BLDC Electronic Speed Controller

Student: Ciufu Victor-Andrei
Grupa: 333AA

Introducere

Proiectul meu este un ESC (electronic speed controller) pentru un motor pe curent continuu fara perii. Deoarece acest tip de motor are o functionalitate mai complexa decat un motor DC cu perii, unde doar legi motorul la o tensiune, este nevoie de un mecanism complex pentru a-l actiona. Fiecare bobina trebuie actionata intr-o ordine astfel incat motorul sa se invarta corespunzator si sa genereze un cuplu indeajuns de mare.
In ultima vreme am fost pasionat de motoare electrice, mai ales de cele de la trotinete electrice, eu dorind sa imi modific una. Dupa ce am vazut cum functioneaza acel tip de motoare, am considerat ca as vrea sa fac si eu un controller pentru un astfel de motor, care va putea fi pus pe o varietate de motoare diferite, unde va trebui sa schimb doar anumite componente in caz ca motorul functioneaza la o tensiune mai mare decat ESC-ul meu pentru a-l converti la tensiunea potrivita.

Descriere generală

O schemă bloc cu toate modulele proiectului vostru, atât software cât şi hardware însoţită de o descriere a acestora precum şi a modului în care interacţionează.

Exemplu de schemă bloc: http://www.robs-projects.com/mp3proj/newplayer.html

Hardware Design

Lista de piese necesare:

- Placa Arduino UNO / doar microcontrollerul ATMega328 (sau orice microcontroller cu viteza ceasului mai mare - eu voi folosi placa Arduino UNO pentru programarea pe microcontrollerul ATMega328)
- Rezistente, diode, condensatoare, fire, breadboard, butoane - diverse functii
- 6 tranzistoare IRF540N - Realizarea de “half-bridge”-uri pentru a trimite/trage curent de pe anumite bobine.

- Comparatoare (se pot include si cele de pe microcontrollerul ATMega328) - citirea de Back EMF de la bobinele necomutate
- Ultimele 2 parti se pot realiza si cu utilizarea unu DRIVER
- Motor Brushless de 5V (de preferat, pentru placuta)
- Optional - un ESC functional pentru debugging

===De introdus scheme electrice si diagrame de semnal===

Software Design

Descrierea codului aplicaţiei (firmware):

- mediu de dezvoltare : Arduino IDE / MPLAB X IDE
- ~~librării şi surse 3rd-party (e.g. Procyon AVRlib)~~
- algoritmi şi structuri pe care plănuiţi să le implementaţi:
- (etapa 3) surse şi funcţii implementate

Rezultate Obţinute

Care au fost rezultatele obţinute în urma realizării proiectului vostru.

Concluzii

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**.

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

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

Export to PDF

From:
https://ocw.cs.pub.ro/courses/ - **CS Open CourseWare**

Permanent link:
https://ocw.cs.pub.ro/courses/pm/prj2023/dene/controllerbldc

Last update: **2023/05/05 05:45**