Robot Pendul Invers

Autor: Ionescu Andrei

Grupa: 333CA

Introducere

Proiectul consta intr-un robotel pe 2 roti, controlat din telecomanda, ce isi tine singur echilibrul.

Scopul acestuia este atat unul didactic (invatat despre controlul automat) cat si unul recreational (practic este o jucarie).

Modificat, proiectul poate fi util in diverse moduri. De exemplu, o varianta mai mare a robotului cu un atasament ar putea servi drept o masuta “smart” care poate urmari utilizatorul. Cu arcuri sau alte modificari ale rotilor, ar putea fi un robot biped capabil sa traverseze teren accidentat.

Descriere generală

Functionarea:

In mod repetat, senzorul de giroscop ofera informatii despre orientarea si acceleratia robotului. Acestea pot fi folosite ca o eroare, diferenta fata de valori tinta, iar un algoritm de control automat PID poate fi folosit de microcontroller pentru a trimite catre driverele motoarelor puterea si directia cu care sa mearga pentru a corecta eroarea mentionata anterior.

Pentru a-si mentine echilibrul, robotul trebuie sa isi mentina centrul de greutate deasupra axei rotilor, iar pentru a se deplasa inainte sau inapoi, robotul trebuie sa isi mentina centrul de greutate in fata, respectiv in spatele axei rotilor. Practic, aceste stari reprezinta valori tinta ale orientarii date de giroscop.

Comenzile robotului sunt transmise prin bluetooth folosind un controller duallshock 4 sau un telefon.

Hardware Design

Piese mecanice folosite:

- Teava PVC
- 2 coliere de teava
- Placi de aluminiu
- Suruburi si piulite necesare
- Roti de diametru 10cm
- Conectori de roata cu ax tip D 6mm

Piese electronice folosite:

- Placa de prototipare
- Pinheaders
- Fire
- Modul S3-DevKitC-1 ce contine un microcontroller ESP32
  - 2 condensatoare de decuplare, 0.1uF si 10uF
- 2 suporti de baterii de 9V
- 2 module de stabilizator de tensiune LM2596 ce folosesc surse in comutatie
- Intrerupator obisnuit cu 3 pini
- Dioda de putere
- Modul giroscop ce contine un MPU6050
  - 2 condensatoare de decuplare, 0.1uF si 10uF
- 2 motoare ce functioneaza la 12V, HG37-060-AA-00
- 2 drivere de motor TB6641FTG
  - Adaptoare/breakout board-uri QFN32
  - 2 rezistente de 20kOhmi
  - 2 condensatoare de decuplare, 0.1uF si 22uF
  - 1 condensator de 1nF

Piese mecanice inlocuite:

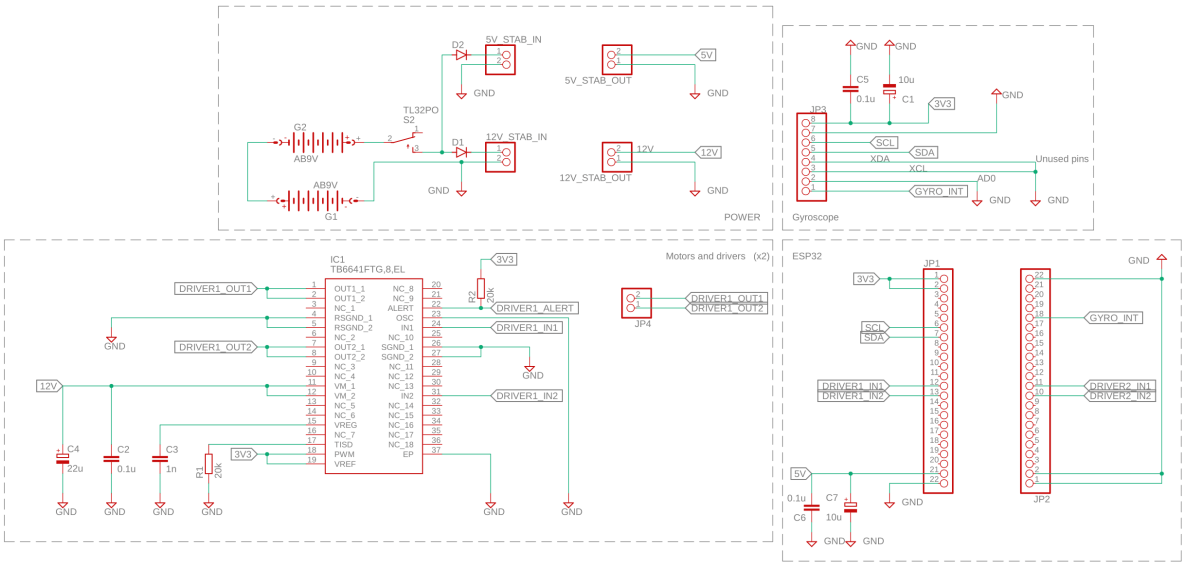
- 2 roti de diametru 7cm (incompatibile cu axul motoarelor)

Piese electronice inlocuite

- Stabilizator liniar de tensiune TS78L03
  - 2 condensatoare, 0.33uF si 470uF
  - 1 dioda de protectie
- Stabilizator sursa in comutatie LM2576T-12
  - 1 bobina 100uH

  - 2 condensatoare, 100uF si 1mF
  - dioda Schottky
- Condensatoare de 1uF (valoare prea mare)

Schema electrica



Software Design

Modelul matematic al pendulului invers:

Pornind de la ecuatia:

$a - \frac{g}{l}sin(\theta) = 0$

unde:

- a este acceleratia unghiulara
- g este acceleratia gravitationala
- l este lungimea pendulului
- θ este unghiul fata de pozitia de echilibru

putem prezice miscarea libera a robotului printr-o metoda matematica. Aceasta este folositoare

deoarece algoritmul de control automat PID necesita un astfel de model pentru a functiona.

Algoritmul de control automat:

Cunoscut ca si algoritmul controller-ului proportional-integral-derivat (PID pe scurt), acesta este o metoda bazata pe o bucla de control cu feedback. Concret, consta in calcularea unei valori de eroare $e(t)$, diferenta dintre valoarea tinta SP si variabila de proces masurata PV, apoi aplicarea unei corectii bazata pe 3 termeni:

- termenul **proportional** modifica valoarea $e(t)$, precum ii spune si numele, **proportional cu eroarea calculata**. Singur, acesta poate functiona, dar pot aparea probleme in cazuri dinamice, precum miscarea accelerata sau cu acceleratie non-constanta.
- termenul **integral** modifica valoarea $e(t)$ in functie de **eroare si timpul** pentru care aceasta a **persistat**. Singur, acesta poate functiona, dar creaza un raspuns prea slab initial si prea puternic in apropierea punctului PV.
- termenul **derivat** modifica **rata de schimbare a erorii**, tintind la una constanta. Singur, acesta nu poate functiona intrucat nu modifica eroarea in sine, dar este cheia functionarii corecte a celorlalti termeni.

Ecuatia:

$u(t) = K_pe(t) + K_i\int_0^t{e(\tau)d\tau} + K_d\frac{de(t)}{dt}$

unde:

- $u(t)$ este corectia ce va fi aplicata sistemului
- K_p, K_i si K_d sunt parametrii alesi prin incercare si esec sau prin calcul matematic (pe baza modelului discutat anterior)
- t este timpul instantaneu

ne ofera schimbarea ce trebuie efectuata asupra puterii motoarelor pentru a mentine echilibrul robotului.

Lock anti-phase drive:

Trebuie mentionat ca driverele de motor sunt speciale, avand capacitatea de a fi controlate prin PWM in mod diferit: pinul de PWM poate fi tras la tensiunea de referinta in timp ce pinii de intrare (ce controleaza directia) pot accepta un semnal PWM. Astfel, un duty cycle de 50% reprezinta o franare rigida (motoarele se vor opune activ miscarii) in timp ce 0% si 100% reprezinta miscare la putere maxima inainte sau inapoi. Aceasta caracteristica este avantajoasa robotului intrucat ajuta la un control mai precis.

Pentru a se deplasa, robotul ar trebui pur si simplu sa isi mentina centrul de greutate in fata sau in spatele axului rotilor. Acest lucru se efectueaza din software prin schimbarea valorii SP DAR mai

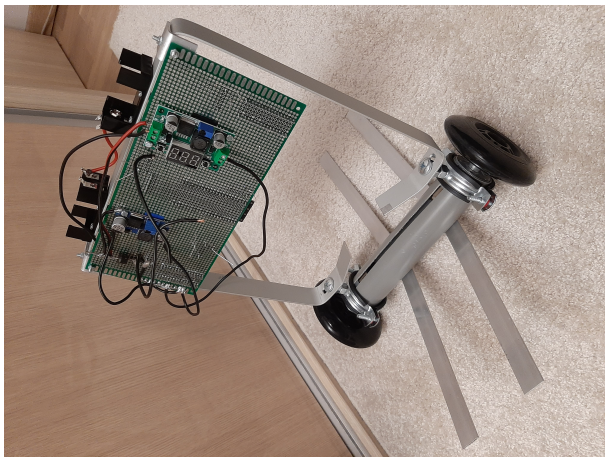
intai ar trebui ca robotul sa plece brusc in directia opusa directiei dorite de mers, lasand inertia sa “traga” rotile de sub greutatea robotului.

Comunicare bluetooth:

Intrucat Arduino are optiunea de a comunica prin bluetooth folosind module externe, am ales un microcontroller ESP32-S3, acesta avand integrat o antena si modulele necesare. Astfel, un controller Duallshock 4 ar putea fi conectata la robot pentru manevrarea wireless a acestuia, intr-un mod relativ simplu: adresa MAC a microcontroller-ului trebuie sa fie aceeasi cu cea stocata de controller.

Rezultate Obţinute

In final, suportul mecanic hardware este terminat, componentele electronice sunt lipite dar lipsesc conexiunile, dat fiind dificultatea cositoririi firelor, iar partea de software este la stadiul de teorie intrucat este greu de implementat/testat fara suportul electronic.



Concluzii

Lucruri ce pot fi imbunatatite

- Tot hardware-ul electronic putea fi pus pe propriul PCB in loc de o placa de prototipare. Acest neajuns corectat ar fi rezolvat la randul lui o multitudine de probleme:
  - Driverele de motor ar fi fost mai usor de lipit.
  - Condensatorii de decuplare ar fi putut fi lipiti mai aproape de pinii lor.
  - Nu ar mai fi fost nevoie de cositorirea multor fire, fapt important deoarece este o activitate ce consuma mult timp si prezinta probleme pentru multitudinea de pini mici si grupati.
- Driverele de motor ar fi putut fi luate intr-o capsula mai usor de lipit de mana.
- Firele ar fi fost mai usor de lipit sub forma de cupru solid, nu manunchi de fire mici.
- O greutate ar putea fi adaugata in varful robotului, pentru ca acesta s-ar putea sa fie prea usor ca sa

se deplaseze atunci cand este inclinat in momentul de fata.

Sunt de parere ca proiectul a fost unul de o complexitate crescuta dar ar fi fost dus la bun sfarsit cu mai mult timp la dispozitie!

Download

O arhivă (sau mai multe dacă este cazul) cu fişierele obţinute în urma realizării proiectului: surse, scheme, etc. Un fişier README, un ChangeLog, un script de compilare şi copiere automată pe uC crează întotdeauna o impresie bună .

Fişierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fişierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul). **Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Proiectul Eagle ce contine schematicul robotului.

Jurnal

Progres:

- Tema a fost aleasa aici.
- Resurse Hardware au fost adaugate.
- Resurse Software au fost adaugate.
- Schema electrica a fost adaugata.
- Lucruri ce pot fi imbunatatite au fost mentionate.
- Functionarea generala a robotului a fost adaugata.
- Design-ul software a fost adaugat.
- Rezultatele obtinute au fost adaugate.
- Concluzia a fost adaugata.

Bibliografie/Resurse

Resurse Hardware:

- Documentatie ESP32
- Datasheet si application notes pentru driver-ele de motor
- Datasheet pentru sursa in comutatie (2 folosite: 12V si 5V)
- Datasheet si application notes pentru stabilizatorul LDO liniar 3V3 (in final nefolosit)
- Datasheet giroscop MPU-6050
- Datasheet motoare 12V

Resurse Software:

- Interfatarea unui ESP32 cu un Dualshock 4
- I2C (1) (2) pentru ESP32

Export to PDF