**Programarea aplicațiilor de timp real – Proiect**

**Universitatea Tehnică “Gheorghe Asachi” din Iași**

**Facultatea de Automatică și Calculatoare**

Domeniul: Automatică și Informatică Aplicată

An universitar: 2024-2025

Îndrumător: Ș.l.dr.ing. Florin Cătălin Brăescu

Studenți: Neagu Petru-Eduard

Tarța Tudor-Mihai

**Cuprins**

1. **Tema proiectului**
2. **Dezvoltarea proiectului**
3. **Modul de implementare**
4. **Task-ul App : modificarea stării aplicației**
5. **Task-ul de temperatură**
6. **Task-ul pentru tensiune: convertorul analog-numeric**
7. **Task-ul pentru comunicația cu interfața serială**
8. **Task-ul pentru afișarea informațiilor pe LCD**
9. **Task-urile pentru PWM în mod automat/manual**
10. **Alte funcții importante**
11. **Funcția main**
12. **Alte variabile**

**I. Tema proiectului**

Să se proiecteze şi să se realizeze o aplicaţie încorporată bazată pe sistemul de operare

FreeRTOS şi microcontroller-ul dsPIC33FJ128MC802. Aplicaţia va comanda o trapă prin

intermediul unui servomotor, pe baza informaţiei de temperatură achiziţionate prin intermediul

unui senzor de temperatură digital DS18S20/DS18B20.

Aplicaţia va respecta următoarele detalii de implementare:

* va fi pornită respectiv oprită prin intermediul butonului S2 de pe plăcuţă. Starea aplicaţiei va fi indicată de un LED conectat la pinul RB11. LED-ul va fi aprins atunci când aplicaţia este pornită şi va fi aprins intermitent când aplicaţia este oprită.
* temperatura va fi achiziţionată de senzorul DS18S20 sau DS18B20, deschiderea trapei fiind controlată prin servomotor. Temperaturii de 25°C îi va corespunde poziţia centrală a servomotorului. Servomotorul se va deplasa între poziţiile extreme pentru limitele de temperatură (20°C-30°C). Senzorul de temperatură va fi conectat la pinul RB2 al microcontroller-ului.
* aplicaţia va pune la dispoziţie pe lângă modul automat de comandă a servomotorului şi un mod manual de control prin intermediul unei tensiuni externe aplicate pe pinul RB3. Trecerea dintr-un mod de lucru în altul se face prin intermediul meniului de comenzi implementat pe PC. În modul de lucru manual servomotorul va fi poziţionat pe mijloc pentru o tensiune de 2V şi se va deplasa în poziţiile extreme pentru valorile limită de 1V şi 3V. Modul de lucru va fi indicat vizual prin intermediul unui LED conectat la pinul RB1. LED-ul va fi aprins în modul de lucru automat şi stins în modul de lucru manual.
* aplicaţia va permite comunicaţia prin interfaţa serială cu PC-ul. Placa cu dsPIC va recepţiona comenzi sub forma unor caractere şi va transmite drept răspuns şiruri de caractere.
* pe un ecran LCD vor fi afişate informaţii privind temperatura, modul de lucru, tensiunea achiziţionată de la pinul RB3 precum şi ultima comandă primită.
* aplicaţia va pune la dispoziţia utilizatorului un meniu cu următoarele comenzi:
* interogare mod de lucru
* comutare mod de lucru automat/manual (modul de lucru predefinit este cel automat);
* interogare temperatură.

Conexiunile la portul B al microcontroller-ului vor fi făcute conform tabelului de mai jos:

A table with numbers and letters

AI-generated content may be incorrect.

Modul de implementare al proiectului (numărul de task-uri, sarcinile acestora, modul de implementare al accesului la resurse precum şi celelalte obiecte ale sistemului de operare definite) este la alegerea studentului şi va fi prezentat în proiectul prezentat în ultima şedinţă de proiect.

**II. Dezvoltarea proiectului**

Proiectul a fost implementat utilizând mediul de dezvoltare MPLAB IDE.

Aplicația finală rulează pe microcontrolerul dsPIC33FJ128MC802 / dsPIC33FJ64MC802, iar pentru programarea acestuia s-a utilizat dispozitivul PICkit3.

Pentru gestionarea proceselor software, s-a optat pentru utilizarea sistemului de operare FreeRTOS.

Interfațarea cu perifericele din laborator, precum senzorul de temperatură, comunicația serială și afișajul LCD, s-a realizat prin intermediul unor drivere(fișiere .h și .c). Aceste fișiere au denumiri sugestive, cum ar fi: new\_lcd.c, new\_lcd.h, de18s20.c, ds18s20.h, new\_serial.c, new\_serial.h .

A diagram of a computer circuit

AI-generated content may be incorrect.

Schema microcontroller-ului circuitului dsPIC33FJ128MC802 / dsPIC33FJ64MC802

**III. Modul de implementare**

Atunci când ne referim la modul de implementare al proiectului, este vorba despre numărul de task-uri, sarcinile de lucru și accesarea resurselor.

**Descrierea task-urilor**

Proiectul conține în total 7 task-uri, pe care le găsim în fișierul main.c .

Am utilizat, de asemenea, și facilități FreeRTOS, cum ar fi cozi de mesaje.

1. **Task-ul App : modificarea stării aplicației**

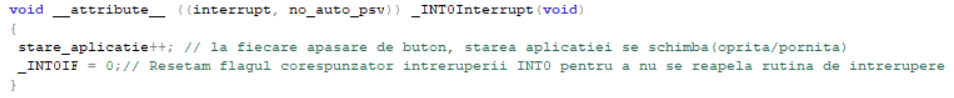
\_INT0Interrupt este apelată la apăsarea butonului S2 de pe plăcuță.

Inițial, aplicația este oprită, deci LED-ul conectat la pinul \_RB11 este aprins intermitent (cu delay de 500 ms); toate task-urile aplicației sunt inițial oprite(cu vTaskSuspend).

După apăsarea butonului S2, LED-ul este aprins și aplicația este pornită(toate task-urile aplicației vor fi funcționale cu ajutorul vTaskResume).

Task-ul acesta, datorită faptului că pornește/oprește aplicația cu ajutorul întreruperii \_INT0 și al butonului S2, va fi task-ul cel mai important.

**Rutina de tratare a întreruperii \_INT0:**



**Inițializarea întreruperii:**

A close-up of a text

AI-generated content may be incorrect.

**Variabila globală, utilă pentru contorizarea stării aplicației:**



**Task-ul:**

A screenshot of a computer program

AI-generated content may be incorrect.

1. **Task-ul de temperatură**

Pentru configurarea temperaturii și afișarea acesteia pe LCD, s-a utilizat senzorul DS18S20, conectat la pinul RB2 al microcontroller-ului(pin setat ca intrare prin \_TRISB2=1).

Pentru a citi temperatura, ne folosim de funcția ds1820\_read().

Pentru a afișa separat pe LCD temperatura, prima dată ne ducem cu cursorul pe linia 1( cu LCD\_line(1) ) și afișăm string-ul “Temp:” (cu LCD\_printf() ).

Cu ajutorul funcției \_itoaQ15, convertim temperatura(cu ajutorul conversiei la int) într-un string, pentru a-l afișa pe LCD (LCD\_printf(sir1) ).

**Task-ul:**

A computer code with black text

AI-generated content may be incorrect.

1. **Task-ul pentru tensiune: convertorul analog-numeric**

Pentru a realiza achizițiile A/D, este necesar să stocăm într-o variabilă externă(în main.c) valoarea din registrul ADC1BUF0.



**Rutina de tratare a întreruperii asociată convertorului analog-numeric**(din adcDrv1.c) are codul de mai jos:

A close-up of text

AI-generated content may be incorrect.

**Task-ul pentru tensiunea externă:**

A computer code with text

AI-generated content may be incorrect.

Pentru a afișa separat pe LCD tensiunea, prima dată ne ducem cu cursorul pe linia 1( cu LCD\_line(1) ) și afișăm string-ul “Tens:” (cu LCD\_printf() ).

Cu ajutorul funcției \_itoaQ15, convertim tensiunea (cu ajutorul conversiei la int) într-un string, pentru a-l afișa pe LCD (LCD\_printf(sir2) ).

1. **Task-ul pentru comunicația cu interfața serială**

Pentru a realiza comunicația serială, este necesar a defini meniul de comenzi (1-interogare mod de lucru, 2-comutare mod de lucru, 3-interogare temperatură).

Dacă apăsăm tasta ‘1’, interogăm modul de lucru, iar LED-ul asociat pinului \_RB1 este stins(mod manual) sau aprins(mod automat), în funcție de mod.

Dacă apăsăm tasta ‘2’, comutăm modul de lucru, cu ajutorul unei variabile,

Dacă apăsăm tasta ‘3’, interogăm temperatura, însă trebuie să fim pe modul automat, iar la fiecare tastă apăsată, cu ajutorul altei variabile, reținem ultima comandă primită.

**Task-ul:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A computer code with text

AI-generated content may be incorrect.**

1. **Task-ul pentru afișarea informațiilor pe LCD**

La acest task, afișăm informații despre mod de lucru(auto/man), tensiune, temperatură și ultima comandă primită pe serială.

A screenshot of a computer program

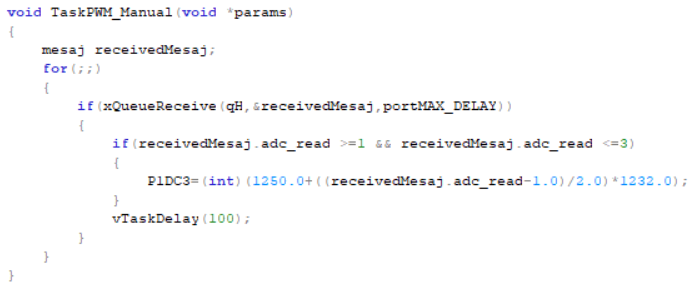
AI-generated content may be incorrect.

1. **Task-urile pentru PWM în mod automat/manual**

Pentru modul manual, cel în care avem tensiunea, trebuie să fie în

intervalul [ 1, 3 ] V. Pentru aceasta, verificăm într-o coadă de mesaje ce am primit în variabila adc\_read și facem câteva calcule pentru a pune valoarea finală în registrul PWM P1DC3.

**Task-ul pentru PWM Manual:**



Pentru modul automat, verificăm temperatura să fie în intervalul [20. 30] de grade și vedem ce este recepționat din coada de mesaje, pentru a calcula și a pune în același registru P1DC3.

**Task-ul pentru PWM Automat:**

**A screenshot of a computer code

AI-generated content may be incorrect.**

1. **Alte funcții importante**

**Funcția initPWM1:**

A white background with black text

AI-generated content may be incorrect.

**Funcția initializare\_temperatura:**

**A white background with black text

AI-generated content may be incorrect.**

**Funcția prvSetupHardware:**

În această funcție, principalele chestiuni pe care le-am făcut sunt: am inițializat LCD-ul, temperatura, ADC-ul, Timer-ul 3, am inițializat întreruperea \_INT0, am făcut \_RB1 și \_RB10 ieșiri, iar \_RB2 și \_RB3 intrări și am inițializat modului PWM, precum și interfața URAT1.

A screenshot of a computer code

AI-generated content may be incorrect.

1. **Funcția main:**

În main, am apelat prvSetupHardware(), și am creat task-urile pentru LCD, interfața serială, stare aplicație, temperatură, tensiune, modul automat și cel manual al pwm-ului. De asemenea, am creat o coadă de mesaje pentru funcționalități ca temperatura și tensiunea externă.

Task-ul cu starea aplicației este cel mai important, deoarece pornește/oprește toată aplicația, task-urile Temp și Tens sunt cele cu temperatura, respectiv achizițiile A/D.

Task-urile cu LCD-ul și interfața serială sunt de asemenea importante, deoarece pe LCD afișăm informațiile finale, iar pe serială comunicăm cu PC-ul ce mod de lucru vom avea, împreună cu alte informații(temperatură, tensiune, ultima comandă).

A close-up of a computer code

AI-generated content may be incorrect.

1. **Alte variabile**

Principalele date folosite, printre care un struct și un define, sunt cele de mai jos:

A screen shot of a computer code

AI-generated content may be incorrect.