

Self balancing robot

Introducere

Nume: Mihaila Octavian-Iulian

Grupa: 333CB

Indrumator: Andrei Zamfir

Descriere generală

Proiectul consta in dezvoltarea unui robot auto-echilibrant care pentru a ramane in pozitie verticala pe doua roti se foloseste de modulul cu giroscop si accelerometru si de cele doua motoare stepper manevrate de catre microcontroller.

Scopul este de a înțelege și implementa algoritmi de control pentru sisteme dinamice(PID) și de a explora aplicabilități în domenii precum navigația și stabilizarea dronelor, precum și controlul orientării sateliților.

Detalii de implementare:

- Cele doua motoare stepper de care dispune robotul sunt controlate precis de drivere A4988 care folosesc microstepping. Corectarea poziției și echilibrul sunt monitorizate continuu de un modul giroscop-accelerometru (MPU6050), care furnizează date despre înclinarea și orientarea robotului. Aceste date sunt folosite de algoritmul de control ce ajustează activitatea motoarelor pentru a menține robotul stabil.
- Alimentarea pentru motoare si placa arduino este asigurată de un acumulator 3S 11.1V, dar intr-un mod individual. Separarea alimentării pentru motoare și placa de dezvoltare facilitează procesul de lucru și depanare, permițând utilizarea unui releu pentru a tăia alimentarea motoarelor în cazul unor situații neașteptate, păstrând în același timp funcționalitatea sistemului de control și comunicare.

Schema bloc:



Hardware Design

• Lista de componente:

Motor Stepper 17HS8401S x2
Driver pentru Motor Stepper A4988 x2
Modul Accelerometru și Giroscop cu 3 Axe MPU6050
Pereche Roți Roșii Pololu 70×8mm (25T, 5.8mm)
Modul releu cu un canal
Acumulator 3S 11.1V lipo
Conector XT60

Schema electrica:



Software Design

Mediu de dezvoltare folosit:

- PlatformIO

Interactiune MPU-6050:

- Comunicare folosind implementarea I2C din cadrul laboratorului.
- Sensibilitatea accelerometrului setata la $\pm 8g$.
- Sensibilitatea giroscopului la ± 500 grade pe secundă (dps).

Calculul unghiului compus din datele colectate de la giroscop si accelerometru:

- Se calibreaza giroscopul si accelerometrul.
- La fiecare iteratie citim valorile furnizate de senzorul MPU-6050 si ajustam unghiul in felul urmator:
- Se aduna unghiul parcurs de giroscop pe axa x.
- Acelasi unghi este ajustat in functie de acceleratia totala(magnitudine).
- Corectarea drift-ului specific pentru senzor MPU-6050.
- Folosirea unui filtru complementar pentru a elimina vibratiile date de schimbarile bruste de unghi.
- Tehnica inspirata din modul in care se calculeaza unghiul pentru un controller de zbor(vezi bibliografie).

Implementare PID controller:

- Algoritmul PID primeste ca input unghiul compus (eroare curenta) si pe cel anterior (eroarea precedenta).
- $pid_result = PID_KP * curr_pid_error + PID_KI * (curr_pid_error + prev_pid_error) + PID_KD * (curr_pid_error - prev_pid_error)$
- Limitam rezultatul maxim la un interval maxim $\pm MAX_PID_RESULT$ care poate fi ajustat in functie de rezultatul dorit.
- Pentru a face tuning, am variat cele trei constante in felul urmator:
 - PID_KP, pentru a controla oscilatiile (overshoot).
 - PID_KI, eroare acumulata de-a lungul timpului, folosit pentru a corecta offset-ul acumulat de-a

lungul timpului.

- PID_KD, proportional cu rata de schimbare a erorii, imbunatatirea stabilitatii cand robotul se afla in punctul de echilibru (sta pe loc), ajuta la eliminarea vibratiilor.

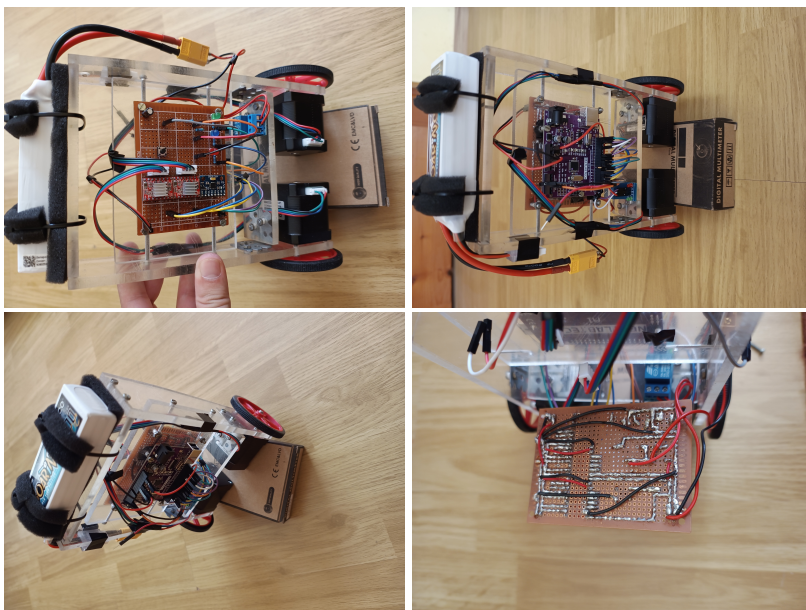
Controlul motoarelor stepper:

- Am scris o biblioteca proprie care se ocupa de acest aspect, utilizand direct registrii pentru a obtine cele mai bune performante.
- Pulsurile sunt generate in functie de rezultatul algoritmului PID descris mai sus.

Battery management:

- Cu ajutorul unui divizor de tensiune compus din 2 rezistente ($R1 = 10\text{Kohm}$, $R2 = 4.7\text{Kohm}$) am redus tensiunea sub 5V astfel incat sa poata fi citita pe un pin analogic.
- Ulterior folosind formula divizorului de tensiune am reprodus nivelul tensiunii din acumulator.
- In momentul in care tensiunea scade sub 11.5V, led-ul rosu de pe robot este aprins.

Rezultate Obținute



Concluzii

Un proiect provocator, atat din punct de vedere al constructiei cat mai ales software.

Ce a mers bine ?

- Constructia si design-ul. Foarte bine echilibrat.
- Placuta realizata pe un pcb de prototipare a iesit chiar frumos si functioneaza fara probleme.

Ce a mers mai putin bine ?

- Destul de greu de tinut un unghi "live" in software. La inclinari rapide dintr-o parte in alta a robotului.
- Problema de mai sus duce la imposibilitatea ca motoarele sa reactioneze rapid la inclinari bruste.

Download

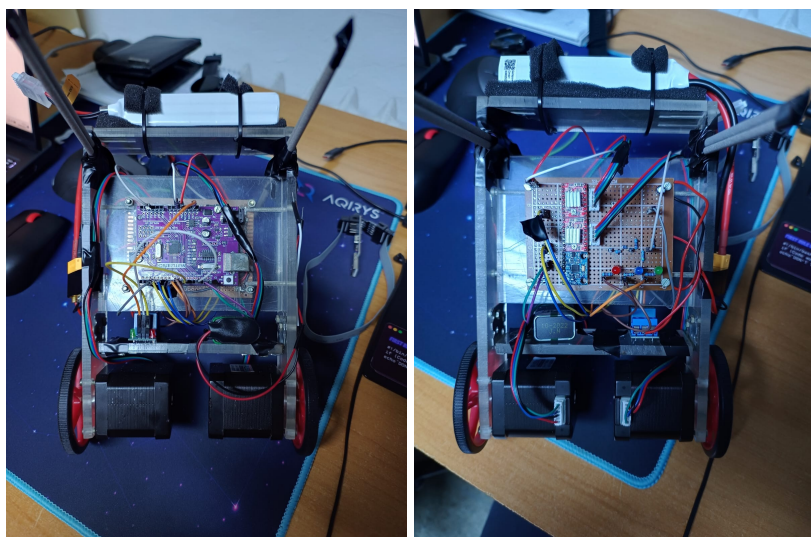
[self_balancing_robot.zip](#)

Jurnal

Stadiu 27.04:



Stadiu 16.05:



Stadiu 21.05:

<https://youtu.be/HcfEya9rgvo>

Bibliografie/Resurse

Datasheets:

- [MPU-6050 Register Map](#)
- [A4988 stepper driver](#)
- [Atmega328P](#)

Resurse Hardware:

- [Setare limita current drivere](#)

Resurse software:

- [Verificare tensiune din baterie](#)
- [Control motoare stepper](#)
- [Calcul unghi giroscop si accelerometru](#)
- [PID](#)

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/azamfir/octavian.mihaila02>



Last update: **2024/05/26 13:46**