**ROMÂNIA**

**MINISTERUL APĂRĂRII NAȚIONALE**

**ACADEMIA TEHNICĂ MILITARĂ „FERDINAND I”**

**Facultatea de Sisteme Informatice și Securitate Cibernetică**

**Departamentul de Calculatoare și Securitate Cibernetică**



***Utilizare senzor ANALOG de lumina SI BUZZER DIGITAL***

***Platforma de dezvoltare frdm-kl25z***

Std. Sg. Maj. Clinci Daniel

Std. Sg. Maj. Lipan Maria

Std. Sg. Maj. Revnic Oana

**București**

2022

**Cuprins**

[1. Conținutul documentului 3](#_Toc122514023)

[2. Scopul documentului 3](#_Toc122514024)

[3. Modelul arhitectural 3](#_Toc122514025)

[4. Modelul datelor 4](#_Toc122514026)

[5. Setup 5](#_Toc122514027)

[6. Modelul interfeței cu utilizatorul 7](#_Toc122514028)

[7. Probleme întâmpinate 8](#_Toc122514029)

[8. Bibliografie 9](#_Toc122514030)

# **Conținutul documentului**

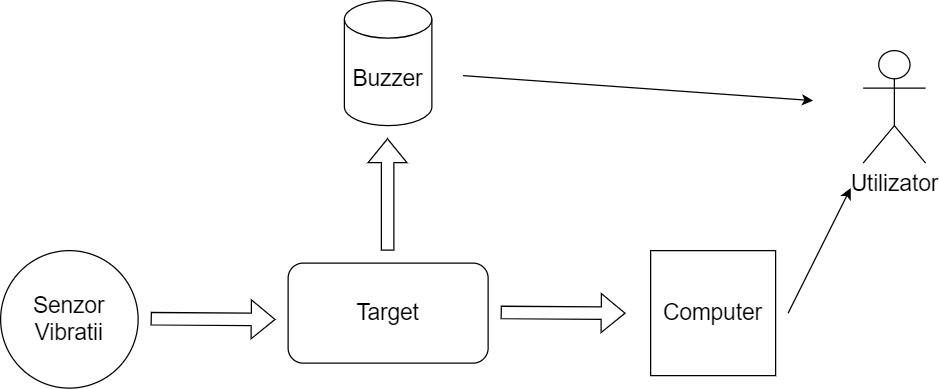
Documentul este alcătuit din opt capitole după cum urmează:

1. Primul capitol, cel prezent, reprezintă o scurtă trecere în revistă asupra elementelor principale ce constituie acest document;
2. Al doilea capitol evidențiază rolul pentru care acest document a fost constituit;
3. Al treilea capitol descrie componentele folosite în alcătuirea schemei arhitecturale pentru proiectul în cauză;
4. Al patrulea capitol ilustrează principalele structuri de date folosite;
5. Capitolul cinci pune în evidență aspectul cu care utilizatorul interacționează în cadrul aplicației;
6. Capitolul sase ilustrează un proiect auxiliar în limbajul de programare Python, care generează graficele necesare interfeței pentru utilizator;
7. Capitolul șapte descrie problemele întâmpinate ;
8. Capitolul opt conține referințele utilizate pentru realizarea proiectului;

# **Scopul documentului**

Acest document are rolul de a descrie precis şi complet soluţia proiectată pentru tema de proiect la materia Sisteme Specializate cu Microprocesoare. Documentul serveşte drept ghid unic de construire a soluţiei pentru echipa de dezvoltare a proiectului.

# **Modelul arhitectural**

****

**Descrierea componentelor**

* **Target** – reprezintă plăcuța FRDM-KL25Z pe care rulează aplicația. Prin intermediul interfețelor sale de intrare și ieșire captează date de la senzori, se ocupă de analiza sau prelucrare lor și le transmite mai departe către utilizator prin intermediul unui computer sau a altor senzori;
* **Senzor de lumina** – senzor analog ce are ca scop captarea luminii din mediul înconjurător și transmiterea ei către Target sub formă de tensiune electrică cuprinsă între 0 și 3.3 Volți;
* **Buzzer** – senzor digital care emite un zgomot atunci când este activat. Acesta primește de la Target un semnal digital pentru a schimba modul de funcționare din pornit în oprit sau invers. Scopul acestui senzor este de a notifica sonor pentru o perioadă anume de timp Utilizatorul de un anumit eveniment;

* **Computer** – prin intermediul lui, Utilizatorul are la dispoziție o interfață grafică de tip grafic de date pentru a vizualiza într-un mod facil datele transmise de Target;
* **LED** – integrat pe Target, își va schimba culoarea la apăsarea tastelor de la 0 la 7, de către Utilizator.

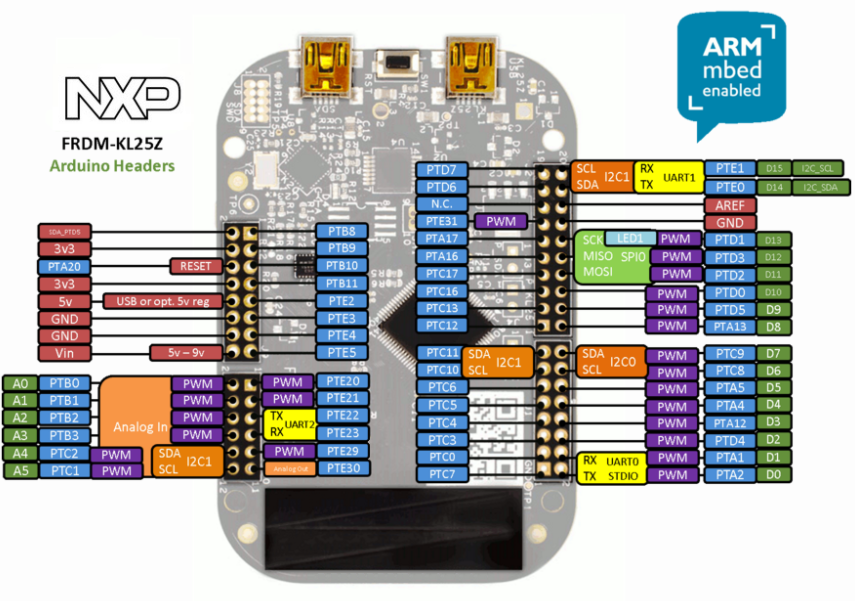
# **Modelul datelor**

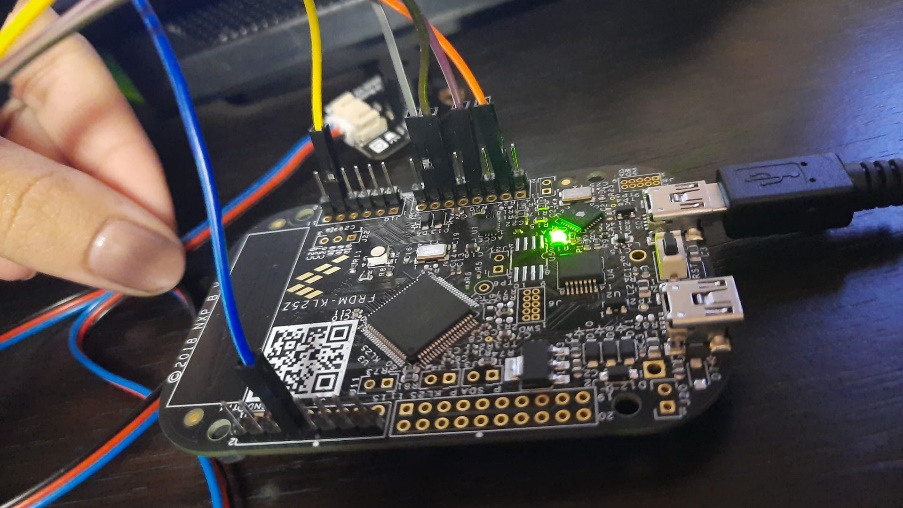
* **ADC**– Componenta ADC din cadrul aplicției are ca scop captarea tensiunii transmise de senzorul de vibrații de pe un port specificat și convertirea acesteia din format analog în format digital. Sursa acestei componente cuprinde o funcție pentru inițializare, o funcție pentru calibrare și o funcție destinată special conversiei.
* **GPIO**– Componenta GPIO realizează conexiunea cu Buzzer-ul. Aceasta are o funcție pentru inițializarea portului , două funcții ce au ca scop oprirea respectiv pornirea buzzer-ului, o funcție pentru inițializarea LED-ului și schimbarea culorii acestuia.

* **UART**– Pentru ca target-ul să transmită sau să primească date către și de la computer este disponibilă componenta UART. Aceasta inițializează comunicarea la o anumită viteză și realizează trimiterea sau primirea de date prin două funcții specifice.
* **PIT**– Asigură aplicției o componentă care se ocupă de evidența timpului. În urma funcției de inițializare, componenta PIT realizează o întrerupere la fiecare 0.1 secunde. Respectiva întrerupere este gestionată prin intermediul unei funcții care incrementează o variabilă.
* **Main**– Reprezintă sufletul aplicției deoarece adună la un loc componentele descrise anterior. În primă fază sunt inițializate componentele, iar apoi, într-o buclă infinită de tip „while” sunt convertite date de la senzorul de vibrații prin intermediul ADC la fiecare 0.3 secunde sunt convertite în format ASCII și sunt transmise prin UART către computer. Dacă datele citite de la senzorul de vibrații măsoară o tensiune mai mare de 2.0 volți buzzer-ul este pornit pentru a emite sunet timp de o secundă.

# **Setup**

1. **Target FRDM-KL25Z**





Firele au fost conectate astfel pe plăcuță:

1. Albastru: PTA12 - semnal digital buzzer
2. Galben: PTC2 - semnal analog senzor vibrații
3. Gri: GND buzzer
4. Verde: GND senzor vibrații
5. Mov: 3.3V buzzer
6. Portocaliu: 3.3V senzor vibrații
7. **Senzori**
   1. Senzorul analogic de Lumină



* 1. Senzorul digital Buzzer



1. **Pași pentru analiza configurației**
2. Conectarea senzorilor la Target;
3. Încărcarea codului pe Target din proiectul realizat in Keil uVision5;
4. Deschiderea unei instanțe PuTTY cu următoarele specificații:
   * Serial line: COM5
   * Speed: 115200
   * Connection type: Serial;
5. Se observă in consola PuTTY valorile primite de la senzorul analogic de lumină;
6. Se apasă tastele de la 0 la 7 pentru a observa modul de funcționare al LED-ului integrat, și anume: pentru fiecare buton de la 1 la 7 apăsat, culoarea LED-ului se schimbă, iar pentru 0 se stinge;
7. Se închide instanța de PuTTY;
8. Se deschide proiectul in PyCharm ce conține cele 2 surse scrise in Python: main.py si final.py;
9. Se rulează întâi main.py pentru a salva datele înregistrate de Target. Pentru obținerea unui rezultat complet, se ține o sursă de lumină puternică timp de 4 secunde deasupra senzorului (sa fie înregistrate valori de peste 2.0 V), apoi se îndepărtează pentru a obține valori medii (valori sub 2.0 V si peste ~0.4 V), apoi se va acoperi senzorul pentru a obține valori mici;
10. Se închide graficul dinamic generat de main.py și se va rula final.py pentru a vizualiza un graficul cu cele 3 praguri;

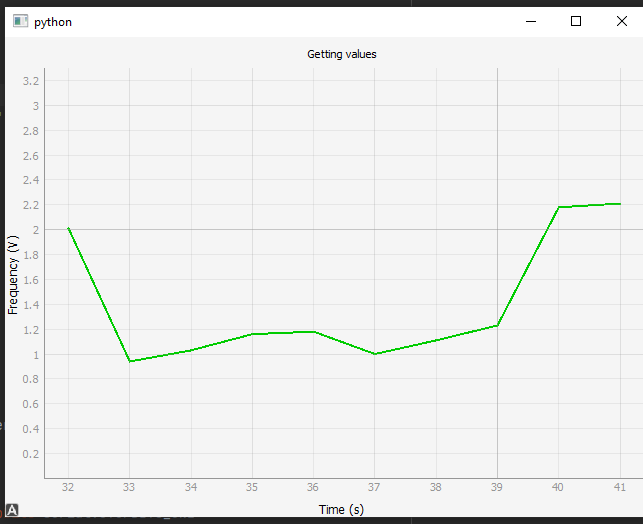
# **Modelul interfeței cu utilizatorul**

Folosindu-ne de PyQt5 si matplotlib, au fost implementate 2 surse de Python, astfel:

* **Main.py**

Se folosește de PyQt5, având ca scop afișarea într-un grafic dinamic valorile primite de la Target și salvarea acestora în fișierul „values.txt”.

Valorile primite sunt convertite in format ASCII, iar mai apoi in float pentru a le putea reprezenta grafic.

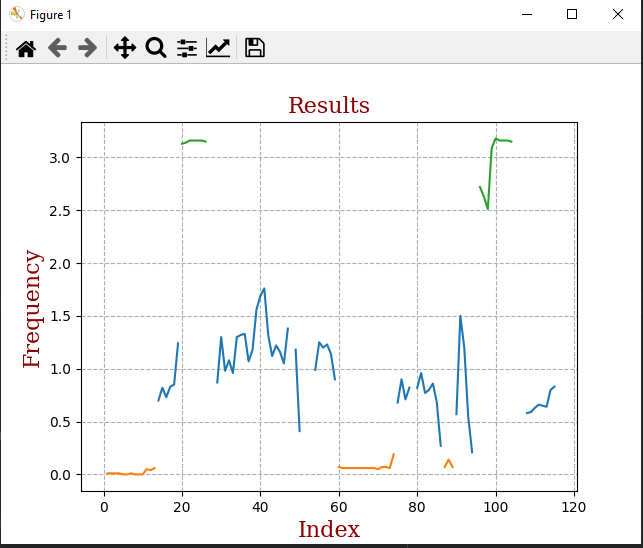


* **Final.py**

Se folosește de matplotlib, având ca scop afișarea într-un grafic static valorile generate de main.py in fișier.

Valorile sunt împărțite in 3 praguri, fiecare având o culoare unica: valori mici (portocaliu), medii (albastru) si mari (verde). Trecerea de la un prag la altul este calculata pe baza valorii medii, obținută cu ajutorul valorilor minim si maxim din fișier.

Fiecare frecventă înregistrată este indexată și afișată pe grafic, colorată corespunzător.



# **Probleme întâmpinate**

* În timpul dezvoltării proiectului senzorul de vibrații a fost supus unor forțe mult prea mari pentru rezistența lui fizică care au dus la ruperea unui fir. Astfel am fost nevoiți să îl înlocuim cu un alt senzor. Am ales să îl înlocuim cu un senzor de lumină de tip tot analog, deoarece astfel nu am fost nevoiți să modificăm funcționalitatea proiectului, senzorii comportându-se relativ asemănător, singura diferență fiind ce acțiune din mediul înconjurător captează;
* Am fost nevoiți să transmitem valorile înregistrate de Target cu o frecvență mai scăzută (interval de 0.3 secunde) pentru a se putea actualiza graficul generat de main.py fără a se bloca sau a afișa eronat.

# **Bibliografie**

* Laboratoare
* PDF-urile FRDM-KL25Z\_\*
* https://www.pythonguis.com/tutorials/plotting-pyqtgraph/