Self Balanced Robot

Introducere

Un robotel cat de cat inalt, cu doua roti, ce se balanseaza cat sa ramana in picioare mergand in fata sau in spate, opus directiei in care teoretic ar cadea la un moment de timp. Astfel, in aproape continua miscare, reuseste sa ramana in picioare, in ciuda centrului inalt de greutate.

Descriere generală

Un robotel ce foloseste un giroscop pentru a intelege in ce directie cade (fata sau spate), intr-un timp destul de scurt cat sa poata actione motoarele pentru a invarti rotile, astfel compensand caderea inclinandu-se in partea opusa necesar cat sa anuleze caderea. Aceasta functie se repeta incontinuu, stand foarte aproape de centrul de balansare tot timpul, rezistand inclusiv la impingeri din partea unei persoane sau alti factori disruptivi decat doar gravitatia.

Schema logica

Giroscop —*informatii*—> **Microprocesor** —*voltaj*—> **Motoare** —*invartire*—> **Roti**

Schema Eagle

Descriere software

Foloseste giroscopul pentru a prelua nivelul de inclinare. Porneste motoarele si invarte rotile in directia opusa (aceeasi directie cu caderea) pana se ajunge iar intr-o pozitie verticala. Apoi se permite iar caderea intr-o directie si repornirea rotilor, facand aceste comenzi la infinit.

Descriere hardware

Un robotel ce se sprijina de pamant doar folosind 2 roti, avand o structura cat de cat inalta, verticala. Rotile sunt conectate la motoare, alimentate de baterii. Motoarele sunt declansate de catre placuta Arduino, ce contine software-ul prezentat mai sus. Giroscopul transmite informatii despre unghiul de inclinare incontinuu microprocesorului, pentru a determina in ce fel sa activeze motoarele.

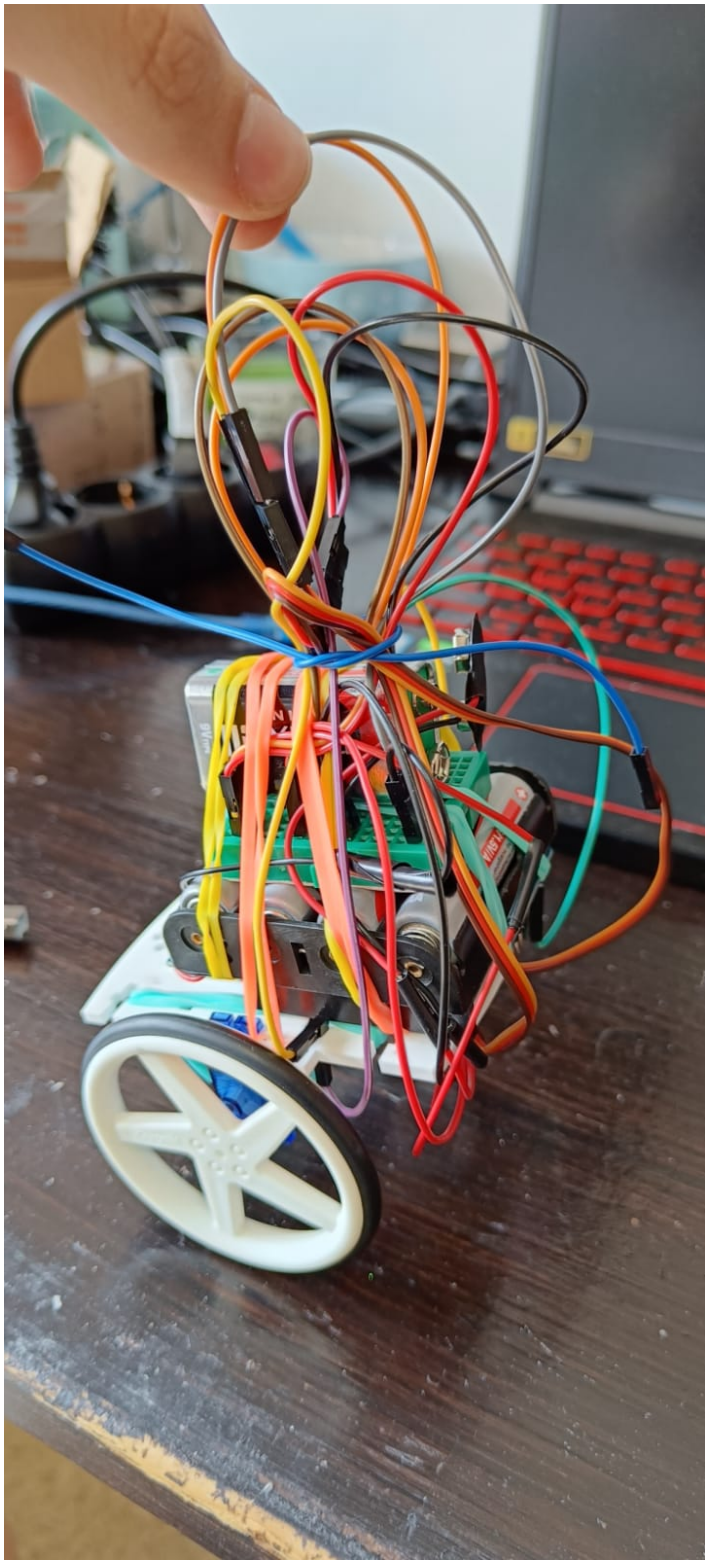
Hardware Design

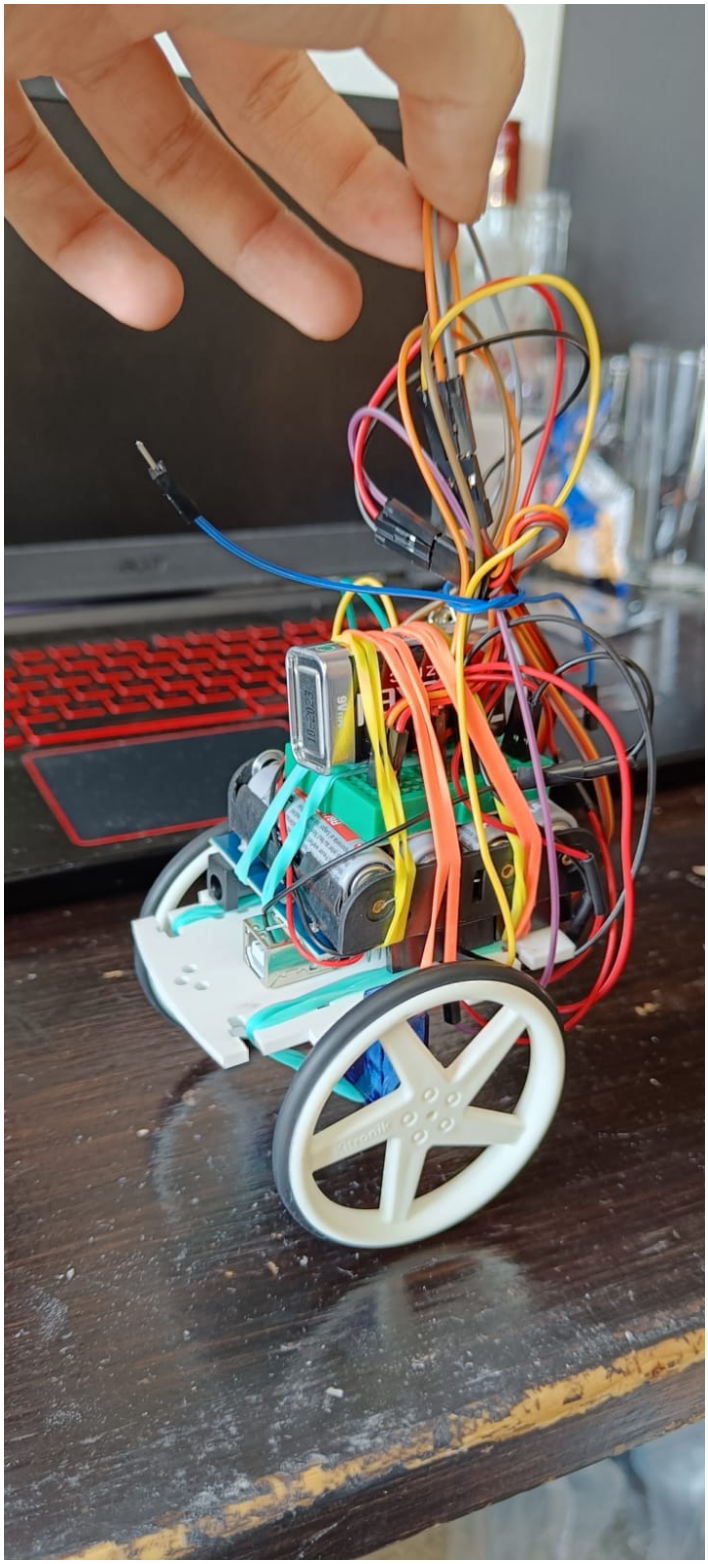
Componente:

- Placa Arduino → Pentru partea software a robotului
- Breadboard → Pentru a conecta componentele intre ele
- 4 Baterii AA + Suport de baterii → Pentru a alimenta motoarele
- Baterie 9V → Pentru a alimenta microprocesorul Arduino
- 2 servomotoare continue → Pentru a controla miscarea rotilor
- 2 roti → Pentru a permite robotului sa se miste, astfel balansandu-se singur
- Modul Giroscop analog → Pentru a determina inclinarea robotului
- Cabluri, elastice, suporti din plastic → Pentru suport fizic

Observatie: Din doua servomotoare ce blocau miscarea rotilor la maxim 90 de grade in fiecare parte, am desfacut motoarele, am taiat accesul la cate un potentiometru intern si am taiat un cub ce bloca invartirea rotilor la mai mult de 90 de grade pentru a obtine 2 servomotoare continue.

Robotul final:





Software Design

- libraria <Servo.h> - pentru controlul servomotoarelor
- librariile “Wire.h” si “I2C.h”- pentru a comunica cu componente I2c (giroscopul) - |Sursa librarii
- |libraria "PID_v1.h" - pentru calculul complex si fine-tuned al vitezelor servomotoarelor

Tot algoritmul este format in mare din 3 mari componente:

1. Giroscopul:

Folosind functii specifice I2C, se preiau din senzor datele giroscopului. Avem acces si la un accelerometru, barometru si senzor de temperatura prin modulul ales, dar ne intereseaza doar giroscopul. In special, pe directia Ox, intrucat avem 2 roti si ne intereseaza doar miscarea fata-spate.

2. Servomotoarele:

Cea mai simpla parte, atasam pinii 11 si 12 la cele 2 motoare si atasam unghiuri suplementare celor 2 servomotoare pentru o rotatie continua in aceeasi directie a robotului.

3. PID

Se foloseste algoritmul PID (Proportional-Integral-Derivative) pentru a determina viteza servomotoarelor in functie de datele de la giroscop. Se primeste un input (in acest caz, unghiul dat de giroscop), un ideal (unghiul de echilibru al robotului) si rezulta un output din algortimul de PID control, folosind 3 valori ce trebuie tunate unic pentru fiecare robot:

Proportional: se calculeaza un output de viteza in functie de eroarea curenta fata de ideal.

Integral: In functie de magnitudiea si durata erorii, se calculeaza acceleratia.

Derivative: In functie de rata de schimbare a erorii se mai influenteaza viteza motoarelor.

O foarte mare parte a timpului a fost consumat facand fine-tuning pe valorile acestui sistem. Acesta foloseste aceste 3 variabile de mai sus (Kp, Ki, Kd), care depind foarte mult de specificatii gen greutatea robotului, diametrul rotilor etc, care se pot afla doar prin trial and error pentru ca robotul sa se balanseze bine.

Rezultate Obţinute

Un robot nu foarte precis. Acesta are nevoie de sprijin din partea userului, din cauza unor caracteristici atat software cat si hardware: - motoare nu destul de puternice - neacuratati hardware din modul cum a fost construit, atat pentru directia imperfecta a motoarelor, cat si a centrului de greutate pozitionat mult in fata robotului - calibrare a constantelor pentru PID imprecise

Videouri exemplu

Concluzii

Personal, a fost o provocare sa construiesc acest robot cu resurse minime cat sa fie stabil si cat de cat ok.

Insa, obiectiv si de departe, partea cea mai dificila dar si cea mai importanta a acestui proiect este calibrarea.

Cele trei constante, **Kp** = Proportional gain, **Ki** = Integral gain si **Kd** = Derivative gain, sunt foarte variabile in functie de fiecare robot si de specificatiile acestuia, cum ar fi greutatea, diametrul rotilor, puterea motoarelor, cat de des se primesc date de la senzor.

Aceste constante sunt extrem de dificil de calculat de mana sau cu vreun tool online, astfel incat pentru controllere PID in aproape toate cazurile se foloseste metoda “Trial and error”. Am subestimat cat timp imi va lua acest lucru si nu am ajuns sa ajung la rezultate ideale. Un alt motiv pentru aceasta concluzie este si ca nu am gasit motoare potrivite, iar servomotoarele mele hackuite cat sa fie continue nu au destula putere cat sa permita robotului sa raspunda la erori mari.

Totusi, pot spune ca pentru un astfel de robot, daca se doreste ajungerea la niste rezultate multumitoare, se pot petrece lejer cateva saptamani doar pe calibrarea parametrilor de mai sus. A fost un proces foarte interesant, foarte frustrant si foarte rewarding (nu in acelasi timp). As recomanda oricui sa parcurga un astfel de proiect.

Download

Arhiva pentru software

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** şi **Resurse Hardware**.

Export to PDF