**Studenți: Grupa: 1.2**

**Cebuc Mihai-Gheorghe**

**Cojocaru Cristian-Marian**

**PROIECT ORGANIZAREA CALCULATOARELOR**

1. TABEL COMPONENTE FOLOSITE

|  |  |  |
| --- | --- | --- |
| **Name** | **Quantity** | **Component** |
| Potențiometru acționare motor | 1 | 250 kΩ Potentiometer |
| Ușa stânga față Ușa stânga spate Ușa dreapta spate Ușa dreapta față Fotorezistența faruri | 5 | Photoresistor |
| Stopul drept Stopul stâng | 2 | Red LED |
| Farul stâng Farul drept | 2 | White LED |
| Buton frână | 1 | Pushbutton |
| Rezistența buton frână Rezistența stanga față Rezistența dreapta spate Rezistența dreapta față Rezistența stanga spate Rezistența foto faruri | 6 | 4.7 kΩ Resistor |
| Rezistența stopuri Rezistența faruri | 2 | 220 Ω Resistor |
| Placuța Arduino | 1 | Arduino Uno R3 |
| Ecran LCD | 1 | MCP23008-based, 32 LCD 16 x 2 (I2C) |

1. ARHITECTURA HARDWARE

Diagram

Description automatically generated

1. CONECTARE PINI ARDUINO

***Potențiometrul acționare motor*** (rezistență internă 250 KΩ) conectat la pin A0 Arduino, la alimentare (5V) și GND.

***Fotorezistență faruri*** conectată la pin A3 Arduino, la alimentare (5V) și legată în serie cu o rezistență de 4,7 KΩ care merge la GND.

***Fotorezistențe uși față*** conectate la pin A1 Arduino, la alimentare (5V) și legate în serie fiecare cu o rezistență de 4,7 KΩ care merge la GND.

***Fotorezistențe uși spate*** conectate la pin A2 Arduino, la alimentare (5V) și legate în serie fiecare cu o rezistență de 4,7 KΩ care merge la GND.

***Led-uri Stopuri*** legate în serie și conectate la GND, respectiv în serie cu o rezistență de 220 Ω care se conectează la pinul 3 Arduino.

***Led-uri Faruri***legate în serie și conectate la GND, respectiv în serie cu o rezistență de 220 Ω care se conectează la pinul 4 Arduino.

***Buton frână***conectat la pinul 2 Arduino, la GND și la alimentare (5V), legat în serie cu o rezistență de 4,7 KΩ.

***Ecran LCD***conectat la alimentare (5V), GND și pinii SCL, SDA la Arduino.

1. TESTARE COMPONENTE

**Valoarea maximă** este 1023 echivalentul pentru 5V.

**Valoarea minimă** este 0 echivalentul pentru 0V.

Pentru ***potențiometrul acționare motor*** avem:

* Între 0 și 30% din valoarea maximă: MOD OPRIT
* Între 31% și 60% din valoarea maximă: MOD ACCESORII
* Peste 61% din valoarea maximă: MOD PORNIT

Pentru ***fotorezistență faruri***:

* Valabil doar în MOD ACCESORII sau MOD PORNIT
  + Între 0 și 50% din valoarea maximă: FARURI PORNITE
  + Peste 50% din valoarea maximă: FARURI OPRITE

Pentru ***fotorezistențe uși***:

* Între 0 și 50% din valoarea maximă: UȘI INCHISE
* Peste 50% din valoarea maximă: UȘI DESCHISE

OBS: Ușile față sunt legate ambele la un singur pin Arduino, iar ușile spate sunt legate ambele la alt pin Arduino, astfel încât pentru valoarea orcărei fotorezistențe din cele 2 mai mare de 50% este semnalată “USA DESCHISA”.

Pentru ***buton frână***:

* Valabil doar în MOD ACCESORII sau MOD PORNIT
  + Când butonul este apăsat se aprind stopurile, iar pe ecranul LCD apare simbolul “#”
  + Când butonul nu este apăsat pe ecranul LCD apare simbolul “.”

1. ARHITECTURA SOFTWARE

5.1 BIBLIOTECI

* + Am inclus biblioteca *<Adafruit\_LiquidCrystal.h>* pentru a folosi ecranul LCD

5.2 DEFINIRE DATE

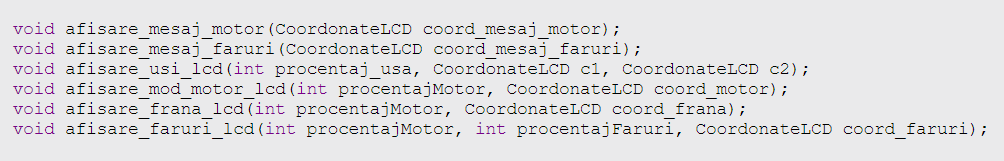
* + moduri de funcționare motor {oprit, accesorii, pornit}
  + mod faruri {oprite, pornite}
  + mod ușă {deschisă, închisă}
  + afisaj pentru uși, mod motor, frână, faruri
  + definire structură coordonate ecran LCD

Text

Description automatically generated

5.3 DEFINIRE FUNCȚII

* + Funcții pentru afișarea funcționalitățiilor mașinii pe ecranul LCD:



* + Funcția *setup* asigură legatura dintre pini și senzori:

Text, letter

Description automatically generated

* + Funcția *loop* asigura sincronizarea datelor afișate în funcție de modul în care se acționează asupra arhitecturii hardware

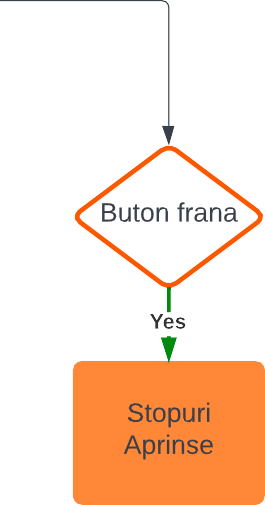
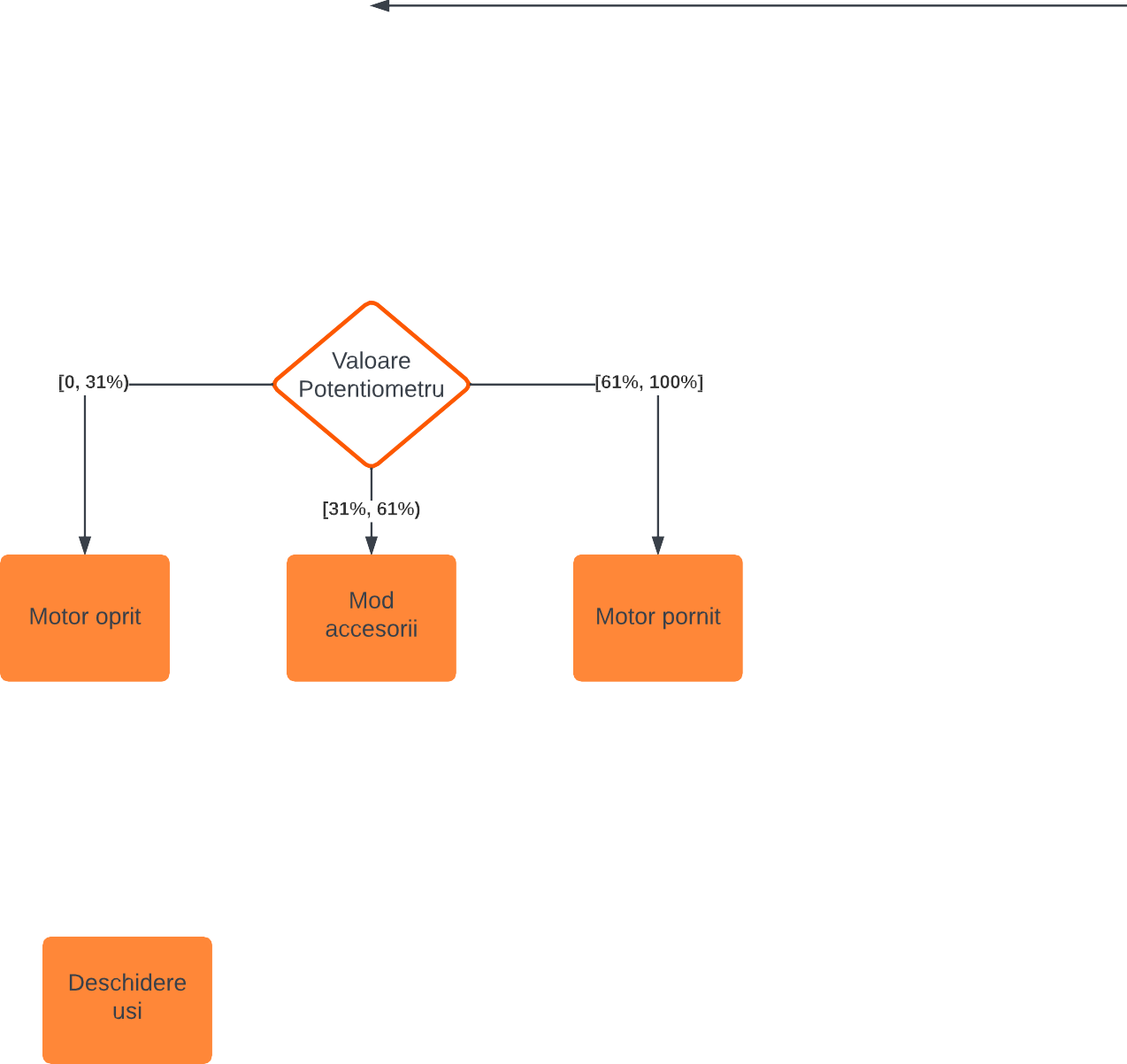
5.4 CITIRE DATE

A picture containing text

Description automatically generated

* + Pe baza acestor date se modifică afișajul LCD, prin apelarea repetată a funcțiilor de afișare folosind funcția *loop*

1. SCHEMA LOGICĂ



1. PSEUDOCOD

Cât timp placuța este pornită codul următor rulează în loop infinit.

*functie* usi (data\_usi\_fata, data\_usi\_spate) *begin*

*daca* data\_usi\_fata <= 50% *atunci*

afiseaza usi\_fata\_deschise;

*altfel*

afiseaza usi\_fata\_inchise;

daca data\_usi\_spate <= 50% *atunci*

afiseaza usi\_spate \_deschise;

*altfel*

afiseaza usi\_spate \_inchise;

end

*functie* faruri (data\_faruri) *begin*

*daca* data\_faruri < 50% atunci

afiseaza faruri\_aprinse;

*altfel*

afiseaza faruri\_stinse;

*end*

*functie* actioneaza\_buton\_frana() *begin*

*daca* buton == apasat atunci

afiseaza frana\_actionata;

aprinde stopuri;

*altfel*

afiseaza frana\_neactionata;

*end*

*citeste* data\_faruri, data\_usi\_fata, data\_usi\_spate , valoare\_potentiometru;

*daca* valoare\_potentiometru <= 30% *atunci begin*

motor <- oprit;

usi(data\_usi\_fata, data\_usi\_spate);

end *altfel* *daca* valoare\_potentiometru <= 60% *atunci begin*

motor <- accesorii;

usi (data\_usi\_fata, data\_usi\_spate);

faruri (data\_faruri);

actioneaza\_buton\_frana();

*end altfel begin*

motor <- pornit;

usi (data\_usi\_fata, data\_usi\_spate);

faruri (data\_faruri);

actioneaza\_buton\_frana();

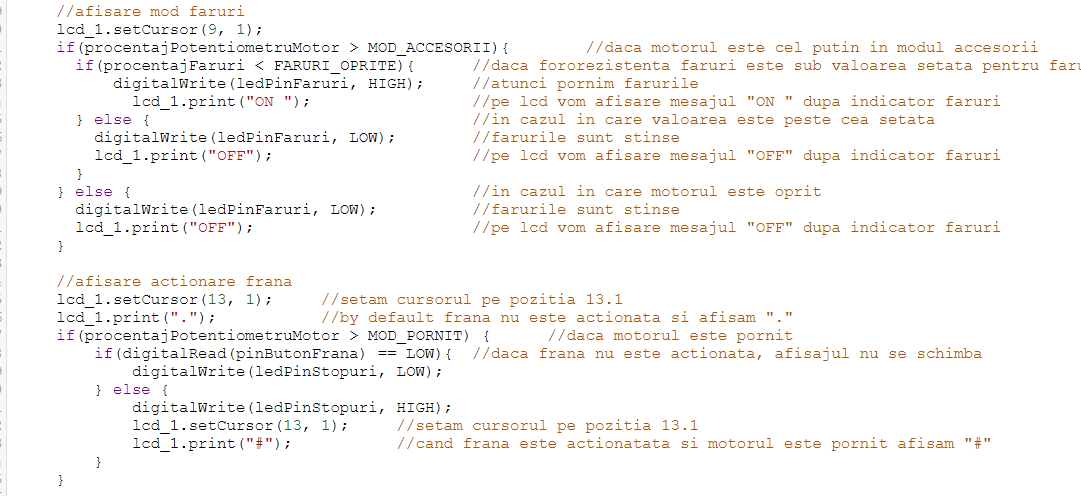
*end*

1. CONCLUZII

* Înainte de optimizare, codul nu era modularizat, aveam valori hardcode-ate (magic numbers):

Text, letter

Description automatically generated with medium confidence



* După optimizare, folosind define-uri (no magic numbers) și funcții:

Text

Description automatically generated

OBS: Cea mai grea parte a fost corelarea arhitecturii hardware cu funcționalitatea implementată prin cod.