

Schimbator de viteze automat bicicleta

Introducere

- La biciclete, schimbatorul de viteze are rolul de a muta lantul pe o roata dintata (in cazul proiectului, pe un pinion) mai mare sau mai mica, pentru a modifica numarul de rotatii efectuate de roata pentru fiecare rotatie completa a pedalelor. Astfel biciclistul poate inainta folosind acelasi efort atat pe teren accidentat, la deal (cu o viteza mai mica) cat si pe asfalt, la vale (cu o viteza mai mare).
- Scopul proiectului este de automatizare a procesului de schimbare a vitezelor, capabil sa decida cand este nevoie de schimbare analizand viteza, cadenta si inclinarea curenta si actionand in consecinta. Astfel biciclistul scapa de grija schimbarii vitezelor si se poate concentra 100% pe pedalat 😊.
- Ideea de la care am pornit a fost, evident, transmisia automata de pe autoturisme, venita in ajutorul soferilor, similar, pentru a-i scapa pe acestia de schimbatul vitezelor manual.
- Proiectul este util atat pentru biciclistii incepatori, care nu stiu cum sa isi ajusteze viteza la conditiile in care se afla, cat si pentru cei mai experimentati care vor sa mentina cu precizie un nivel de efort constant.

Descriere generală



Hardware Design

Lista piese:

- Microcontroller
- Motor cu reductor
- Senzor Hall * 2
- Senzor giroscop
- Convertor tensiune DC-DC * 2
- Breadboard
- Fire
- Baterie 12V

Microcontroller-ul va folosi pinii GPIO3 si GPIO4 ca pini de output (pentru a controla modulul releu cu doua canale, si implicit motorul), si pinii GPIO2 si GPIO1 ca pini de input pentru a citi datele de la senzorii Hall. De asemenea, pinii SDA si SCL se vor folosi pentru comunicatia prin I2C cu senzorul de

giroscop.

Modulul releu cu 2 canale permite ca fiecare din cei doi pini ai motorului sa fie conectat la alimentarea de 12V sau la masa, in functie de starea releului corespunzator. Astfel, se poate alimenta motorul in ambele sensuri (pentru a putea schimba vitezele in ambele sensuri - alimentand un pin la 12V si celalalt la masa), si se poate si opri.

Senzorul Hall trimite datele in format digital (detectarea unui camp magnetic). Sensibilitatea senzorului se ajusteaza folosind un potentiometru.

Deoarece motorul trebuie alimentat la 12V, microcontrollerul si modulul releu la 3V, iar restul senzorilor la 5V, se folosesc 2 convertoare de tensiune DC-DC, pentru a trece de la 12V la 5V, respectiv 3V. Tensiunea de iesire se regleaza cu ajutorul unui potentiometru.

Schema electrica:



Stadiu actual al implementarii hardware:

Deoarece mecanica implicata in schimbarea vitezelor cu ajutorul unui motor mi s-a parut una dintre cele mai provocatoare parti ale proiectului, am incercat mai intai sa montez motorul, dar acesta sa fie controlat de un simplu intrerupator - practic schimbarea vitezelor se face tot "manual", dar cu ajutorul motorului 😊. In proiectul final, intrerupatorul va fi inlocuit de releu, care prin semnalul primit de la microcontroller vor comanda schimbarea vitezei.

Initial am vrut sa folosesc un motor pas cu pas pentru a controla cu exactitate lungimea de cablu trasa de motor pentru a trece de la o viteza la alta. Am renuntat la aceasta varianta din cauza cantitatii mari de curent consumate de motorul pas cu pas pentru a mentine o anumita pozitie. Am optat in final pentru un motor cu reductor, care, pe langa cuplul mare pe care il ofera, nu consuma curent cand stationeaza.

Instalatia experimentală din imaginea de mai jos a reusit sa schimbe toate vitezele, in ambele sensuri 😊.



Se observa in partea din fata intrerupatorul care comanda momentan schimbarea vitezei, sub sa cutia in care se afla bateria de 12V (de dimensiuni cam mari, dar este doar un prototip 😊), iar intre cutie si cadru motorul care infasoara/desfasoara cablul care duce la deraior.

Software Design

Am implementat software-ul folosind [microbit:bit Python Editor](#). Am incercat sa folosesc C/C++, dar am avut probleme in configurarea bibliotecilor necesare pentru micro:bit.

Am implementat functii care verifica senzorii Hall (avand grija sa nu contorizez de mai multe ori aceeasi detectie), care schimba viteza in sus/jos - calibrand numarul de milisecunde pentru care motorul trebuie sa functioneze pentru o schimbare prin teste. De asemenea, am implementat o functie care salveaza persistent in memorie viteza curenta - astfel in cazul unui reset se pastreaza cunoscuta starea in care a ramas schimbatorul. O alta functie importanta este cea de recalibrare, care actioneaza schimbatorul spre cea mai mica viteza pana cand este detectat faptul ca s-a ajuns in capat. Aceasta functie este utilizata daca nu se gaseste fisierul cu viteza curenta (in principiu dupa flash), dar si in caz ca viteza curenta nu este 1 dar se detecteaza capatul cablului. Am implementat si o functie pentru citirea acceleratiei de pe senzorul MPU6050, prin i2c. Am folosit datasheet-ul senzorului pentru a afla adresa i2c si adresele registrelor de interes.

In bucla principala, retin mereu timpii ultimelor 3 detectii ale senzorului Hall de pe pedala, si in functie de diferentele dintre ei (deci de timpii ultimelor 2 rotatii complete ale pedalelor), decid daca este nevoie de o schimbare a vitezei. Daca timpii sunt prea diferiti, inseamna ca nu se pedaleaza consistent deci datele se ignora. Daca sunt sub o anumita limita, se comanda schimbarea vitezei in sus, si daca sunt peste o alta limita se comanda schimbarea vitezei in jos. In plus, se seteaza un cooldown: un numar de detectii ale senzorului de pe pedala care nu vor comanda o noua schimbare - pentru a lasa cadenta sa se stabilizeze la un nou nivel dupa schimbarea anterioara. In plus, detectia unei decelerari puternice (franare) - se traduce prin schimbarea vitezei in jos la urmatoarea rotatie a

pedalelor detectata.

Pentru calibrarea motorului am implementat mai intai un program care schimba succesiv vitezele in sus si apoi in jos, la fiecare 3 rotatii ale pedalelor.

Rezultate Obținute

Am obtinut un prototip functional, utilizabil, care chiar daca nu tine cont de la fel de multi parametri pe cat planuiam initial, isi face treaba destul de bine. Lucrul cu senzorul de giroscop/accelerometru a fost dificil si din cauza faptului ca este mai dificil de testat. Atasez o demonstratie, in care vitezele se schimba dupa cadenta (viteza cu care se invart pedalele). Mai intai am simulat o cadenta mare → vitezele se schimba spre pinioanele mai mici, apoi una normala, apoi o cadenta mica, caz in care lantul revine pe pinioanele mai mari.

[Video demo](#)

Atasez si o demonstratie in conditii reale:

[Vide demo 2](#)

Concluzii

Experienta de a duce o idee pana la un prototip functional a fost una inedita si interesanta, din care am avut multe de invatat.

Download

[alexandru.carp332cc.zip](#)

Bibliografie/Resurse

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

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/alucaci/alexandru.carp3008>



Last update: **2024/05/26 22:39**