**ROMÂNIA**

**MINISTERUL APĂRĂRII NAŢIONALE**

**ACADEMIA TEHNICĂ MILITARĂ „FERDINAND I”**

**FACULTATEA DE SISTEME INFORMATICE ȘI SECURITATE CIBERNETICĂ**



**Proiect Microprocesoare**

Echipa IPC:

Istoc Alexandru

Pitulan Antonia

Costan Raluca-Ionela

**Cuprins**

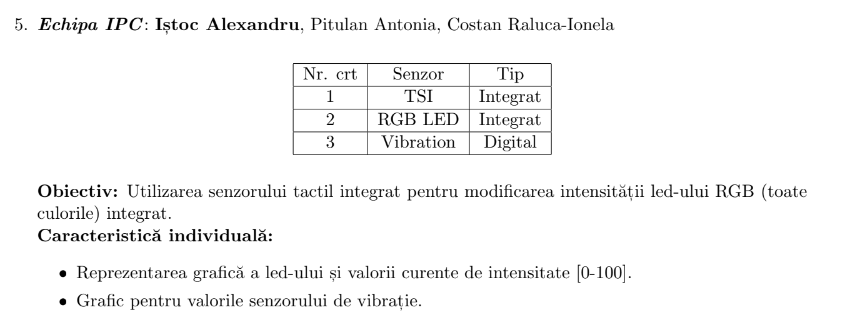
1. Obiective proiect
2. Configurare  
    2.1. Configurare senzor tactil   
    2.2. Configurare led-uri RGB  
    2.3. Configurare modul GPIO pentru senzorul de vibratii  
    2.4. Configurare modul UART   
   3.Senzorul de vibratii  
    3.1. Prezentarea senzorului  
    3.2.Reprezentare grafica a modului de conectare al senzorilor   
    3.3. Configurare senzor digital de vibratie

4. Setup  
 4.1. Setup senzor de vibratii  
 4.2. Setup senzor tactil si RGB

5. Probleme intampinate

6. Bibliografie

1. **Obiective proiect**

Proiectul curent urmareste 3 obiective:

1. Utilizarea senzorului tactil integrat pentru modificarea intensitatii led-ului RGB integrat.   
   Led-ul RGB va avea toate cele 3 culori active, astfel obtinandu-se culoarea alba.
2. Transmiterea si interpretarea datelor de la senzorul digital de vibratii
3. Realizarea unei interfete grafice care sa prezinte doua grafice in timp real, unul pentru fiecare senzor folosit

Cateva exemple de cazuri in care ar putea fi folosit in viata de zi cu zi:

- modificarea intensitatii unui display folosind comanda tactila

- modificarea intensitatii unor becuri inteligente folosind comanda tactila

1. **Configurare  
    2.1. Configurare senzor tactil**

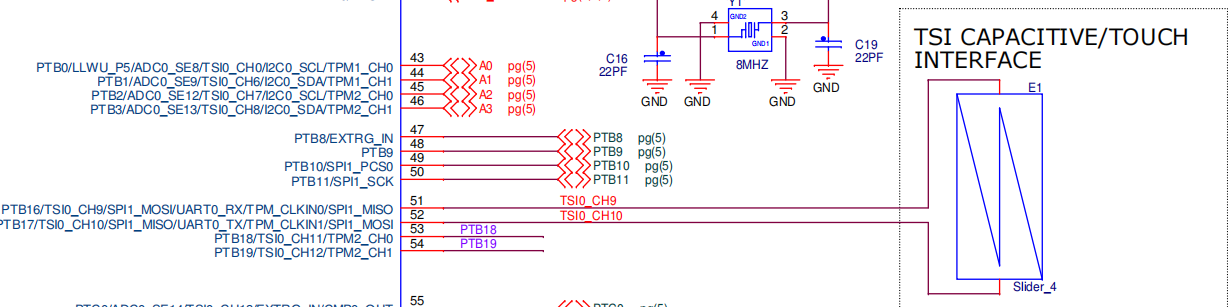
Senzorul tactil este integrat pe placuta si poate fi accesat folosind porturile PTB16 si PTB17.

Fiecare pin de TSI efectueaza o masurare capacitiva prin scanarea sursei de curent/ alimentare, incarcarii si descarcarii de electrozi/electrice, o data sau de mai multe ori.

Un oscilator de referință marcheaza timpul de scanare și stochează rezultatul intr-un registru pe 16 biți pana la terminarea scanarii.

Intre timp, o cerere de intrerupere este trimisa catre procesor pentru post-procesare daca intreruperea TSI este activata si functia de lucru cu DMA nu este selectata.

Modulul TSI poate lucra folosind curenti de intensitate mica, astfel salvand energie.



(Extras din FRDM-KL25Z\_Schematics.pdf )

Din Schematics observam ca vom utiliza modulul TSI0, pe canalele 9 si 10.

Modulul TSI are trei registrii:

1. GENCSeste registrul general de status si control, prin intermediul caruia putem configura modulul TSI
2. DATA folosit pentru a accesa valorile date de senzor
3. TSHDregistrul de threshold

Pentru a folosi acest senzor, el trebuie sa fie initializat si configurat.  
In cod, functia responsabila de aceasta etapa este urmatoarea si consta in activarea ceasului pentru PTB16 SI PTB17, si modificarea registrului GENCS al moduluilui TSI0:

void Touch\_Init()

{

// Enable clock for TSI PortB 16 and 17

SIM->SCGC5 |= SIM\_SCGC5\_TSI\_MASK;

TSI0->GENCS = TSI\_GENCS\_OUTRGF\_MASK | // Out of range flag, set to 1 to clear

TSI\_GENCS\_MODE(0u) | // Set at 0 for capacitive sensing. Other settings are 4 and 8 for threshold detection, and 12 for noise detection

TSI\_GENCS\_REFCHRG(0u) | // 0-7 for Reference charge

TSI\_GENCS\_DVOLT(0u) | // 0-3 sets the Voltage range

TSI\_GENCS\_EXTCHRG(0u) | //0-7 for External charge

TSI\_GENCS\_PS(0u) | // 0-7 for electrode prescaler

TSI\_GENCS\_NSCN(31u) | // 0-31 + 1 for number of scans per electrode

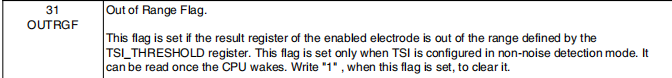
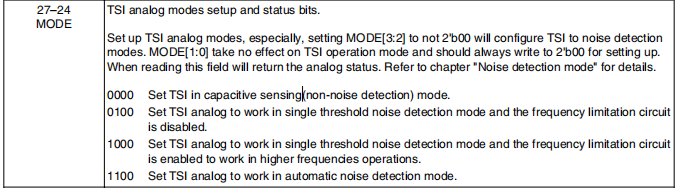
TSI\_GENCS\_TSIEN\_MASK | // TSI enable bit

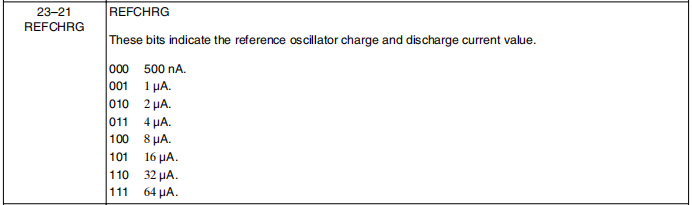
TSI\_GENCS\_STPE\_MASK | // Enables TSI in low power mode

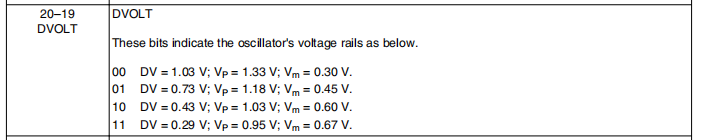
TSI\_GENCS\_EOSF\_MASK ; // End of scan flag, set to 1 to clear

}

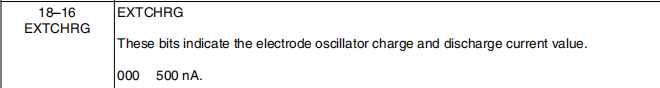
Descriere registrii folositi in functia de initializare :

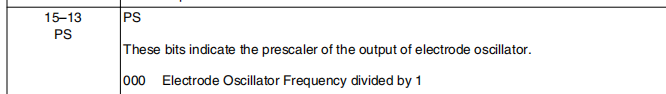
* **TSI\_GENCS\_OUTRGF** este bitul 32 din registrul GENCS si reprezinta campul de out of range.   
   Peste acesta vom aplica masca TSI\_GENCS\_OUTRGF\_MASK care are valoarea 0x80000000u, iar in binar 10000000000000000000000000000000, ceea ce inseamna ca i-am resetat valoarea 
* **TSI\_GENCS\_MODE**se intinde pe bitii 27-24 din registrul GENCS si reprezinta campul de configurare si status al modurilor de functionare analogice.   
   Ii vom da valoarea 0, reprezentand modul fara detectie de zgomot. 
* Campurile **TSI\_GENCS\_REFCHRG, TSI\_GENCS\_DVOLT,** seteaza intensitatile de incarcare si descarcare, precum si voltajul oscilatorului de referinta. Pentru amandoua am asignat valoarea de la prima varianta posibila.

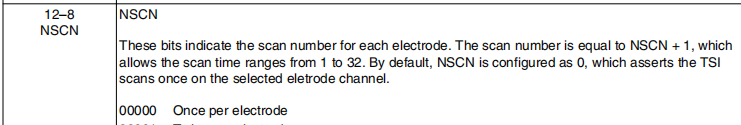


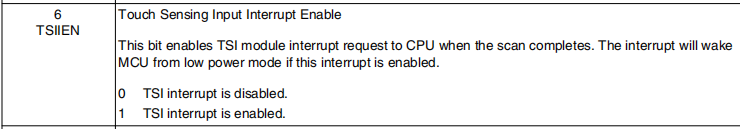
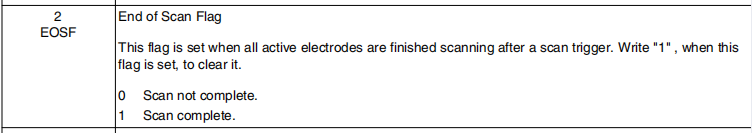


* **TSI\_GENCS\_EXTCHRG,TSI\_GENCS\_PS** si **TSI\_GENCS\_NSCN  
  setez intensitatea curentului,valoarea de divizare a frecventei** Oscilatorului cu electrod si numarul de valori scanate de electrodul de pe canalul selectat.

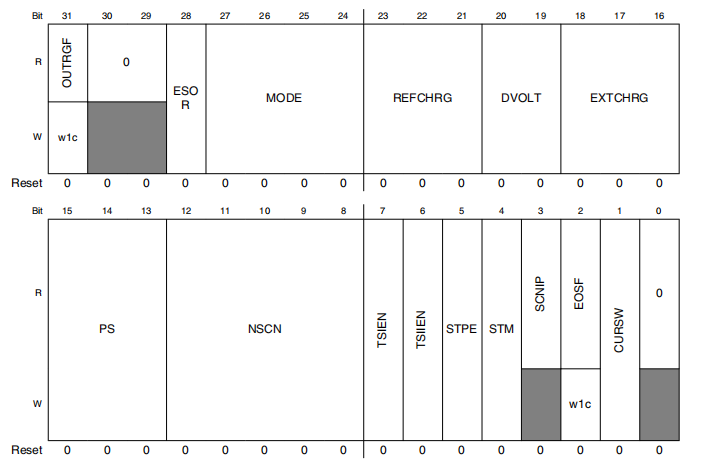






* **TSI\_GENCS\_TSIEN\_MASK** masca pentru activarea modulului TSI  
  
* **TSI\_GENCS\_STPE\_MASK** masca pentru a permite modului TSI sa functioneze in regim de energie redus
* **TSI\_GENCS\_EOSF\_MASK** masca pentru marcarea sfarsitului de citire a senzorului  
  

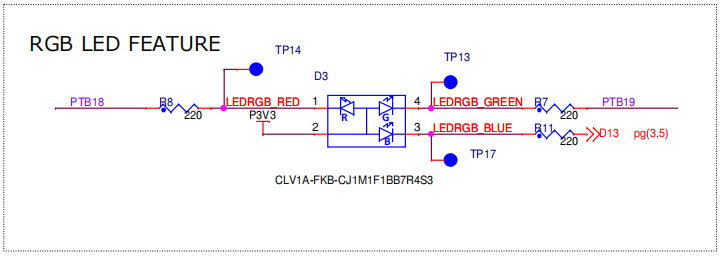
Motivarea folosirii si setarii acestor biti ai registrului GENCS este data de urmatoarea schema regasita in Referance Manual la capitolul 42.3.1.



**2.2. Configurare led-uri RGB**

In acest proiect am folosit led-ul RGB, acesta fiind integrat pe placuta.

Canalul Rosu poate fi accesat pe portul PTB18, cel verde pe PTB19, iar cel albastru pe PTD1.

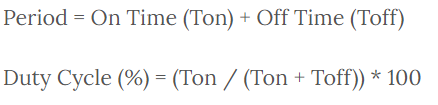
Schema extrasa din Schematics

|  |  |  |  |
| --- | --- | --- | --- |
| On-board Usage | FRDM-KL25Z Pin Name | ALT 3 | ALT 4 |
| Red LED | PTB18 | FTM2\_CH0 |  |
| Green LED | PTB19 | FTM2\_CH1 |  |
| Blue LED | PTD1 |  | FTM0\_CH1 |

Tabel 1. extras din Pinouts

Intensitatea luminoasa a ledului RGB va fi modificata cu ajutorul modulului PWM.  
 Acest modul are trei parametrii principali: On Time, Off Time si Duty Cycle (factorul de umplere).

On time si Off Time formeaza impreuna perioada modulului PWM, iar factorul de umplere exprima raportul dintre On Time si Perioada, exprimat in procente.



In cod, avem urmatoarea functie care se ocupa de initializarea modulului PWM si a Led-ului RGB:

void pwm\_led\_init(void)

{

// Enable clock to PORTB and PORTD

SIM->SCGC5 |= SIM\_SCGC5\_PORTB\_MASK;

SIM->SCGC5 |= SIM\_SCGC5\_PORTD\_MASK;

// Activarea ceasului pentru fiecare led

PORTB\_PCR18 |= PORT\_PCR\_MUX(3);

PORTB\_PCR19 |= PORT\_PCR\_MUX(3);

PORTD\_PCR1 |= PORT\_PCR\_MUX(4);

// Select FLL as the Clock to TPM (PWM)

SIM->SOPT2 |= SIM\_SOPT2\_TPMSRC(1);

// Enable clock to TPM2 and TPM0 (PWM) Module

SIM->SCGC6 |= SIM\_SCGC6\_TPM2(1);

SIM->SCGC6 |= SIM\_SCGC6\_TPM0(1);

// Configure as edge aligned PWM

TPM2->SC |= (TPM\_SC\_CPWMS(0)| TPM\_SC\_CMOD(1));

TPM0->SC |= (TPM\_SC\_CPWMS(0)| TPM\_SC\_CMOD(1));

// Clock prescaler is 7 (PWM clock devided by 128)

// This makes PWM clock as 48000000/128 = 375000 Hz (375Khz)

TPM2->SC |= TPM\_SC\_PS(7);

TPM0->SC |= TPM\_SC\_PS(7);

// Configured as Edge Aligned PWM //table 31-34

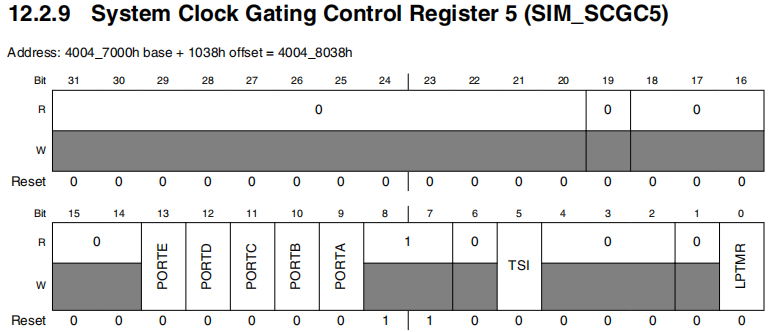
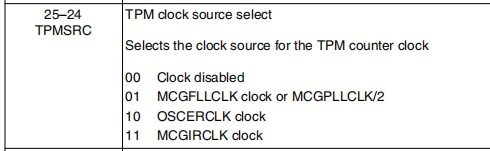
TPM2->CONTROLS[0].CnSC | = (TPM\_CnSC\_MSB\_MASK |TPM\_CnSC\_ELSB\_MASK);

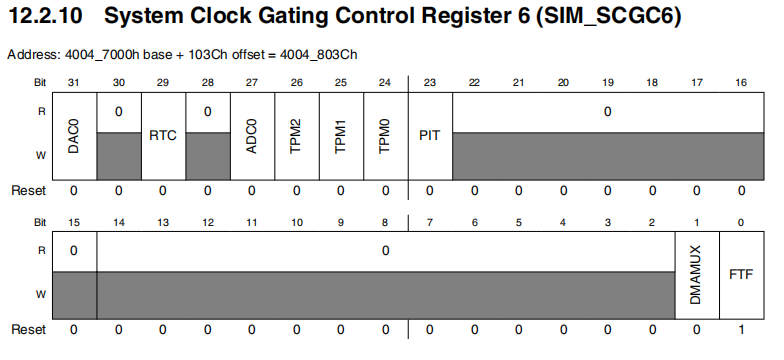
TPM2->CONTROLS[1].CnSC | = (TPM\_CnSC\_MSB\_MASK | TPM\_CnSC\_ELSB\_MASK);

TPM0->CONTROLS[1].CnSC | = (TPM\_CnSC\_MSB\_MASK | TPM\_CnSC\_ELSB\_MASK);

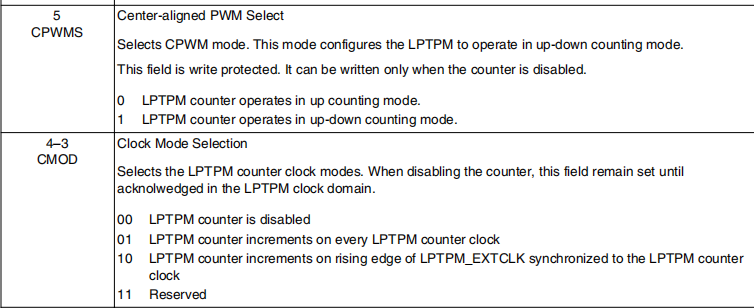
}

Descriere registrii folositi in functia de initializare:

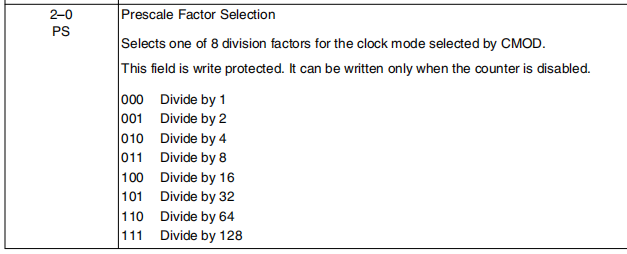
* Prima parte (primele 5 linii de cod), are doua roluri:  
  -primul de a initializa led-ul RGB, prin activarea ceasului pentru porturile B si D (modificarea valorii registrului **SIM->SCGC5** cu mastile corespunzatoare celor doua porturi)  
    
    
    
  -al doilea de a alege modul de lucru al fiecarui port, prin setarea valorii multiplexate **PORT\_PCR\_MUX** conform Tabelului 1.
* **SIM->SOPT2** selectarea sursei de ceas pentru numaratorul modulului TPM.  
   Am ales pentru acest registru alternativa 1, insemnand ca putem folosi fie intreaga frecventa MCGFLLCLK (care are valoarea de 48MHz), fie doar jumatate din aceasta.  
  
* **SIM->SCGC6** activarea semnalului de ceas pentru modulul TPM  
  Drept valori date s-au folosit masti care sa seteze bitii 26 si 24.



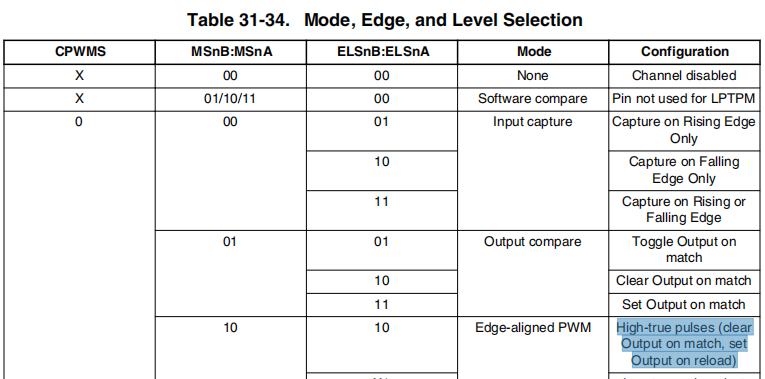
* **TPM2->SC si TPM0->SC** selectarea divizorului de frecventa si modului de functionare edge aligned.

Pentru a functiona in cazul edge aligned trebuie sa facem clear pe bitul 5, iar pentru a seta mod crescator pentru numarator, trebuie sa alegem varianta 01.  
   


Bitii 2-0 au rolul de a seta valoarea divizorului de frecventa. Asadar vom folosi o frecventa egala cu ceasului intern divizata la 128, adica am ales varianta 111( in binar), reprezentand 7 in baza10.



* **TPMn->CONTROLS[channel]. CnSC** acest registru sumarizeaza configuratia finala a modulului PWM.  
   In cod am utilizat doua masti TPM\_CnSC\_MSB\_MASK si TPM\_CnSC\_ELSB\_MASK al caror rezultat in urma operatiei or este 0x20u | 0x8u <=>100000 | 001000 = 101000, ceea ce ne conduce in urmatorul tabel din Referance Manual la varianta evidentiata cu albastru.

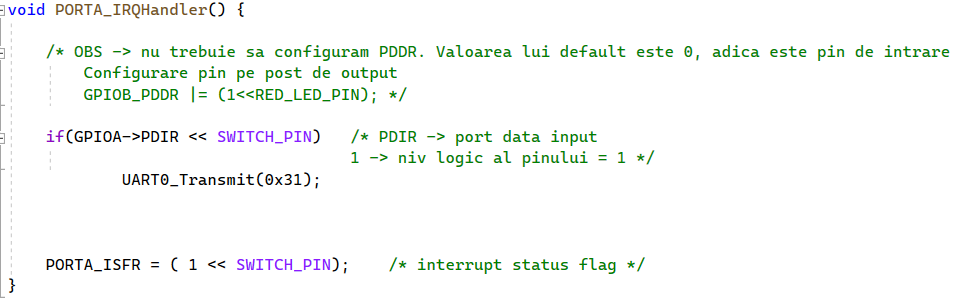


**2.3. Configurarea modulului GPIO pentru senzorul de vibratii**

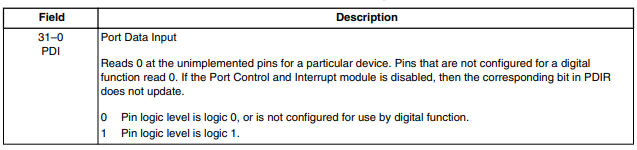
Modulul GPIO este folosit pentru a prelua informatiile de la senzor de pe pinul PTA12 definit la inceputul fisierului.

#define SWITCH\_PIN (12) /\* PORT A \*/

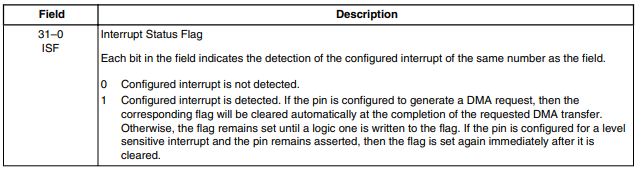
Se verifica daca sunt date in registru. In cazul in care sunt se transmite valoarea 1 catre UART.

****

**GPIOx\_PDIR(**Port Data Input Register**)**

****

**PORTx\_ISFR (**Interrupt Status Flag Register**)**

****

**2.4. Configurarea modulului UART**

In fisierul uart.c se va initia modulul UART care va prelua datele si le va afisa pe ecran. Ne vom

**Graphical user interface, text, application

Description automatically generated**

* **UART0\_S1 este un registru de status pentru modulul UART**

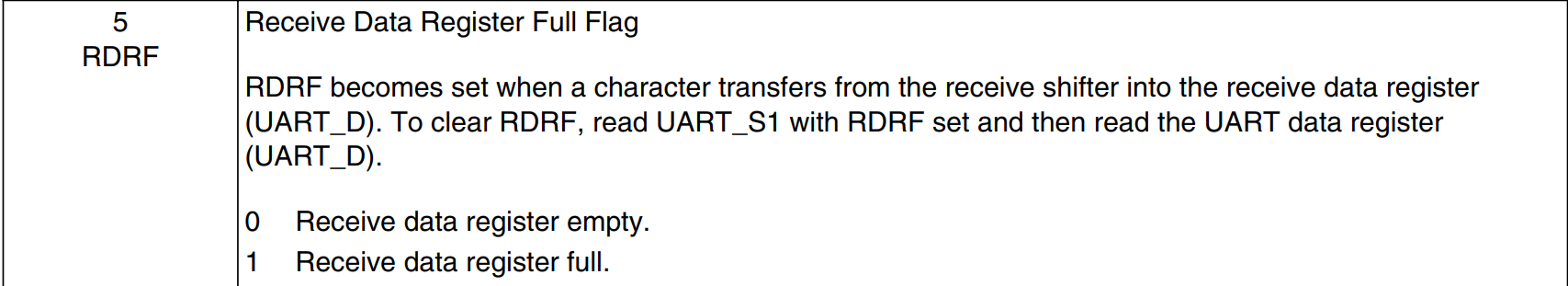
**Timeline

Description automatically generatedA screenshot of a computer

Description automatically generated**

* **UART0\_S1\_RDRF**

folosi de flag-ul RDRF din registrul UART0\_S1 pentru a vedea daca buffer-ul contine informatii pentru a fi afisate sau daca acesta este gol.

****

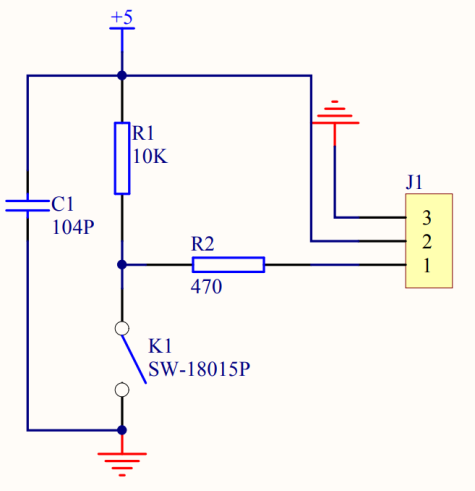
1. **Senzorul de vibratii  
    3.1. Prezentarea senzorului**

Senzorul DFR0027 este un sensor digital pentru detectarea vibratiilor.

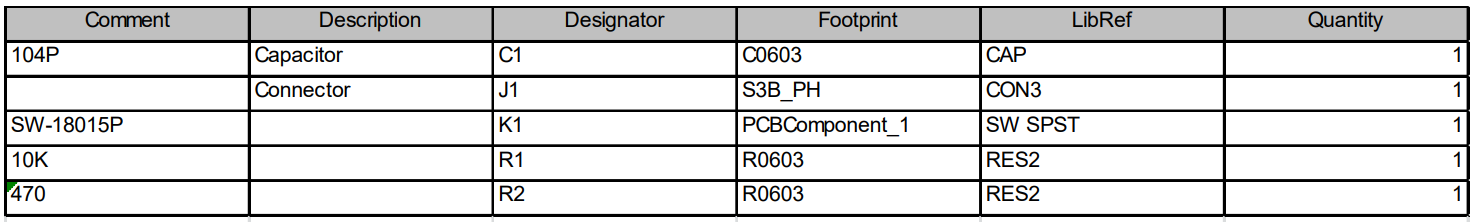
A close-up of a circuit board

Description automatically generated with medium confidence

*Fig 1. Imaginea senzorului*



*Fif 2. Circuitul senzorului*

**

**3.2 Mod de conectare**

Diagram

Description automatically generated with medium confidence

*Fig 3. Mod de conectare*

*A picture containing electronics

Description automatically generated*

Fir negru – conectat la GND

Fir alb – conectat la P3V3

Fir gri – conectat la PTA12

1. **Setup** Am impartit acest proiect in doua proiecte facute in Keil uVision5: unul pentru senzorul de vibratii, celalalt pentru modificarea intensitatii led-ului RGB prin utilizarea senzorului tactil.

**4.1. Setup senzor de vibratii**

Avem fisierele:

A picture containing table

Description automatically generated

* In ***gpio.c***: functiile *PIN\_Init()* si PORTA\_*IRQHandler()*
* In ***Uart.c***: functiile *UART0\_Init(), UART0\_Transmit(),* si *UART0\_IRQHandler()*
* In **main.c**: apelez functiile *UART0\_Init(), PIN\_Init()* si *UART0\_Transmit()*

**Main.c**

**Text

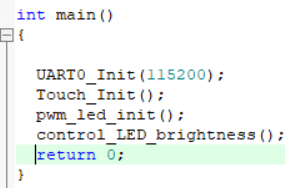
Description automatically generated**

In functia main() se initializeaza modulele si se creez o bucla infinita in care verific flagul TC din registrul UART0\_S1. Acesta verifica daca transmiterea datele preluate de la senzor a fost terminata. Prin flagul TC se anunta daca s-a terminat vibratia pe care senzorul o simte, caz in care acesta are valoarea 1.

Programul meu va afisa permanent valoarea 0(cu un anumit delay efectuat printr-o instructiune repetitiva de tip for() in carea variabila i ia valori de la 0 la 1.000.000), iar in momentul in care senzorul detecteaza vibratia se va pune valoare de citit in registrul de date.

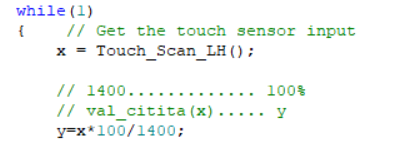
**4.2.Setup senzor tactil si RGB**

In sursa main.c avem urmatoarele apeluri de functii:



Primele trei sunt de initializare a modulelor Uart, TSI, PWM si LED, discutate in capitolul 2.

Functia principala este *contol\_LED\_brightness().*

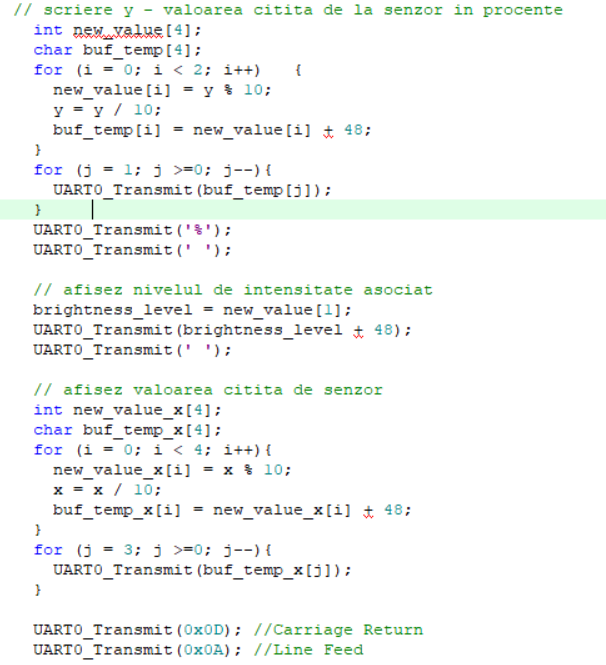


In aceasta functie avem o bucla infinita in care se apeleaza functia de citire a valorii de la TSI.

Se calculeaza apoi procentul pe care il reprezinta valoarea citita de la TSI, in functie de valoarea maxima pe care o poate returna modulul. Rezultatul obtinut este direct proportional cu pozitionarea degetului pe senzorul tactil.

Urmeaza afisarea in terminal a procentajului, nivelului de intensitate corespunzator, si valoarea returnata de functia de citire a modulului TSI.

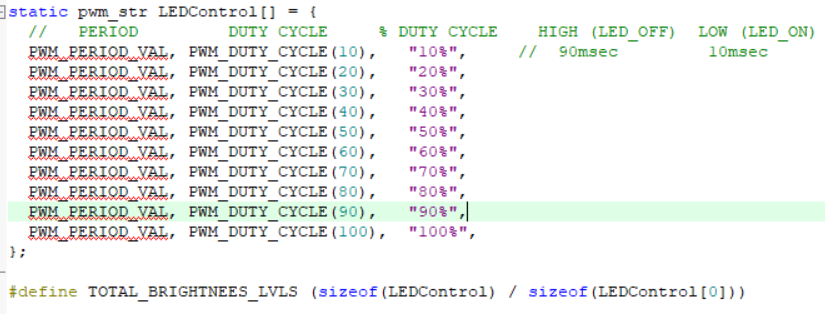
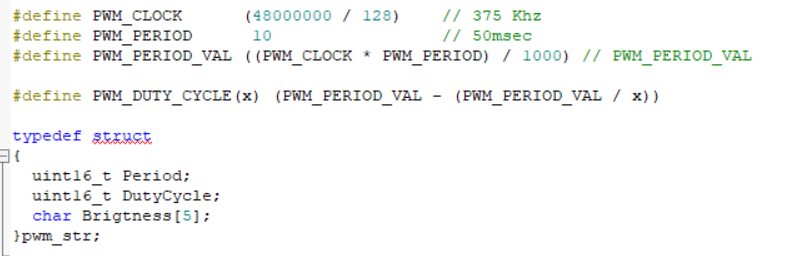
Variabila brightness\_level este calculata in functie de valoarea procentajului,fiind cifra zecimalelor, si poate lua valori intregi din intervalul 0-9.  
De exemplu: pentru procentajul cu valoarea 83, brightness\_level va avea valoarea 8.



Ultima parte a functiei este modificarea intensitatii in functie de brightness\_level, de structura *LEDControl si v*ariabila globala *TOTAL\_BRIGHTNEES\_LVLS.*

Structura *LEDControl* se foloseste de frecventa ceasului, perioada si factorul de umplere al modelului PWM configurat anterior.

Variabila *TOTAL\_BRIGHTNEES\_LVLS* reprezinta numarul total de intensitati pe care le poate lua led-ul RGB.

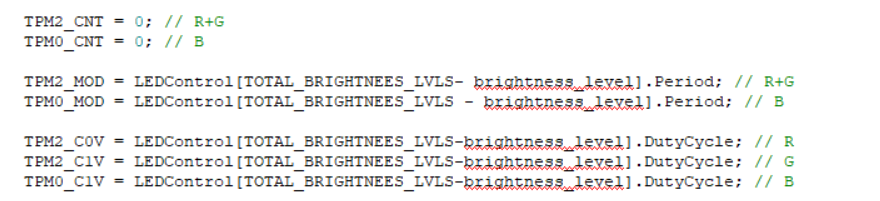


In registrul TMPx\_CNT se afla valoarea curenta a numaratorului modulului PWM. Pentru a nu influenta raspunsul luminos, acesta este setat pe 0.

In registrul TMPx\_MOD se afla valoarea pana la care trebuie sa numere numaratorul inainte de a genera impuls.

In registrul TMPx\_CnV se afla valoarea echivalenta factorului de umplere.

Deoarece led-ul RGB este activ pe low, pentru a obtine intensitatea exprimata de brightness\_level, trebuie de fapt sa setam duty cycle ca fiind *TOTAL\_BRIGHTNEES\_LVLS - brightness\_level*



1. **Probleme intampinate**

* Pentru senzorul de vibratie am folosit un pin pe care il conectam la placuta diferit de cel pe care il foloseam in cod
* Gasirea unor timpi de delay potriviti
* Gasirea valorii maxime a senzorului TSI
* Gasirea valorii potrivite de offset pentru senzorul TSI

**6.Bibliografie**

**FRDM-KL25Z\_Datasheet**

**FRDM-KL25Z\_Pinouts**

**FRDM-KL25Z\_Schematics**

**FRDM-KL25Z\_ReferenceManual**

**Senzor vibratie**

* <https://raw.githubusercontent.com/Arduinolibrary/DFRobot_Vibration_sensor/master/Digital%20Vibration%20Sensor%20SCH.pdf>
* <https://github.com/undacmic/MCULabs/blob/main/Resurse/FRDM-KL25Z_UserManual.pdf>
* <https://github.com/undacmic/MCULabs/blob/main/Resurse/FRDM-KL25Z_ReferenceManual.pdf>
* <https://wiki.dfrobot.com/DFRobot_Digital_Vibration_Sensor_V2_SKU_DFR0027>
* <file:///C:/Users/crist/Downloads/Digital%20Vibration%20Sensor%20SCH.pdf>

**Senzor TSI**

* <https://www.nxp.com/docs/en/application-note/AN4330.pdf>
* <https://forum.digikey.com/t/using-the-capacitive-touch-sensor-on-the-frdm-kl46z/13246>

**Modul PWM**

* <https://learningmicro.wordpress.com/controlling-led-brightness-using-pwm/>

**Desenarea graficelor in Python**

* <https://www.youtube.com/watch?v=0V-6pu1Gyp8&ab_channel=Dr.SachinSharma>