**ROMÂNIA**

**MINISTERUL APĂRĂRII NAŢIONALE**

**ACADEMIA TEHNICĂ MILITARĂ „FERDINAND I”**

**FACULTATEA DE SISTEME INFORMATICE ȘI SECURITATE CIBERNETICĂ**

**Specializarea: Calculatoare și sisteme informatice pentru apărare și securitate națională**



***Proiect microprocesoare – Senzor de vibratie piezo A***

**Stud.Plt. NEAMU Florin**

**Stud.Sg.Maj. OLTEAN Madalina**

**Bucuresti**

**2022**

Cuprins

[Scopul proiectului 3](#_Toc92876263)

[Componente 3](#_Toc92876264)

[Pasi in rularea programului 10](#_Toc92876265)

[Probleme intampinate 11](#_Toc92876266)

[Referințe 12](#_Toc92876267)

# Scopul proiectului

Scopul acestui proiect este de a prelua datele de la un sensor de vibratie analogic, si-anume senzorul de vibratie piezo A si de a afisa datele preluate de la acesta intr-un graphic folosind placuta NXP KL25Z. (Platforma)

In contexte practice (Draw.io)pentru utilizarea acestui sensor se incadreaza:detectarea batailor la usa si initierea unei sonerii in urma vibratiei usii saudeclansarea unei sonerii atunci cand masina de spalat a terminat de spalat, dat fiind ca acesta vibreaza in timpul programului de spalare.[5]Se pot gasi utilitati multiple pentru a face acest sensor folositor.

# Componente

* UART (Reference Manual)

Am folosit comunicatia UART pentru a transmite date de la ADC catre PC, pentru a le afisa ulterior intr-un grafic.Baudrate-ul folosit pentru transmisie este 115200.

Registrii configurati in cadrul modulului UART sunt:

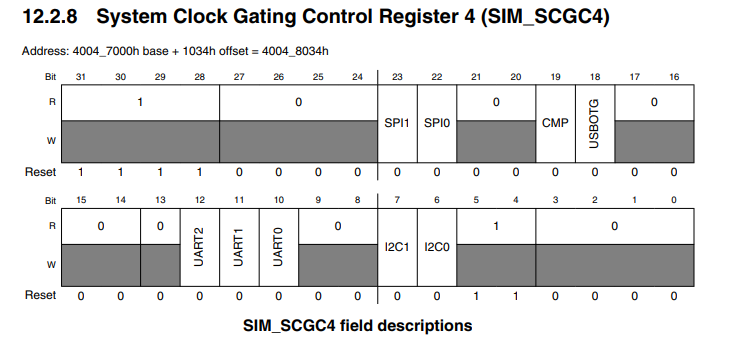
* SIM\_SCGC4 – pentru a activa canalul UART0 (am setat bitul 10)->in Fig. 0.1 se afla descrierea registrului pentru a evidential ca bitul 10 corespunde canalului UART0

Fig. 0.1 Descrierea registrului SIM\_SCGC

* SIM\_SCGC5 – pentru a activa ceasul pe portul A, pe care functioneaza transimisia si receptia. (am setat bitul 9)->Fig. 0.2 evidentiaza faptul ca bitul 9 corespunde portului A

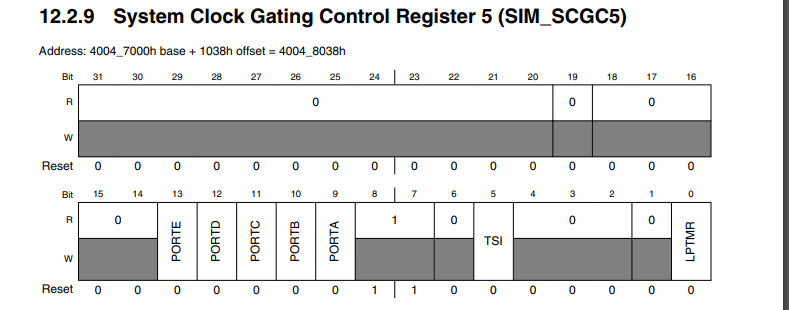


Fig. 0.2 Descrierea registrului SIM\_SCGC

* SIM\_SOPT2 – folosim acest registru pentru a seta sursa de ceas pentru UART0, pentru care am setat bitii 26-27 cu valoare 01.-> Fig. 0.3 evidentiaza ca bitul 26 si 27 corespund pentru UART0

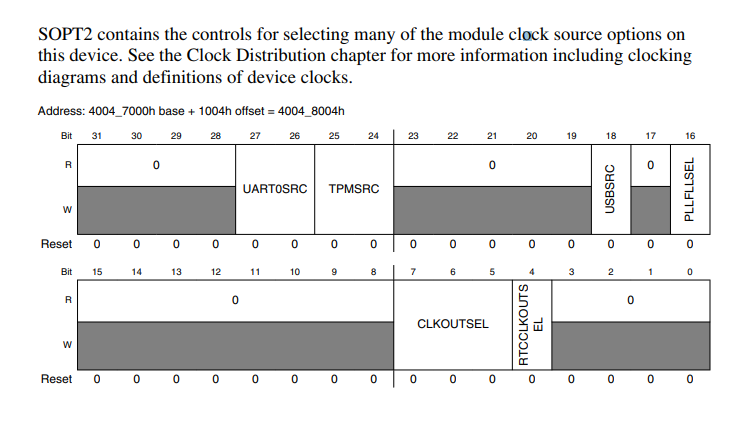


Fig.0.3 Descrierea registrului SOPT2

* PORTA\_PCR1 si PORTA\_PCR2 –setam bitii de multiplexare la 010 (bitii 10 si 8) deoarece acesti pini sunt utilizati pentru UART0 alternativa 2.->Fig.0.4 evidentiaza faptul ca 8 si 10 sunt bitii de multiplexare

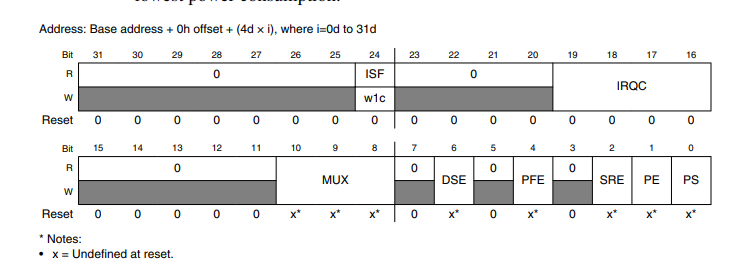


Fig. 0.4 Descrierea PORTA\_PCR1, PORTA\_PC

* UART0\_BDL si UART0\_BDH - pentru setarea baudrate-ului. Baudrate-ul nostrum este de 115200.->Fig. 0.5 evidentiaza continutul registrului UART0\_BDH, iar Fig. 0.6 evidentiaza continutul registrului UART0\_BDL

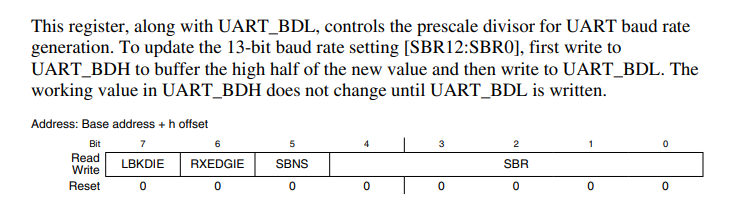


Fig.0.5 Descriere registru UART0\_BDH

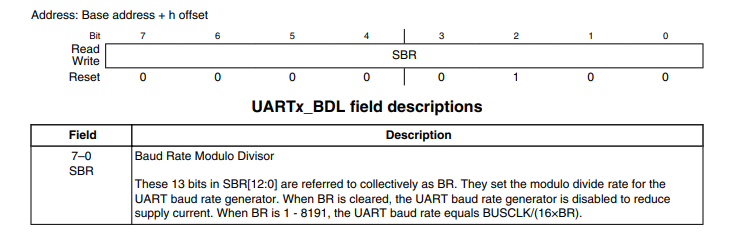


Fig.0.6 Descriere registru UART0\_BDL

* UART0\_C1 – pentru a controla numarul de biti pe care se face receptia ( Bitul 4 / 9-Bit or 8-Bit Mode Select )->Fig.0.7 evidentiaza continutul registrului UART0\_C1

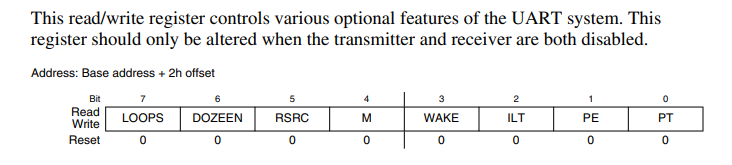


Fig.0.7 Descriere registru UART0\_C1

* UART0\_C2 – pentru a controla intreruperile pe UART0. Dezactivam intreruperile de emisie si le activam pe cele de receptie. ( bitul de Enable Transmitter este 3, iarcel de Enable Receiver este 2)->Fig. 0.8 evidentiaza continutul registrului UART0\_C2

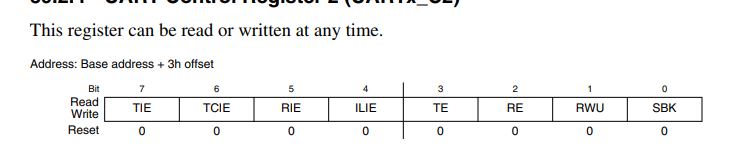


Fig.0.8 Descriere registru UART0\_C2

* UART\_S1 – pentru a verificadaca s-a primit un octet (bitul 5 )->Fig. 0.9 evidentiaza continutul registrului UART\_S1

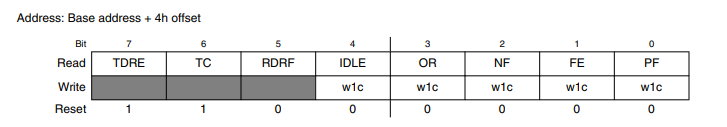


Fig.0.9 Descriere registru UART0\_S1

* UART\_D – UART data register->Fig. 0.10 evidentiaza continutul registrului UART\_D

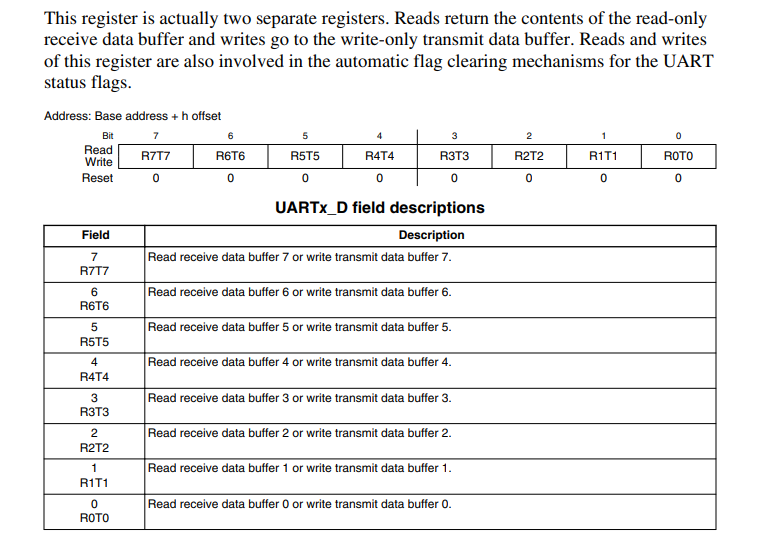


Fig.0.10 Descriere registru UART\_D

* ADC ( Analogic Digital Converter)

Pentru acest proiect am folosit pinul PTB0 care este corespunzator pentru ADC0\_SE8. Pe acest pin se face conversie pe canalul 8.Registrii folositi in cadrul modulului ADC sunt:

* SIM\_SCGC6 –pentru a activa ADC0 (setam bitul 27)->Fig. 0.11 evidentiaza continutul registrului SIM\_SCGC6 si corespondenta dintre bitul 27 si ADC0

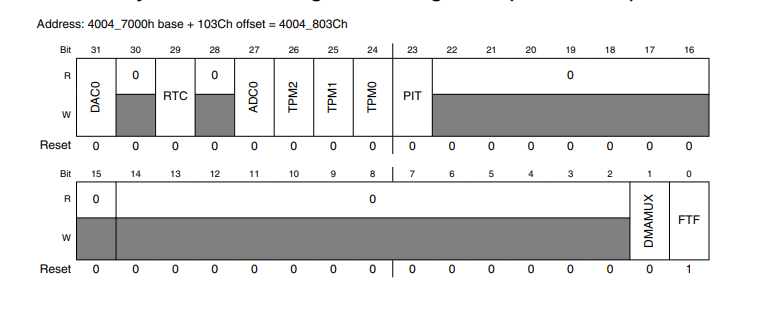


Fig. 0.11 Descriere registru SIM\_SCGC6

* ADC0\_SC1A – pentru status si pentru activarea intreruperilor ( bitul 6 Interrupt Enable )->Fig. 0.12 evidentiaza continutul registrului ADC0\_SC1A si necesitatea setarii bitului 6 pentru Interrupt Enable

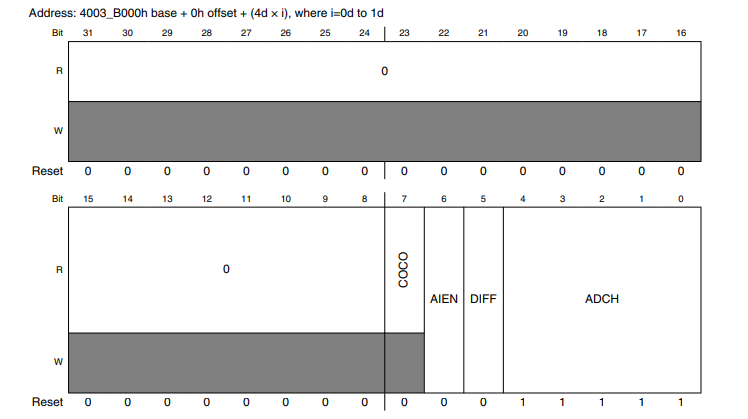


Fig. 0.12 Descriere registr ADC0\_SC1A

* ADC0\_CFG1 pentru a seta rezolutia convertorului->Fig. 0.13 evidentiaza continutul registrului ASC0\_CFG1

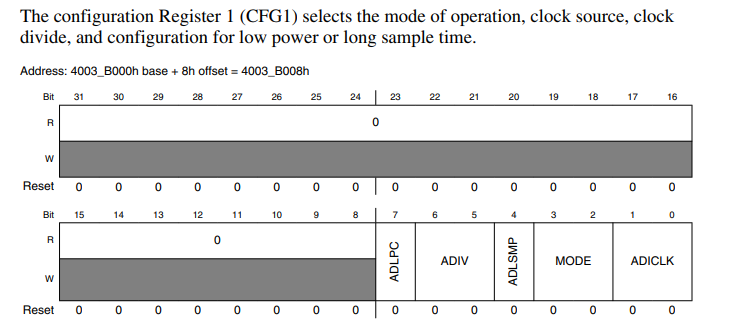


Fig.0.13 Descriere registru ADC0\_CFG1

* ADC0\_RA – se pune rezultatul conversiei.

Datele transmise aici(Rn) conțin rezultatul unei conversii ADC a canalului selectat de starea și registrul de control al canalului corespunzător (SC1A:SC1n). Pentru fiecare registru de control al stării și canalului, există un registru de rezultat al datelor corespunzător.Biții neutilizați din R n sunt șterși în moduri justificate la dreapta nesemnați și poartă bitul cu semn(MSB) în modurile complement 2 cu semn extins. De exemplu, când este configurat pentru 10 biți modul single-ended, D[15:10] sunt șterse. Când este configurat pentru modul diferenţial pe 11 biţi,D[15:10] poartă semnul bitului, adică bitul 10 extins prin bitul 15.

* PIT ( Periodic Interrupt Timer )

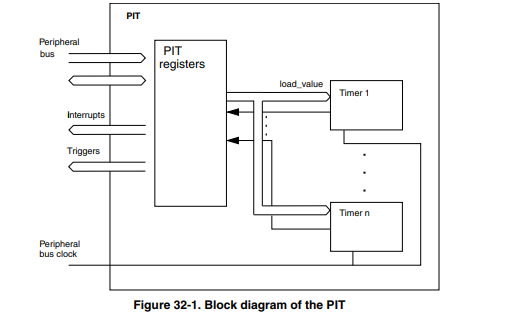
Folosim PIT pentru timer.

Fig.0.15 Schema block a modulului PIT

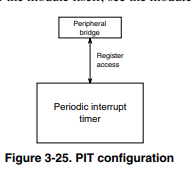


Fig.0.14 Configurare PIT

Fig.0.14 si Fig.0.15 ajuta la intelegera modului de functionare a perifericului PIT.

* SIM\_SCGC6 – pentru a activa PIT (setam bitul 23)->Fig.0.16 evidentiaza descrierea registrului SIM\_SCGC6 si corespondenta dintre bitul 23 si modulul PIT

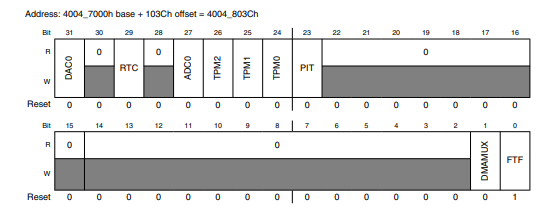


Fig.0.16 Descriere registru SIM\_SCGC6

* PIT\_MCR- activeaza si dezactiveaza timerul PIT (Bitul 30 – Module Disable: 0- Enable /1-Disable)
* PIT\_LDVAL1 – seteaza valoarea de start a timerului-> Fig.0.17 evidentiaza continutul registrului PIT\_LDVAL1

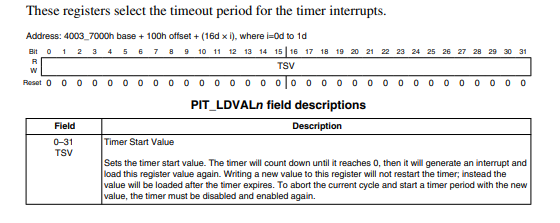


Fig.0.17 Descriere registru PIT\_LDVAL1

* PIT\_TCTRL1-setam time interrupt enable (bitul 30 )-> Fig.0.18 evidentiaza continutul registrului PIT\_TCTRL1 si corespondenta dintre 30 si Time Interrupt Enable

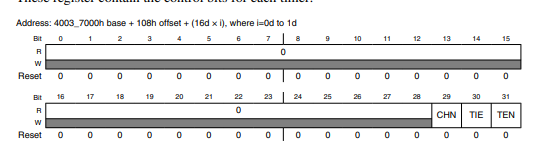


Fig.0.18 Descriere registru PIT\_TCTRL1

# Pasi in rularea programului

Pentru a putea rula programul, trebuie sa conectam senzorul la placuta. Conform (Schema Senzor) stim ca trebuie sa conectam firul negru la GND, firul rosu la alimentare si firul albastru la analog out.

Asadar pentru a vedea la ce pini conectam senzorul pe placuta am consultat (Schematic platforma)si am conectat firul negru la pinul 7 de pe placuta deoarece acesta este GND, firul rosu la pinul 4 deoarece acesta are alimentare de 3.3V (cat are nevoie si senzorul), iar firul albastru se leaga la PTB0 , pe pin-ul 9.(Fig. 1).

Pentru a efectua schema bloc din Fig.0.19 am folosit (Draw.io).In Fig.0.19 este evidentiata comuicatia dintre sensor, placuta si laptopul care va afisa datele.

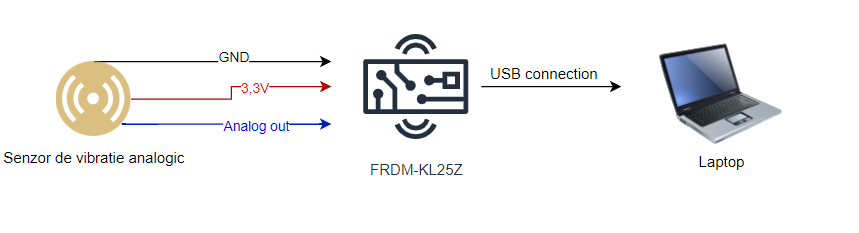


Fig.0.19 Schema bloc a infrastructurii

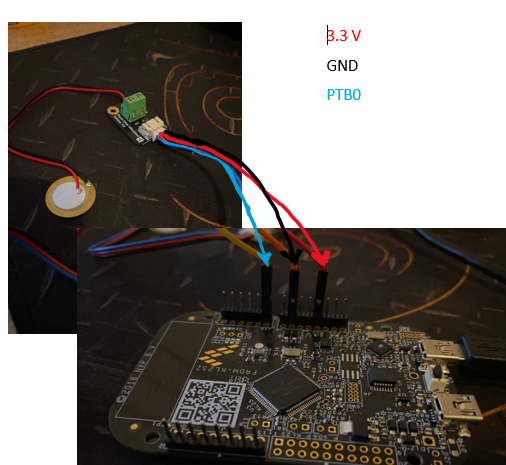


Fig. 1: Conectarea intre placuta si senzor

Dupa conectarea senzorului la placuta, conectam placuta la laptop prin intermediul cablului USB, care se introduce in placuta pe portul SDA.Pentru a incarca programul pe placa vom face Build si Load in Keil.

Dupa efectuarea acestor configurari se va rula scriptul script.py, care va plota datele primite de la senzor astfel:

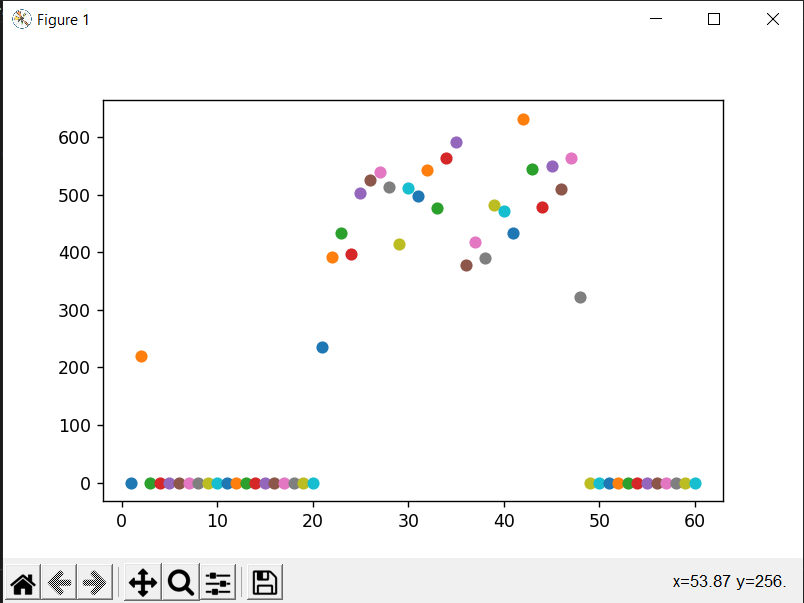


Fig. 2 Reprezentare grafica

Pe axa Ox este evolutia in timp, iar pe Oy se afiseaza valorile trimise de senzor, care reprezinta intensitatea vibratiei. Se observa ca la detectarea unei vibratii din partea senzorului, pe grafic apar valori in care se regaseste nivelul de intensitate al vibratiei. Daca acestea nu exista, la fiecare secunda se va afisa 0.Aceste observatii se fac pe baza Fig. 2, care evidentiaza reprezentarea grafica a datelor primite in timp real de la senzor.

# Probleme intampinate

* Afisarea graficului in timp real in python:

Reusim afisarea graficului in timp real cu ajutorul functiei scatter, insa aceasta afiseaza graficul sub forma de puncte, iar noi doream o afisare continua, sub forma unei linii.

# Referințe

*Draw.io*. (fără an). Preluat de pe https://app.diagrams.net/: https://app.diagrams.net/

*Platforma*. (fără an). Preluat de pe https://os.mbed.com/platforms/KL25Z/

*Reference Manual*. (fără an). Preluat de pe https://spivey.oriel.ox.ac.uk/dswiki/images-digisys/5/56/KL25-refman.pdf

*Schema Senzor*. (fără an). Preluat de pe Arrow: 3. https://static6.arrow.com/aropdfconversion/167b6b55a929a37e29a9b0e40b6dd662c54db14/dfr00 52\_web.pdf

*Schematic platforma*. (fără an). Preluat de pe https://www.nxp.com/downloads/en/schematics/FRDM-KL25Z\_SCH\_REV\_E.pdf

Sursa utilizata la scrierea codului:

https://github.com/undacmic/FRDM-KL25Z-Real-time-sound-detection