**Istoric Proiect - Cântar**

1. Verificarea funcționării amplificatorului prin măsurarea rezistenței la pini cu un multimetru.
2. Lipirea cu cositor a firelor dezlipite de la amplificator.
3. Testarea codului de calibrare pe celula de 150kg. Oscilațiile mari la rezultate ne-au determinat să căutăm alte exemple și alte biblioteci.
4. Testarea codului de calibrare pe 4 celule de 50kg. Rezultatele au fost mult mai satisfăcătoare, dar am constatat că erorile provin și din cauza plăcii de lemn, care nu era o suprafață plană.
5. Realizarea unei noi plăci pentru cântar din plexiglas, găurirea plăcii și fixarea cu șuruburi a celulelor pe noua placă.
6. Retestarea codului pe noua placă și obținerea unor rezultate mult mai bune. Calibrarea cu o greutate mai mare reduce erorile de măsurare.
7. Adaptarea codului pentru a-l face prietenos cu utilizatorul. Am realizat un meniu care se afișează după pornirea Serial Monitor.

A screenshot of a computer program

Description automatically generated

Fig.1 – Funcția care afișează meniul

* 1. Funcția ***setup()*** conține setarea ratei de transmisie serială a datelor (am folosit 57600 baud), inițializarea comunicației cu amplificatorul, stabilizarea celulei de greutate și verificarea posibilelor erori.
  2. Funcția ***loop()*** afișează meniul apelând funcția ***displayMenu()*** și permite introducerea de la tastatură, în Serial Monitor, a opțiunii pe care utilizatorul dorește să o aleagă. Astfel, avem trei cazuri de utilizare:
     1. *Calibrare* – accesat prin trimiterea cifrei 1
     2. *Măsurare* – accesat prin trimiterea cifrei 2
     3. *Exit* – accesat prin trimiterea cifrei 3

Pentru a evita afișarea repetată a meniului, am folosit un *if statement* pentru a verifica dacă meniul este deja afișat, astfel că pentru reținerea statusului am declarat o variabilă booleană ***menuDisplayed***. Dacă această variabilă are valoarea *false*, atunci putem afișa meniul (nu l-am mai afișat până atunci), iar apoi e setată pe *true* după apelarea ***displayMenu()*** o dată. După alegerea unei opțiuni, flag-ul este resetat pe *false* pentru a da posibilitatea alegerii unei noi opțiuni.

A black text on a white background

Description automatically generated

Fig.2 – if statement referit în paragraful de mai sus

De asemenea, am mai folosit o funcție, ***clearSerialBuffer()***, care să garanteze posibilitatea alegerii unei noi opțiuni odată ce am executat unul din cazurile meniului. Din cauza unor caractere rămase în buffer-ul de la Serial Monitor, codul sărea peste partea de user input.

Fiecare caz apelează o funcție care calibrează, măsoară sau închide programul.

A screenshot of a computer program

Description automatically generated

Fig.3 – Cazurile meniului și apelurile de funcții

* 1. Funcția ***calibrate()*** setează întâi zeroul cântarului, iar apoi cere utilizatorului să plaseze un obiect pe cântar și să introducă greutatea sa cunoscută. Pe baza acestora, e calculat factorul de calibrare, care poate fi apoi salvat sau nu în EEPROM.
  2. Funcția ***measureMode()*** verifică dacă sunt disponibile noi date de la celula de greutate și, în cazul în care sunt, afișează valoarea de măsurare returnată de funcția ***getData()***.
  3. Cazul de *Exit* presupune intrarea într-o buclă infinită, fiind doar o simulare a cum ai putea opri Serial Monitor.
  4. Pentru toate funcțiile din switch, există posibilitatea ieșirii din execuția codului aferent cazului prin introducerea tastei ***m***.

1. Realizarea prezentării, ilustrând problemele pe care le-am întâmpinat, soluțiile cu care am venit și bibliografia aferentă muncii noastre.

Link Prezentare Proiect

<https://view.genially.com/672b5611f5e593d2a8b60c88/interactive-content-proiect-br>