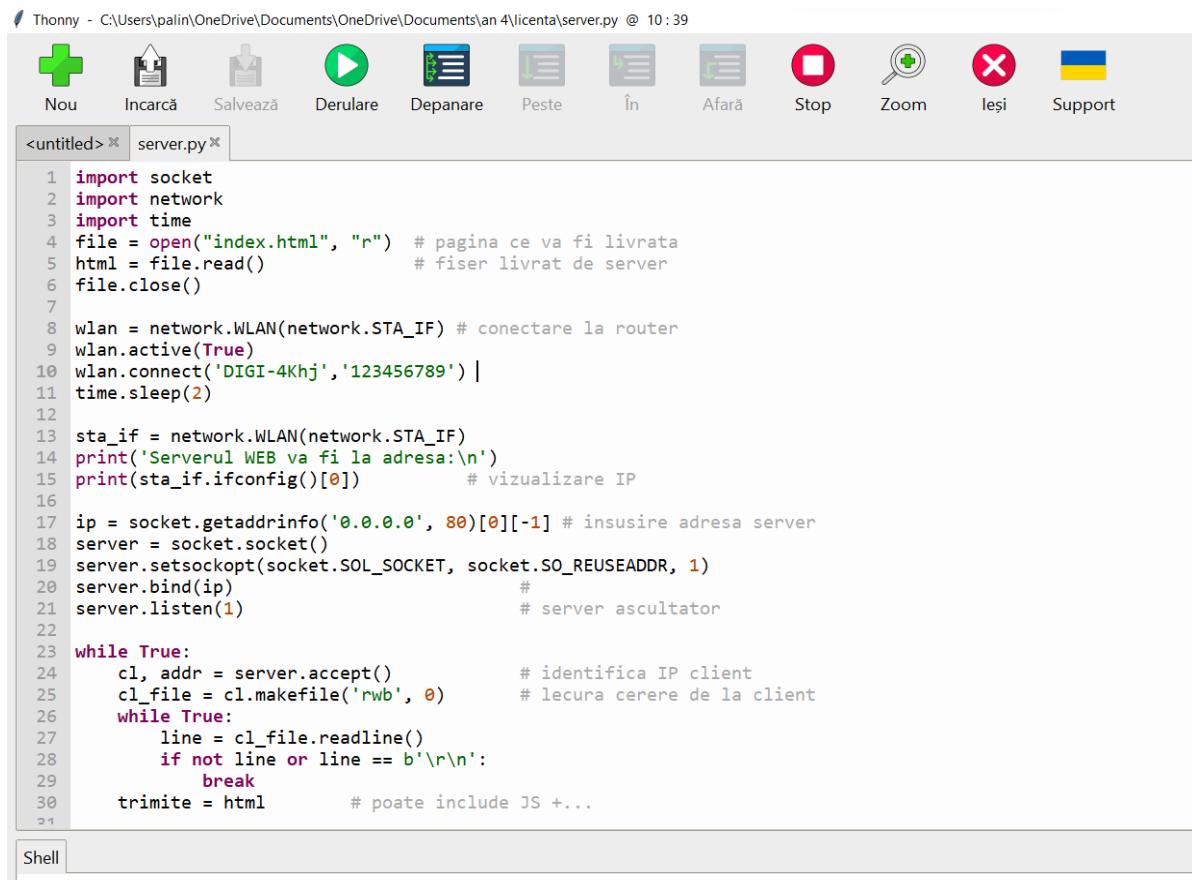
Partea software va fi scrisă în Python și include diferite module pentru interacțiunea cu componentele hardware, cum ar fi modulele MFRC522 pentru modulul RFID și RPLCD pentru ecranul LCD. În plus, sistemul folosește biblioteci Python precum Flask pentru dezvoltarea interfeței web și RPiGPIO pentru controlul componentelor hardware.



## • Rezultate intermediare obținute:

Un prim pas în realizarea practică a proiectului după partea hardware și asamblarea componentelor, a fost realizarea conexiunii Raspberry Pi Pico W la o rețea locală de tip server web clasic (Apache). Baza conexiunii la rețea o reprezintă circuitul **Infineon CYW43439** care permite implementarea pentru Pico W a modulelor de tip network și socket a căror componente permit realizarea aplicațiilor de tip web server.

Folosind programul Thonny, am rulat și lansat în execuție pe Pico W și am reținut adresa alocată de către router. Am configurat o funcție WLAN obiect cu adresa locală de internet (SSID “DIGI-4Khj”, Parolă “123456789”).



The screenshot shows the Thonny IDE interface with the following details:

- Toolbar:** Includes icons for New, Load, Save, Run, Debug, Find, Stop, Zoom, Exit, and Help.
- File List:** Shows a single file named "server.py".
- Code Editor:** Displays the Python code for a web server:

```
1 import socket
2 import network
3 import time
4 file = open("index.html", "r") # pagina ce va fi livrata
5 html = file.read()           # fisier livrat de server
6 file.close()
7
8 wlan = network.WLAN(network.STA_IF) # conectare la router
9 wlan.active(True)
10 wlan.connect('DIGI-4Khj', '123456789') |
11 time.sleep(2)
12
13 sta_if = network.WLAN(network.STA_IF)
14 print('Serverul WEB va fi la adresa:\n')
15 print(sta_if.ifconfig()[0])      # vizualizare IP
16
17 ip = socket.getaddrinfo('0.0.0.0', 80)[0][-1] # insusire adresa server
18 server = socket.socket()
19 server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
20 server.bind(ip)                  #
21 server.listen(1)                # server ascultator
22
23 while True:
24     cl, addr = server.accept()    # identifica IP client
25     cl_file = cl.makefile('rwb', 0) # lecure cerere de la client
26     while True:
27         line = cl_file.readline()
28         if not line or line == b'\r\n':
29             break
30     trimite = html      # poate include JS +...
```
- Shell:** A small window at the bottom left labeled "Shell".

- **Dificultăți/provocări întâmpinate:**

Una din principalele provocări întâmpinate a fost integrarea componentelor hardware și software pentru a asigura o funcționare corectă și stabilă a proiectului. Am mai întâmpinat probleme legate de conectarea la server și folosirea corectă a programului Thonny.