Modificarea unei valori numerice (0-F) afișate pe BCD prin intermediul acționării butonului push.



Proiect realizat de: Clapa Adrian

Corățu Luca

Stoica Alexandru

Cuprins

1.Scopul proiectului……………………………………………………………..3

2.Simularea Circuitului……………………………………………………….....3

3.Periferice folosite……………………………………………………………...3

4. Configurare…………………………………………………………………...4

5. Inițializarea modulelor………………………………………………………..5

5.1 Inițializarea BCD……………………………………………………..5

5.2Inițializarea Push-button………………………………………………8

5.3 Inițializare UART…………………………………………………….8

6.Descrierea modului de funcționare……………………………………………9

6.1 Transmiterea si primirea de date pe UART…………………………..9

6.2 Utilizarea întreruperilor……………………………………………..10

6.2.1 Înreruperi pe UART………………………………………..10

6.2.2 Întreruperi pe portul A……………………………………...11

6.3 Modificarea valorii afișate…………………………………………..11

6.4 Interfața grafică……………………………………………………...13

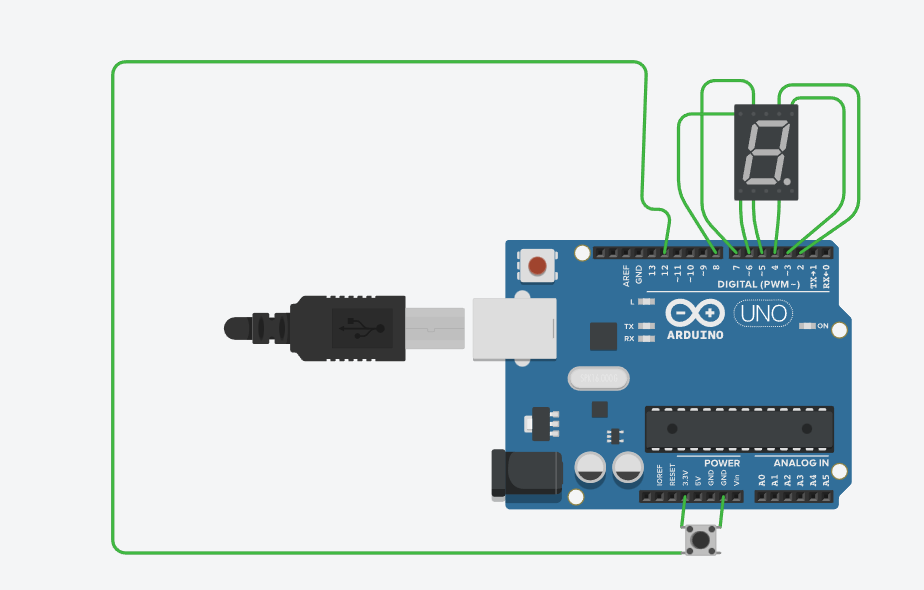
7. Dificultăți întâlnite…………………………………………………………..14

# Scop Proiect:

Proiectul propus are ca scop crearea unui numărător de valori hexazecimale cuprinse în intervalul 0-F.

Dezvoltarea se va face in IDE-ul Keil Studio, avand drept target microcontroller-ul specializat FRDM-KL25Z care va incrementa sau decrementa o valoare si o va trimite unui BCD pentru afisare.Modul de incrementare sau de decrementare este ales dintr-o interfață grafica dezvoltată în pyQT5.

# Simulare Circuit:



# Periferice folosite:

## UART:

Uart este un protocol de comunicație care foloseste o comunicație serială asincronă cu rată de transmisie configurabila(Baud rate biti/s , ex. 115200).

Componentele MCU folosesc preponderent UART ca și protocol de comunicație între dispozitive datorită configurării simpliste si folosirii doar a două semnale RX,TX.

Avantajul folosirii comunicației seriale constă in costul redus de implementare,fiind necesar doar un fir pentru comunicația half-duplex si doua fire pentru comunicatia full-duplex.

## BCD:

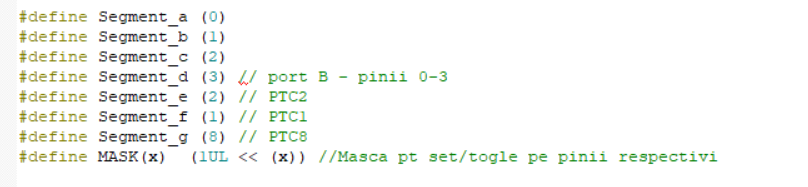
Un display LED cu 7 segmente folosite pentru reprezentarea cifrei sau literei dorite plus inca un segment pentru un punct (acesta nu este folosit in cadrul proiectului).

## Push-Button

# Configurare:

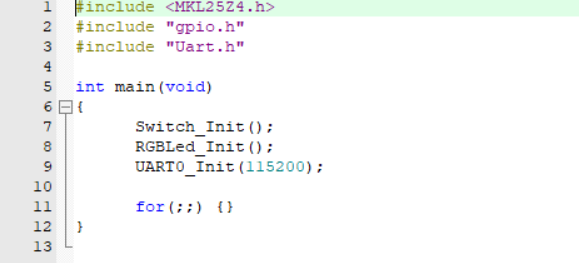


Setarea porturilor specifice perifcericelor alese:





Segmentele a,b,c,d sunt pe portul B,segmentele e,f,g sunt pe portul C iar butonul este pe portul A

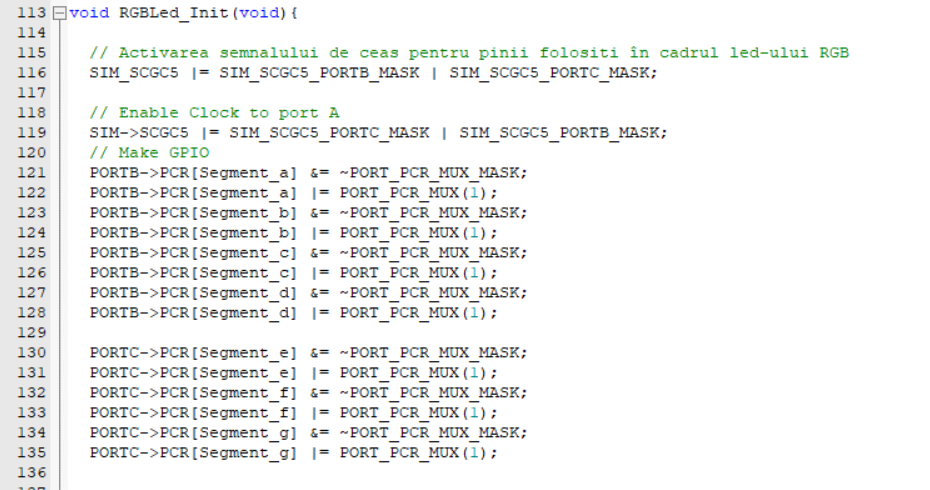


În functia main am inclus fișiere gpio.h si Uart.h în care am definit pinii folosiți, o mască pentru set și toggle pentru pinii respectivi și funcțiile utilizate.

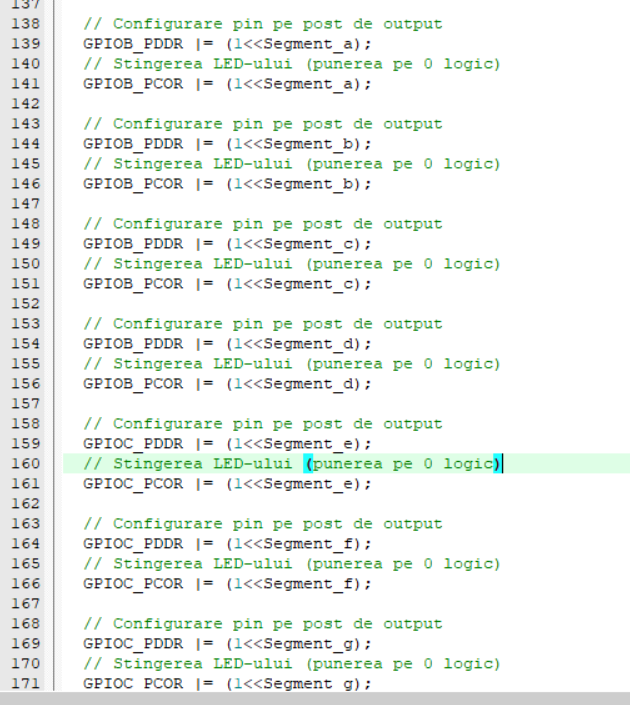
UART0\_Init(115200) initializează modului UART cu un baud rate de 115200.

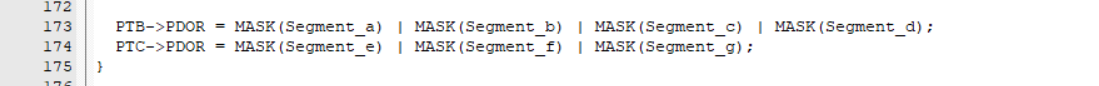
# Inițializarea modulelor

## Inițializare BCD:



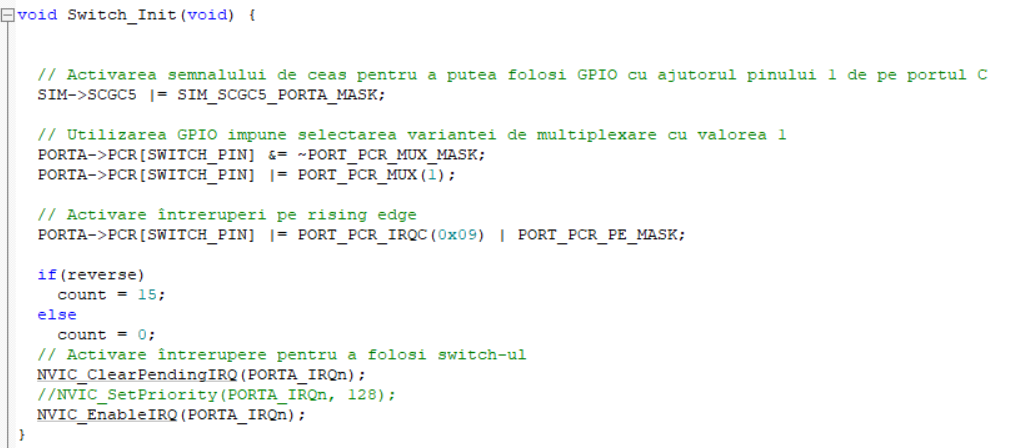
Activăm semnalul de ceas pentru toate porturile folosite și selectăm valoarea de multiplexare.





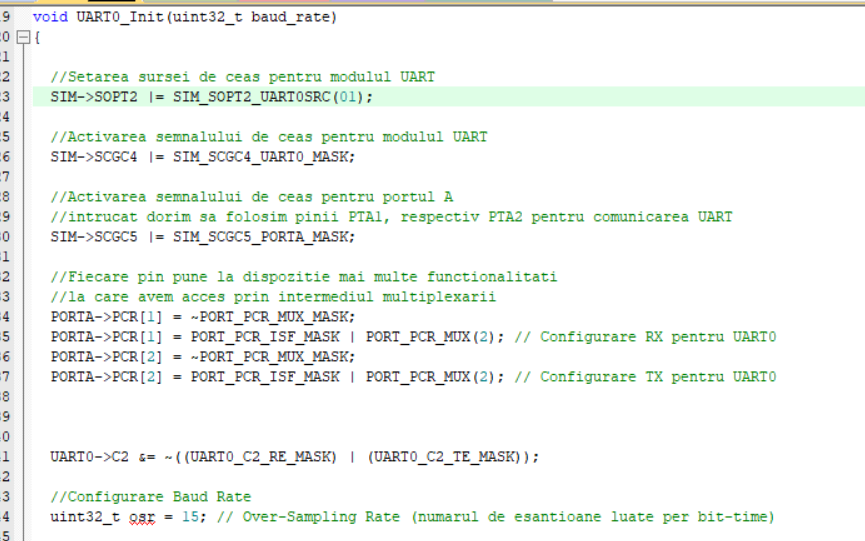
Configurăm fiecare pin pe post de output și stingem LED-urile prin setarea pe 0 logic.

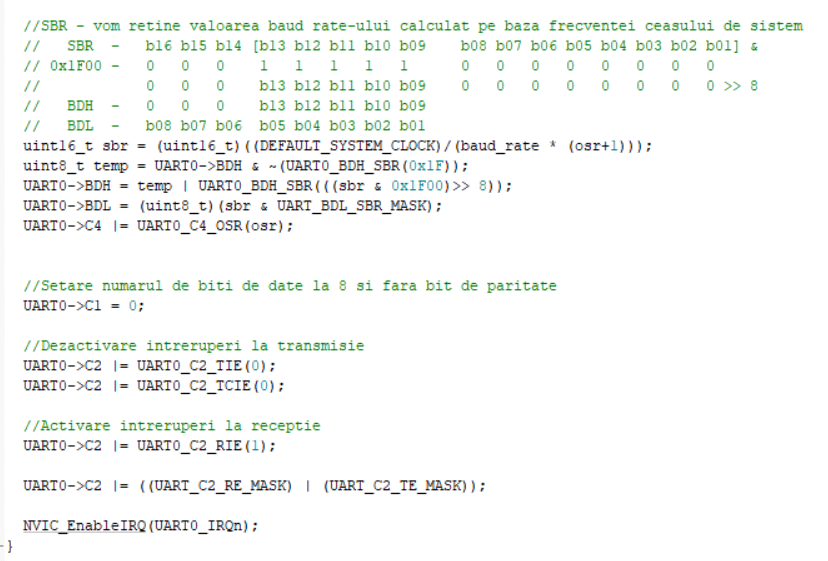
## Inițializare push-button:



If-ul este folosit pentru a putea inițializa counter-ul atat pentru numaul crescator cât și pentru număratul descrescător.

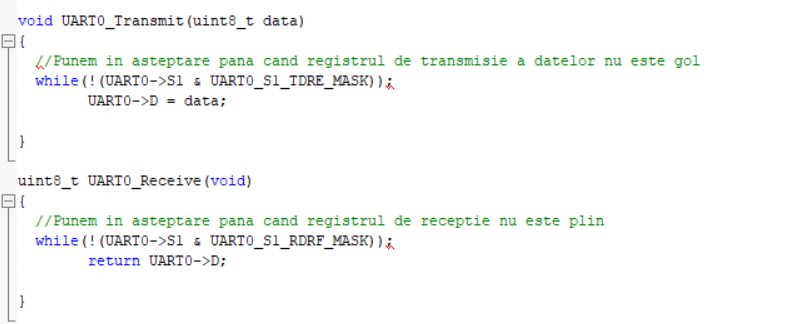
## Inițializare UART:





# Descrierea modului de funționare:

## Transmitere și primire de date prin UART:



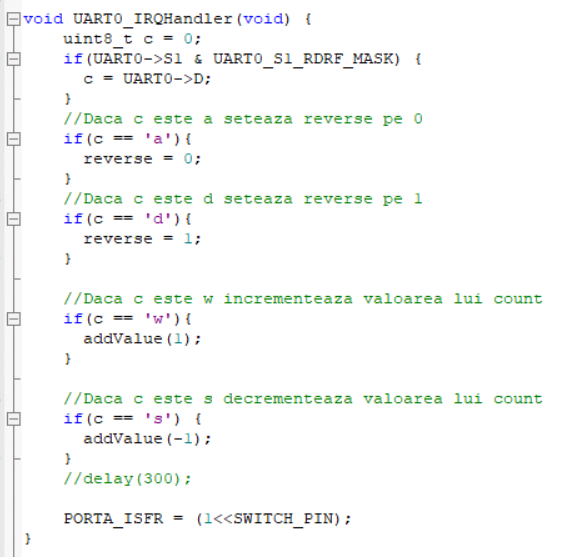
Avem nevoie de aceste 2 funcții pentru a sincroniza count-ul cu interfața grafică.

În UART0\_Transmit 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.

În UART0\_Receive asteptăm pâna cănd câmpul RDRF (Recieve Data Register Full Flag) nu este gol, adică bufferul de primire s-a umplut.

## Utilizarea întreruperilor:

### Întreruperea pe UART:



Folosim această întrerupere pentru comunicarea cu interfața grafică. Astfel, putem realiza schimbarea ordinii de numărare, incrementarea și decrementarea valorii folosind doar interfața.

Am utilizat câteva caractere pentru a putea prelucra corect instrucțiunile primite. Acestea sunt:

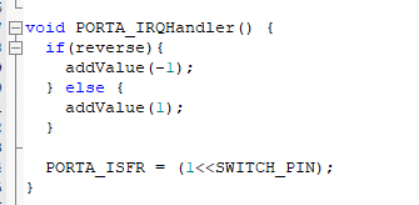
‘a’ – pentru a trece in numărare crescătoare

‘d’-pentru a trece în numărare descrescătoare

‘w’- pentru a incrementa valoarea ce trebuie afișată

‘s’- pentru a decrementa valoarea ce trebuie afișată

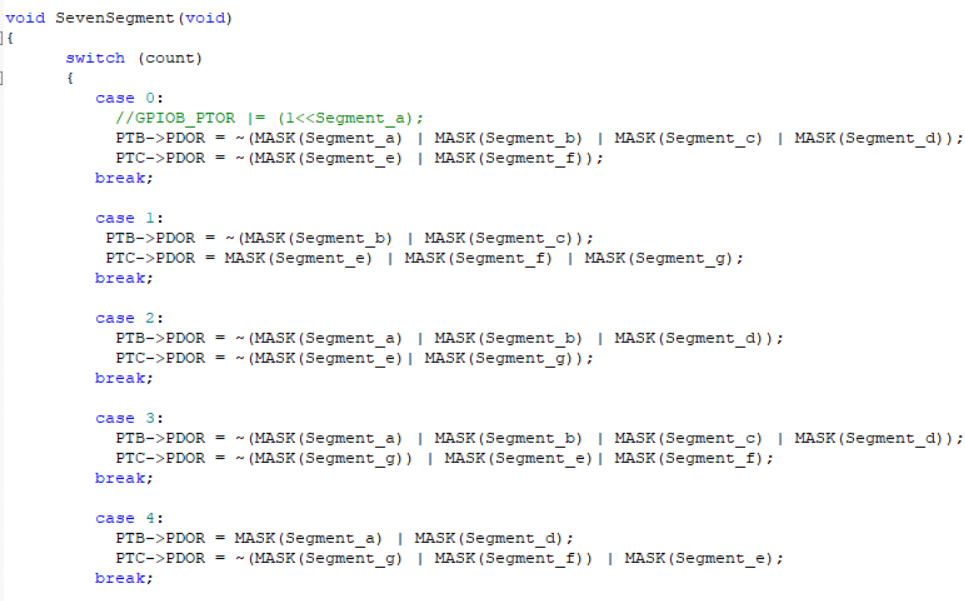
### Întreruperea pe portul A:

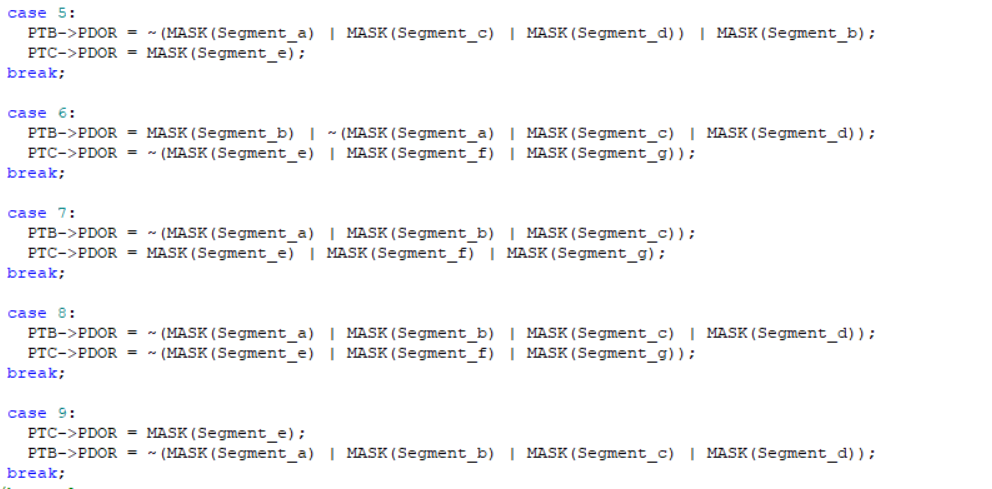


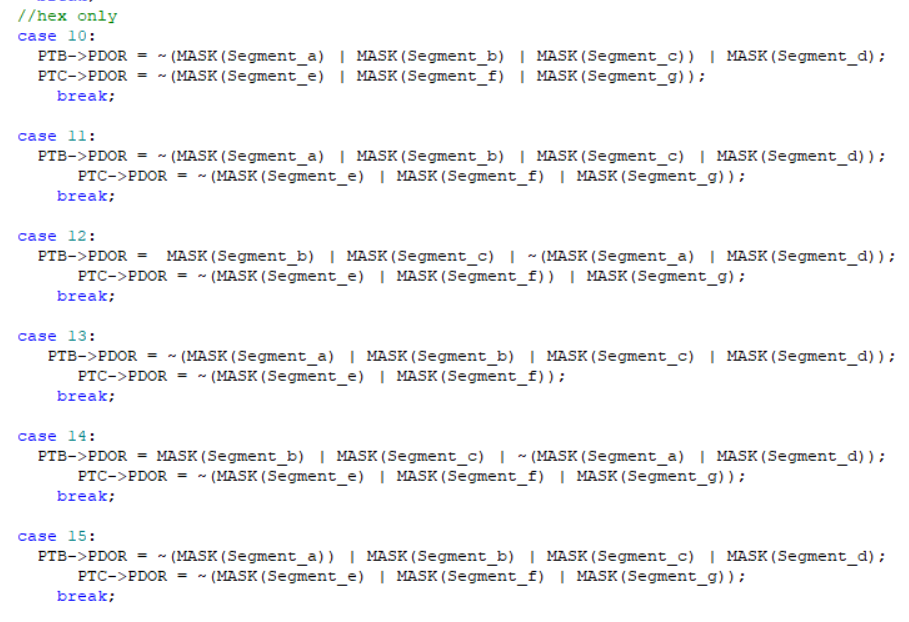
Această întrerupere modifică valoarea count-ului in funcție de modul de numărare in care ne aflăm.

## Modificarea valorii afișate:

Avem un switch\_case pentru fiecare valoare reprezentată și aprindem segmentele relevante.



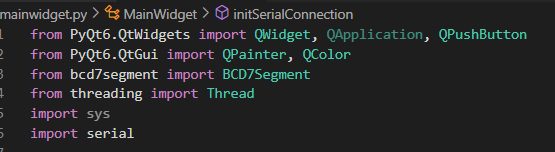




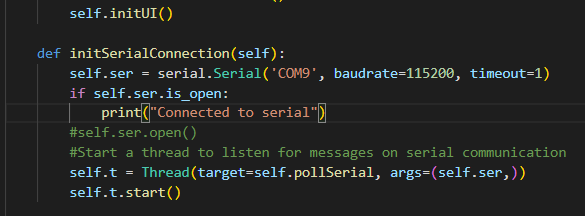
Același lucru și pentru valorile hexazecimale.

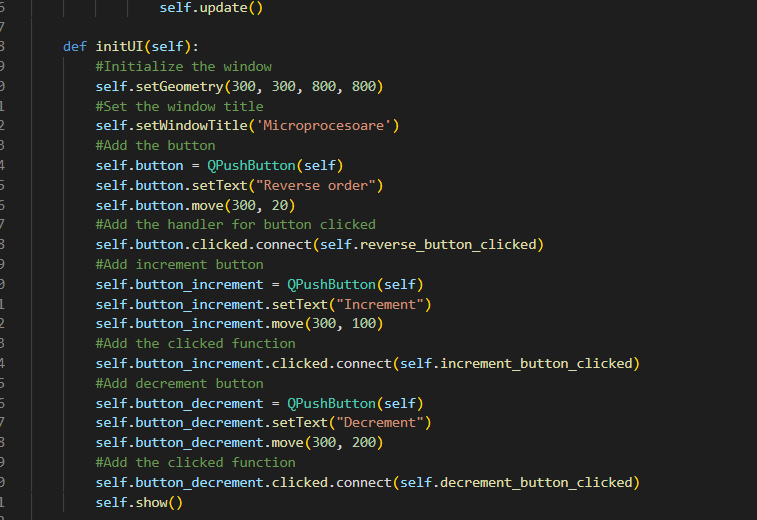
## Interfața Grafică:

Am ales sa folosim PyQT 6 pentru a realiza interfața grafică.

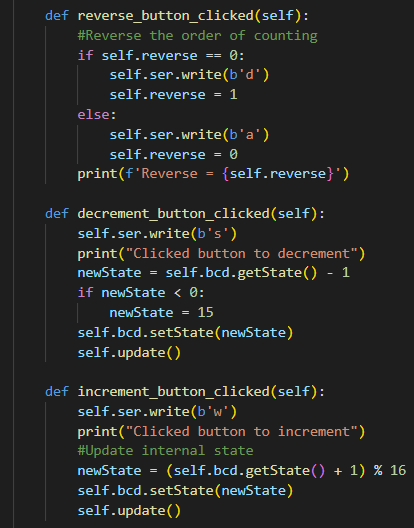


Am folosit si modulele de thread-uri si comunicare serială pentru a realiza legătura cu restul sitemului. Cu aceste 2 module am pornit o comunicare serială pe un port ales și am creat un thread care să aștepte mesaje.

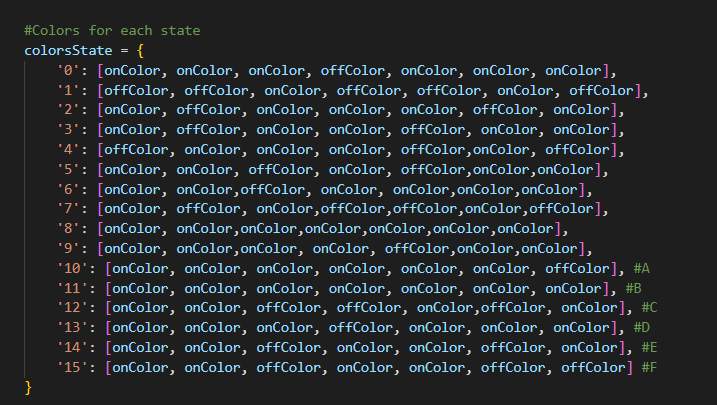
Adăugarea fiecărui elemen in UI:



Funcțiile fiecărui buton :



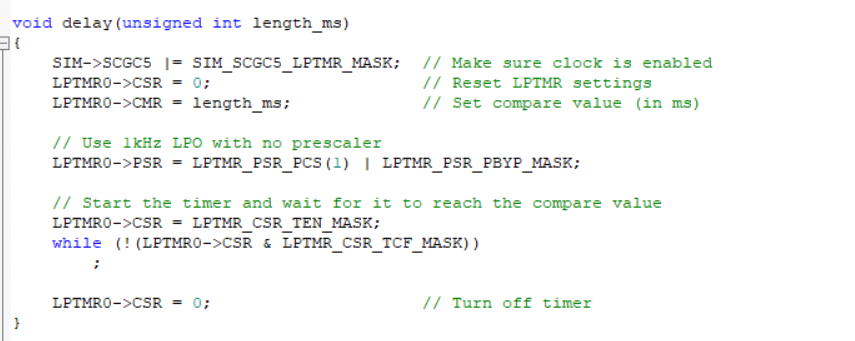
Pentru reprezentarea propriu-zisă am creat un dicționar cu toate stările posibile la BCD-ului, cu fiecare segment aprins sau stins după caz



# Dificultăți întâlnite:

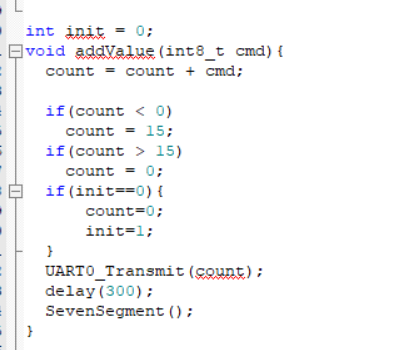
Prima problemă a apărut înca din faza de documentație deoarece nu am gasit aproape nimic legat de BCD. Singurul document care ne-a ajutat este atașat in proiect sub forma unui PDF.

A doua problemă a apărut în momentul în care am implementat modificarea valorii afișate folosind butonul.Din cănd in cănd sarea peste valori si incrementa/decrementa cu 2 sau chiar 3.Am rezolvat această problema adaugănd o funcție de delay



Aceasta previne trimiterile suplimentare de intreruperi la o singură apăsare de buton.

Ultima problemă a apărut in momentul în care trimiteam din interfața grafică comenzi diferite una după alta.Acest lucru am rezolvat cu o standardizare de modificare a valorii folosind o singură funcție:



Init este o variabila careia ii atribuim inițial valoarea 0 pentru a simboliza faza de pornire a proiectului. Astfel, putem începe mereu număratul de la 0 si functia addValue nu influentează punctul de plecare.