**ROMÂNIA**

**MINISTERUL APĂRĂRII NAȚIONALE**

**ACADEMIA TEHNICĂ MILITARĂ „FERDINAND I”**

**Facultatea de Sisteme Informatice și Securitate Cibernetică**

**Departamentul de Calculatoare și Securitate Cibernetică**



***Utilizare senzorI INTEGRATI RGB SI TSI SI SENZOR DIGITAL PUSH BUTTON***

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

Std. Plt. Florin-Gabriel CABAUA

Std. Sg. Maj. George-Codrin ANDRONACHE

Std. Sg. Maj. Cristian MOGODAN

Grupa C114B

**București**

**2023**

Cuprins

[1. Prezentarea senzorului TSI 3](#_Toc124273821)

[2. Prezentarea led-ului RGB 4](#_Toc124273822)

[3. Prezentare Push Button DFR0029-Y 4](#_Toc124273823)

[4. Scopul proiectului 6](#_Toc124273824)

[5. Conectare senzor – placa de dezvoltare 6](#_Toc124273825)

[6. Descriere program 7](#_Toc124273826)

[6.1. Touch Sensing Input 7](#_Toc124273827)

[6.1.1. Functia main 7](#_Toc124273828)

[6.1.2. Libraria mbed.h 7](#_Toc124273829)

[6.1.4. Functia main loop 9](#_Toc124273830)

[6.1.5. Reprezentarea grafica 10](#_Toc124273831)

[6.2. Push Button 13](#_Toc124273832)

[6.2.1. Functia main 13](#_Toc124273833)

[6.2.2. Inițializarea modulelor 14](#_Toc124273834)

[6.2.3. Initializarea modulului PIT 15](#_Toc124273835)

[6.2.4. Initializarea modulului GPIO 16](#_Toc124273836)

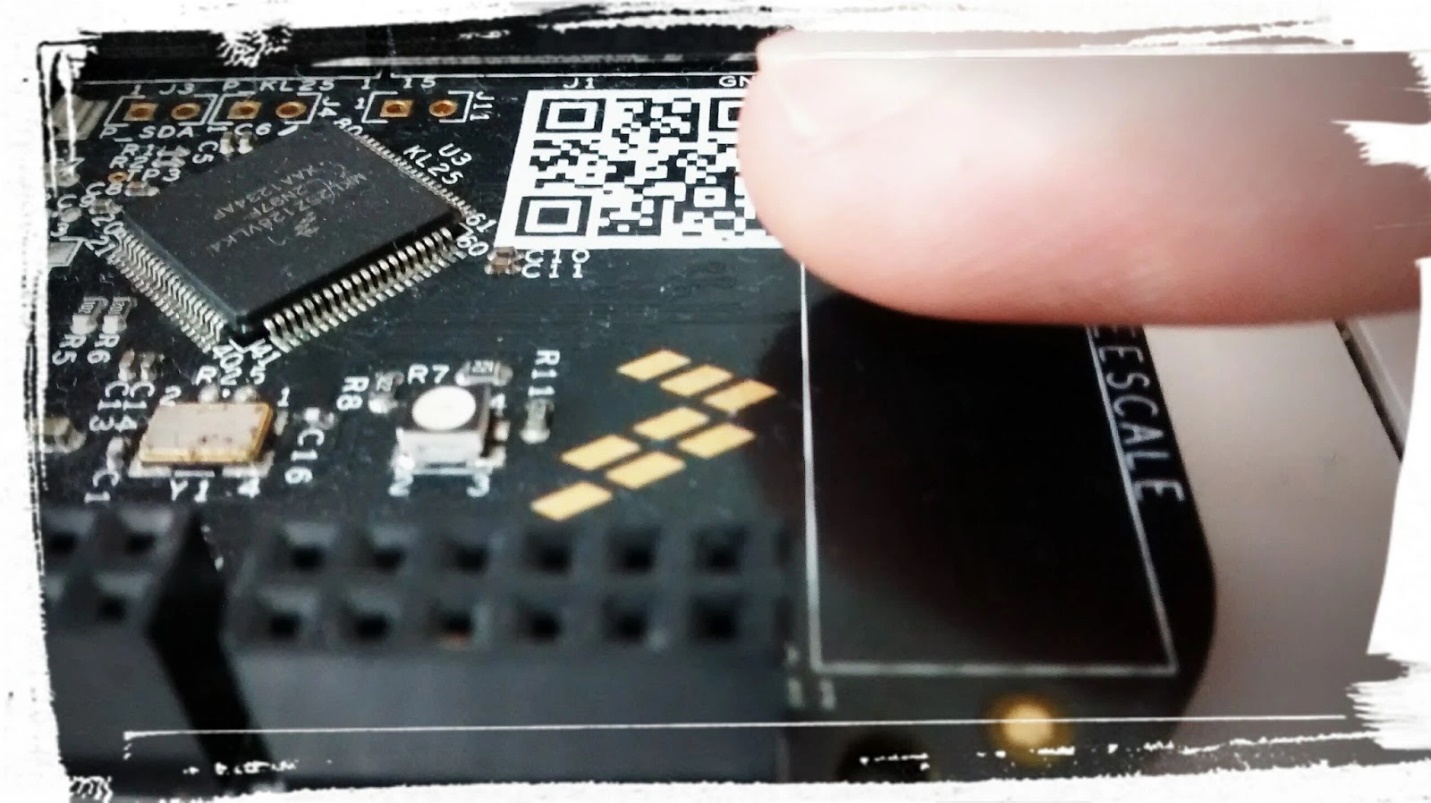
[6.2.5. Transmitere date prin UART 16](#_Toc124273837)

[7. Dificultati intampinate 17](#_Toc124273838)

# Prezentarea senzorului TSI

FRDM-KL25Z este o placă de dezvoltare pentru microcontrolerul Kinetis KL25Z Cortex-M0+ de la NXP. Placa dispune de un senzor de temperatură integrat (TSI) care vă permite să măsurați temperatura dispozitivului. TSI este o tehnologie de detectare capacitivă care măsoară modificarea capacității cauzată de o schimbare a temperaturii.

TSI-ul de pe FRDM-KL25Z poate fi utilizat pentru a măsura temperatura dispozitivului atât în modul inactiv, cât și în cel activ, iar măsurarea poate fi efectuată fie în mod continuu, fie la cerere. Senzorul poate măsura un interval de temperatură de la -40 la 85 de grade Celsius cu o rezoluție de 0,03125 de grade. Datele măsurate pot fi citite folosind microcontrolerul de bord și pot fi folosite pentru a controla diferite funcții ale dispozitivului.



# Prezentarea led-ului RGB

LED-ul RGB de pe placa de dezvoltare FRDM-KL25Z este o diodă emițătoare de lumină multicoloră care poate produce o gamă largă de culori prin amestecarea culorilor primare roșu, verde și albastru. LED-ul este controlat de microcontrolerul de bord și poate fi folosit pentru a indica starea dispozitivului, pentru a afișa efecte vizuale sau pentru a crea iluminare interactivă.

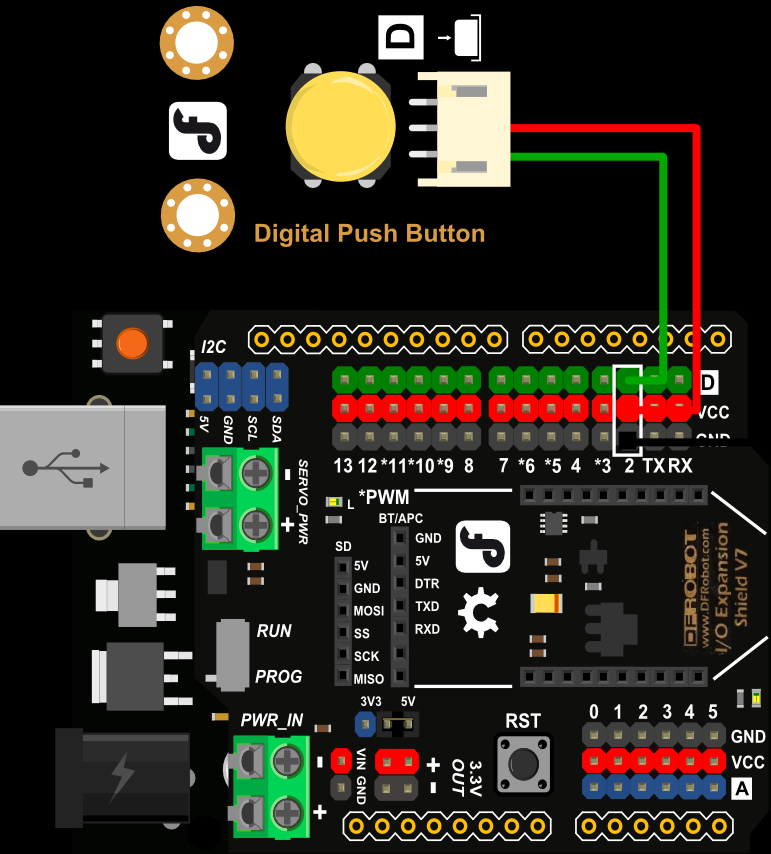
LED-ul RGB este conectat la microcontroler printr-un set de pini de ieșire digitală, iar culoarea LED-ului poate fi controlată prin setarea nivelurilor corespunzătoare pe acești pini. Luminozitatea fiecărei culori poate fi, de asemenea, controlată prin utilizarea Pulse-Width Modulation (PWM) pentru a varia ciclul de lucru al semnalului digital aplicat LED-ului.



# Prezentare Push Button DFR0029-Y

Butoanele noastre digitale reproiectate sunt dotate cu LED-uri, suprafață aurie de imersie și pălării colorate. O majusculă D în colțul din stânga indică faptul că este un senzor digital. O ilustrație a unui buton pictat în spate care anunță funcția acestuia. Apăsarea acestuia este o adevărată distracție, mai ales cu capacul de plastic scos. Durata lungă de viață presabilă îi asigură și durabilitatea.



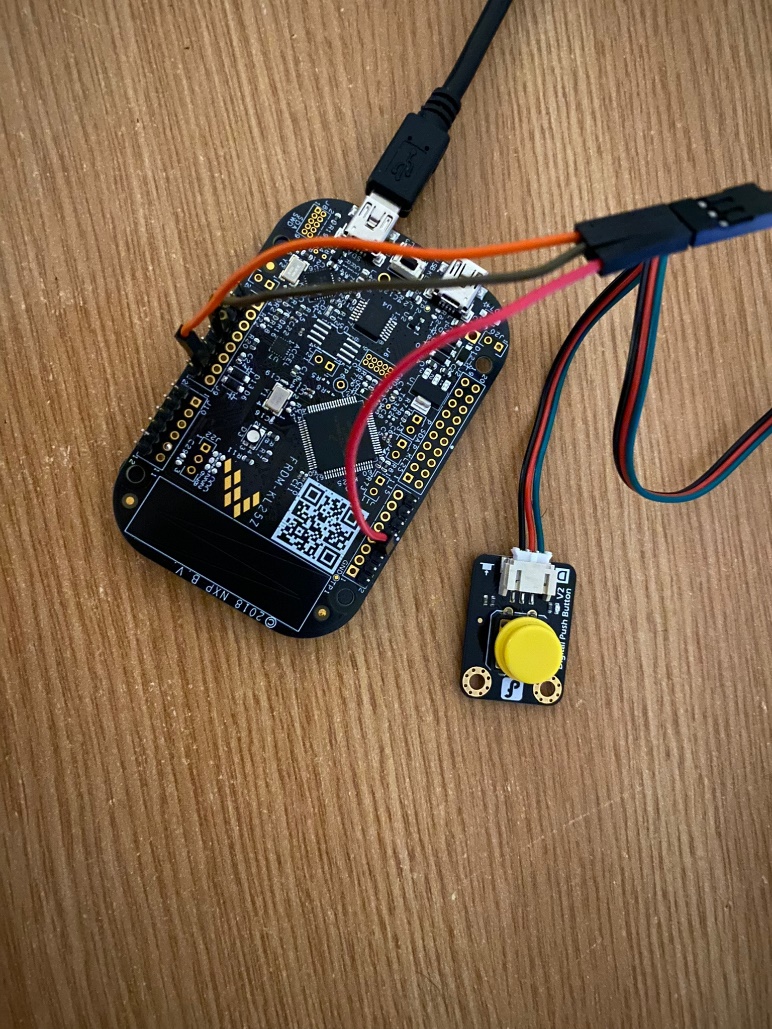


# Scopul proiectului

Scopul proiectului este acela de a implementa un numarator care sa contorizeze de cate ori a fost apasat push button-ul intr-un interval de timp stabilit din interfata grafica si prin utilizarea senzorului tactil integrat sa se modifice intensitatea led-ului RGB integrat pe placuta. Numarul de apasari de buton va fi afisat in final in interfata prin suprascrierea valorii anterioare.

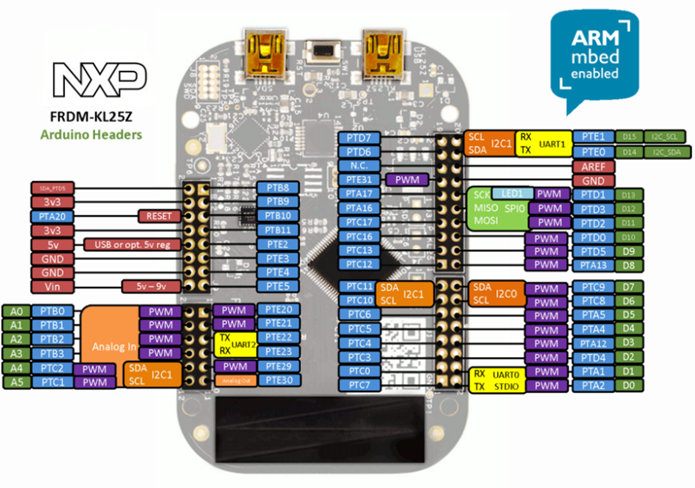
Afisarea unui grafic pentru fiecare informatie primita de la senzorul TSI. De asemenea, afisarea descrescatoare a numarului de secunde introdus in care se numara apasarile de buton.

# Conectare senzor – placa de dezvoltare



Componentele vor fi conectate astfel:

* Push Button:
  + Alimentare la 3.3V;
  + Ground la GND;
  + Pinul la PTA12 (FRDM-KL25Z);
* TSI: Integrat pe FRDM-KL25Z;
* RGB: Integrat pe FRDM-KL25Z.



# Descriere program

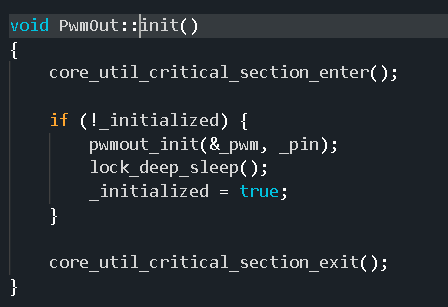
## Touch Sensing Input

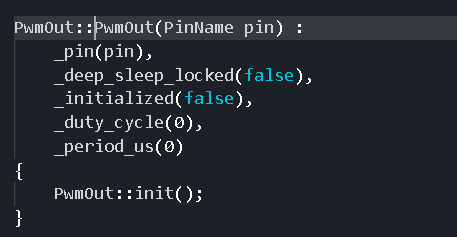
### Functia main

În fișierul main.cpp am inclus fisierele header specifice lucrului cu senzorul tactil, si anume: mbed.h și TSISensor.h. Vom folosi si stdio.h pentru trimiterea datelor catre script-ul de Python ce va reprezenta grafic valoarea intensitatii capacitorului.

### Libraria mbed.h

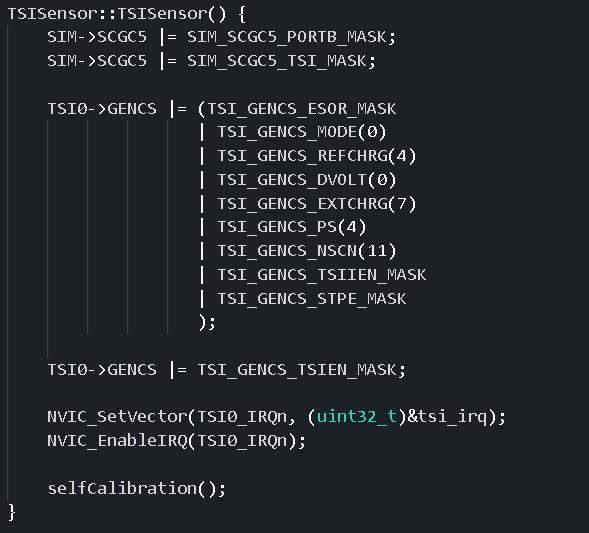
Din libraria mbed.h ne vom folosi de clasa PwmOut, care este folosita pentru a configura si controla periferalele de tip PWM pe un microcontroller. In cazul de fata, vom intializa o instanta a clasei PwmOut cu LED-ul de pe placuta de culoare albastra:

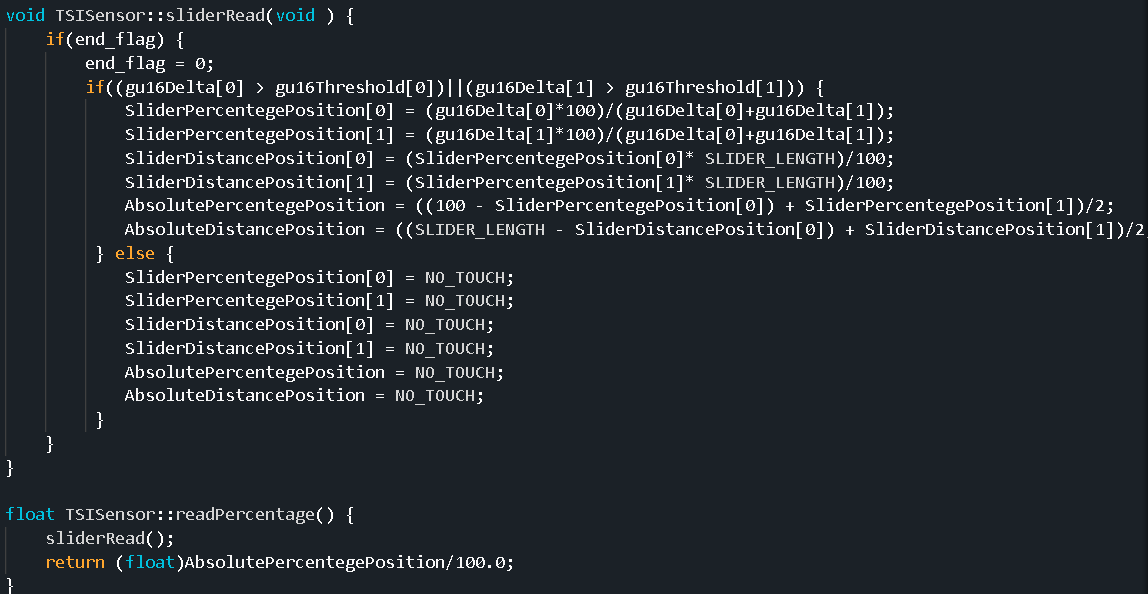




* + 1. *Libraria TSISensor.h*

Din libraria TSISensor.h ne vom folosi de clasa TSISensor, care va interactiona in mod direct cu senzorul tactil. Ne vom folosi de metoda readPercentage, care va intoarce valoarea intensitatii capacitorului.





### Functia main loop

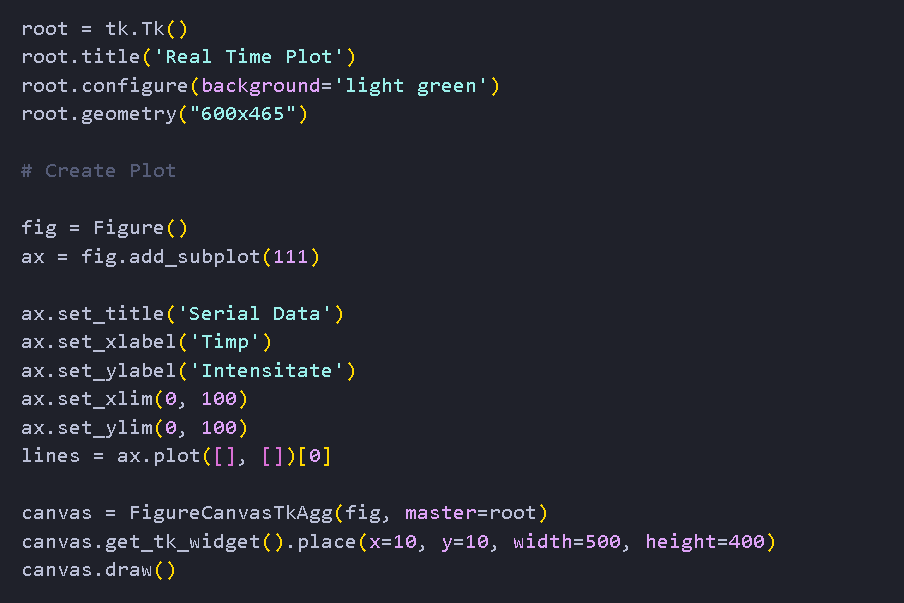
În bucla while, luminozitatea LED-ului este setată la 1, minus citirea senzorului tactil, în procente.



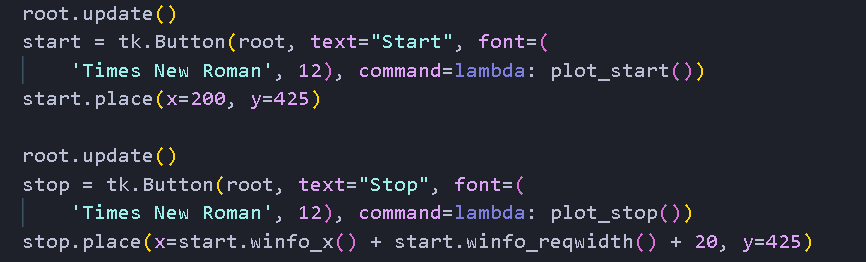
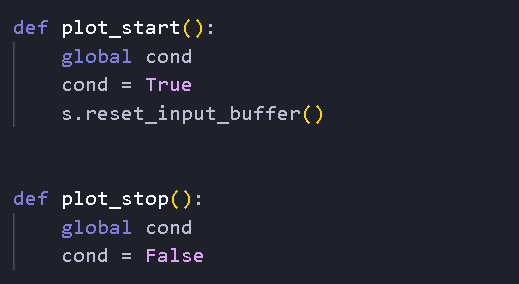
Această linie setează ciclul de funcționare al semnalului PWM la 1 minus citirea senzorului tactil, astfel încât LED-ul va fi aprins când senzorul tactil este atins și stins când nu este. Senzorul tactil citit este în procente.Printf este folosit pentru a trimite valoarea procentului citirii senzorului tactil în portul serial, astfel încât valoarea sa poata sa fie afisata grafic.Linia finală a buclei folosește biblioteca „ThisThread” pentru a astepta timp de 100 de milisecunde înainte de a repeta bucla, astfel încât luminozitatea LED-ului să poată fi observată și senzorul tactil să poată fi citit la o rată rezonabilă.

### Reprezentarea grafica

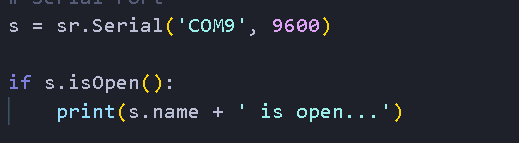
Pentru reprezentarea grafica ne vom folosi de un script de Python care foloseste libraria tkinter pentru crearea unei aplicatii de tip GUI care afiseaza date real-time dintr-un port serial. Datele sunt citite din portul serial folosind libraria pyserial si sunt reprezentate grafic cu libraria matplotlib.

Când programul este rulat, acesta creează o fereastră Tkinter cu titlul „Real Time Plot” și o culoare de fundal de verde deschis. Apoi creează o figură și îi adaugă un singur subplot (ax). Subplotul este configurat cu un titlu, etichete x și y și limite x și y.

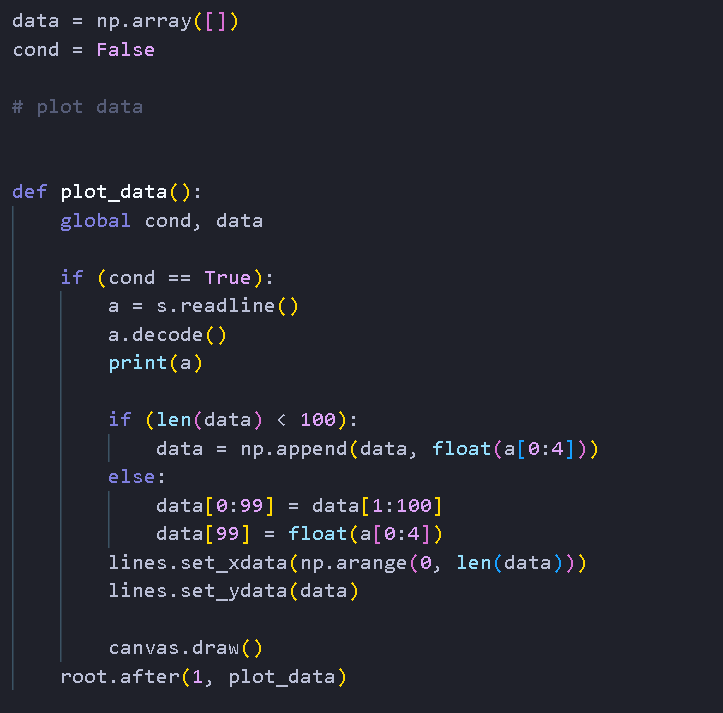
Apoi, programul creeaza doua butoane, “Start” si “Stop”, care pornesc si opresc reprezentarea grafica . Acest lucru se realizeaza modificand variabila globala “cond” de la true la false.

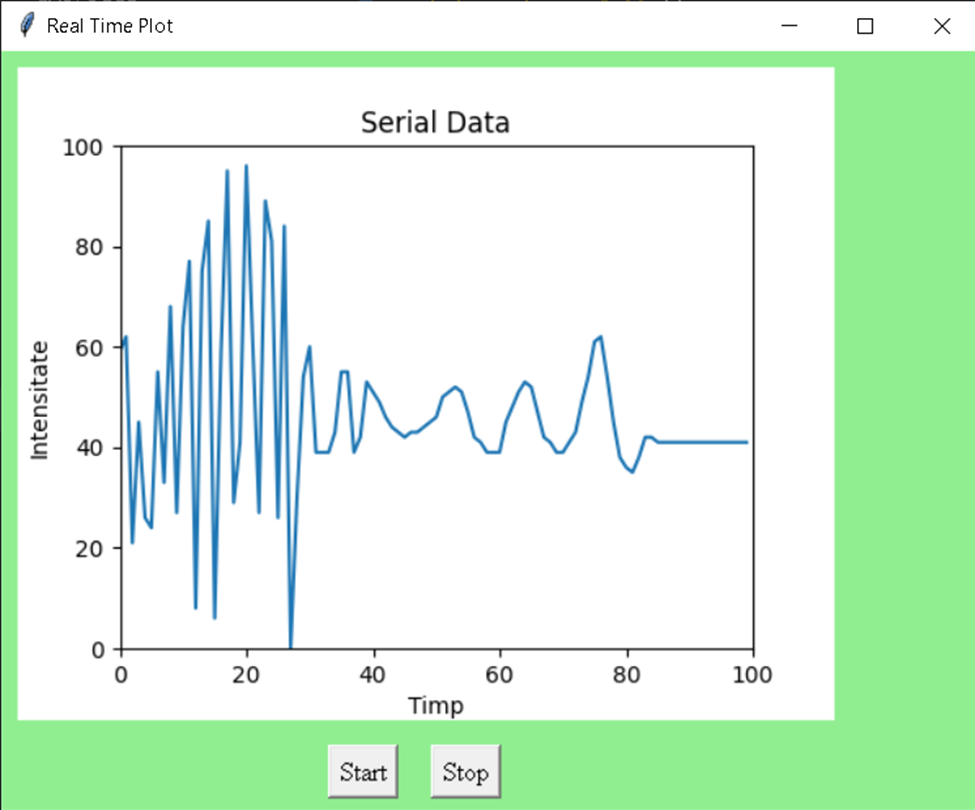


Programul deschide ulterior o conexiune seriala la placuta, care este conectata la “COM9”, cu un baud rate de 9600. Daca portul este deschis, afiseaza la consola acest lucru.



Programul va folosi functia after pentru a apela functia plot\_data la fiecare milisecunda. Functia plot\_data citeste datele din port, le adauga la un array, si face update la grafic, redesenand-ul dupa fiecare update.

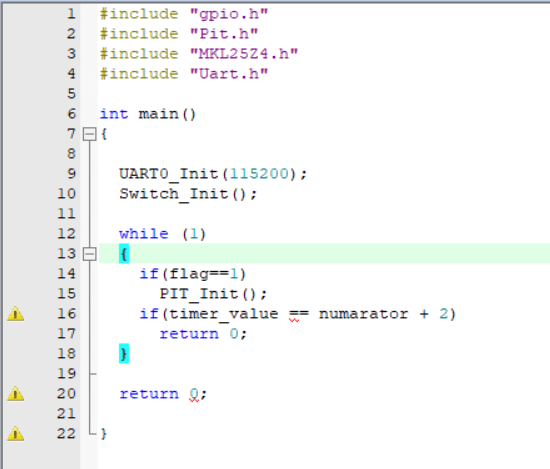


La final, interfata grafica va arata in felul urmator:

## Push Button

### Functia main

În fișierul main.c am inclus fișierele header în care sunt declarate funcții și variabile ce urmează a fi folosite: *gpio.h* (funcția Switch\_Init), *uart.h* (funcția UART0\_Init, variabila globala flag), *Pit.h* (funcția PIT\_Init) și fișierul header generat de către mediul de dezvoltare Keil, *MKL25Z4.h*, specific plăcii de dezvoltare.

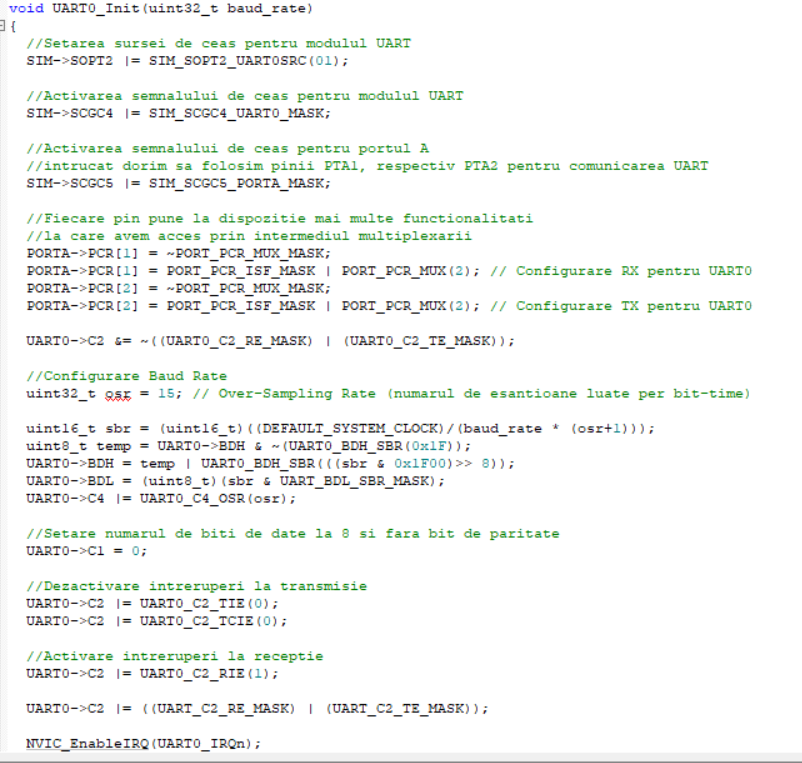


Logica principală a programului este următoarea: întâi se apelează funcțiile de inițializare a modulelor UART si GPIO, apoi, într-un ciclu infinit, se verifică daca sunt trimise date prin interfata seriala – mai exact durata in care se vor contoriza apasarile de buton urmand ca odata primita valoarea setata din interfata grafica sa se initializeze modulul de PIT.

### Inițializarea modulelor

#### Initializarea modulului UART

Vom folosi modulul UART0 pentru comunicația serială cu PC prin cablul USB.



În registrul SIM\_SOPT2 (System Options Register 2) setăm pe 01 câmpul UART0SRC (biții 27-26) pentru selectarea ca ceas al modulului MCGFLLCLK anterior configurat.

Graphical user interface, text, application, email

Description automatically generated

În registrul SIM\_SCGC4 (System Clock Gating Control Register 4) setăm pe 1 câmpul UART0 (bitul 10) pentru activarea ceasului pentru acest modul.

Text

Description automatically generated with medium confidence

În registrul SIM\_SCGC5 (System Clock Gating Control Register 5) setăm pe 1 câmpul PORTA (bitul 9) pentru activarea ceasului acestui port.

Graphical user interface, text, application

Description automatically generated

În regiștrii de control ai pinilor 1 și 2 din portul A (PORTA\_PCR1/2), setăm câmpul MUX (biții 10-8) pe valoarea 2, care înseamnă folosirea acestora în modulul de UART0 (RX/TX).



### Initializarea modulului PIT

Text

Description automatically generatedVom folosi modulul PIT pentru a executa un anumit set de instructiuni la o anumita durata de timp.

În registrul PIT\_CHANNEL[0].LDVAL setam durata de timp, respectiv 1 secunda, la care se va executa setul de instructiuni.

### Initializarea modulului GPIO

Vom folosi modelului GPIO pentru utilizarea push button.

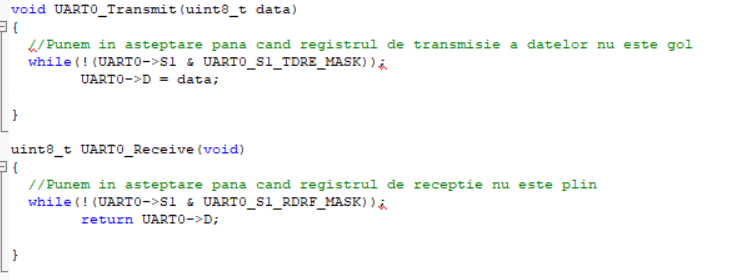
Graphical user interface, text, application

Description automatically generated

În registrul SIM\_SCGC5 (System Clock Gating Control Register 5) setăm pe 1 câmpul PORTA (bitul 9) pentru activarea ceasului acestui port.

În registrul PORTA\_PCR[12] setam valoarea de multiplexare pe 1, adica GPIO apoi activarea intreruperii pe rising edge.

### Transmitere date prin UART



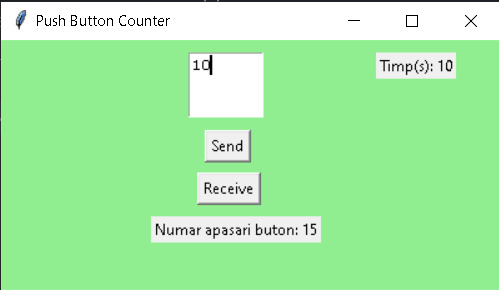
O dată la 10 ms, este apelată funcția UART0\_Transmit ce primește ca parametru valoarea primită de la senzor.

Valoarea octetului dat ca parametru este pus în registrul UART0\_D (UART Data Register). Nu se părăsește funcția până când câmpul TDRE (Transmit Data Register Empty Flag) din registrul UART0\_S1 (UART Status Register 1) este 1, adică bufferul de trimitere s-a golit.

Text

Description automatically generated

O dată la 10 ms, este apelată funcția UART0\_Receive ce primește ca parametru valoarea primită de la tastatura (atunci cand este trimis).

In final, interfata grafica va arata astfel:

# Dificultati intampinate

In timpul realizarii acestui proiect, am intampinat dificultati, precum:

* Din cauza sensibilitatii prea mari a push button-ului, apasarile de buton variau cu +-2 apasari la afisarea rezultatului;
* Normalizarea datelor primite de la TSI in timpul masurarii intensitatii led-ului RGB integrat pe placuta.