
<i>Bălău Bogdan</i>	<i>Puf Vlad</i>	<i>Romila Andrei-Răzvan</i>
bogdanbalau96@yahoo.com	vlad_puf@yahoo.com	rar021295@gmail.com

Titlul proiectului:

Voice controlled motor

Rezumat:

Proiectul de față își propune realizarea controlului unui Motor BLDC prin intermediul comenzilor vocale implementat după modelul de realizare a unui ECU din domeniul automotive, utilizând mai multe platforme ce îndeplinesc diferite roluri în cadrul sistemului realizat. Mai precis, prin controlul motorului, aplicația de față își dorește setarea unei trepte de viteză (valori între 0 și 5) corespunzătoare unui semnal sonor emis de utilizator (semnalele sonore reprezentând chiar treptele de viteză, respectiv : 0 – “zero”, 1 - “one”, 2 – “two”, 3 – “three”, 4 – “four”, 5 – “five”), precum și afișarea treptei de viteză curentă prin intermediul a cinci LED-uri.

Introducere:

Odată cu evoluția tehnologică, viața cotidiană și-a modificat și aceasta cursul, ținând pasul cu noile inovații. În acest sens, anumite device-uri au devenit indispensabile vieții de zi cu zi, cum ar fi spre exemplu telefonul mobil. Smartphone-urile oferă multiple beneficii utilizatorului de rând, în afară de cele de bază ale unui telefon.

Aplicația realizată oferă încă o latură de utilizare a unui smartphone, și anume, cea de a permite controlul unui motor electric. Concret, prin rostirea către telefon a unei comenzi (“zero”, “one”, “two”, “three”, “four”, “five”), motorul electric își va modifica viteza de rotație conform unor praguri prestabilite.

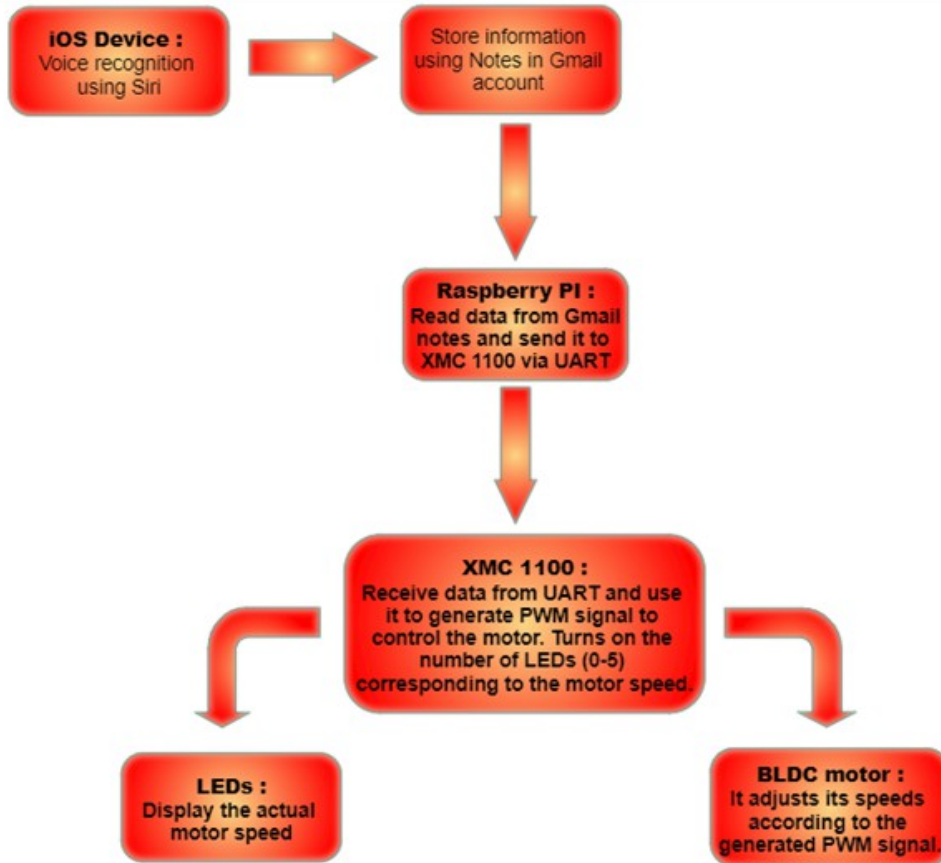
Descriere soluție și detalii de implementare:

Componente utilizate :

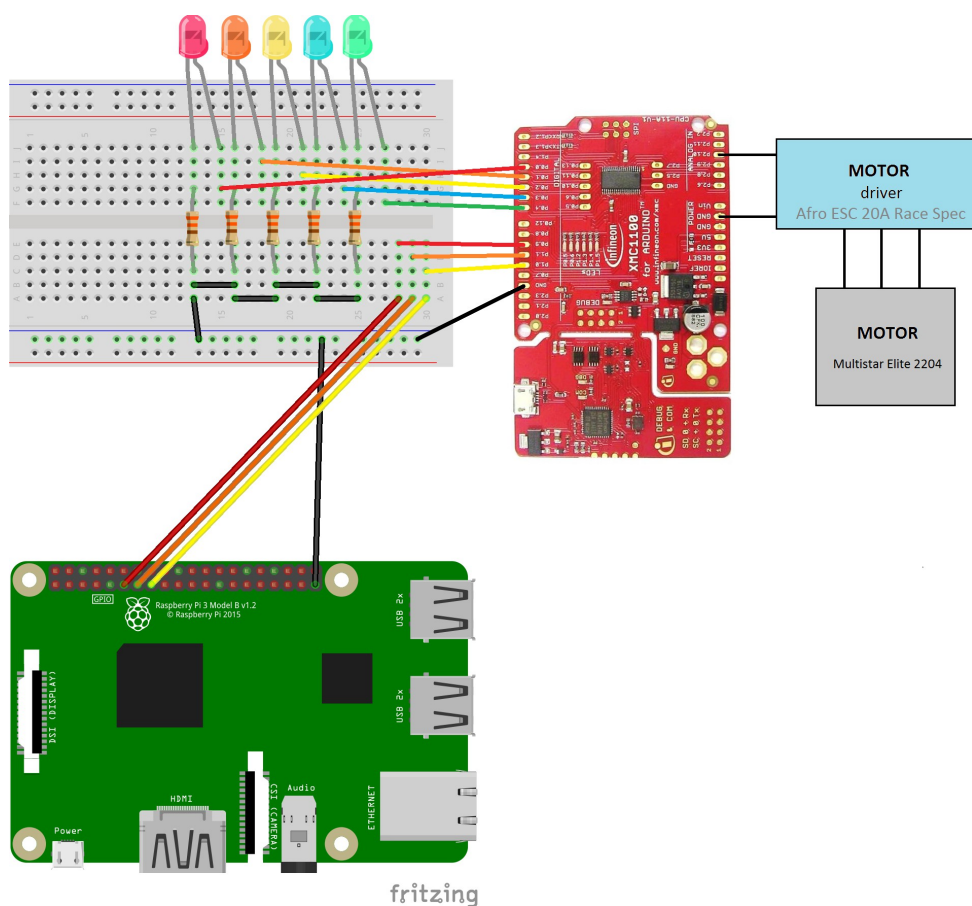
- ➔ Dispozitiv iOS (iPhone)
- ➔ **Raspberry Pi 3 model B**
- ➔ Infineon XMC 1100 Boot Kit
- ➔ Driver de motor BLDC : **Afro RaceSpec 20A ESC**
- ➔ Motor BLDC : **Multistar Elite 2204**
- ➔ 5 LED-uri

Aplicația implică parcurgerea a patru etape :

- 1) Recunoașterea vocală prin intermediul Siri (asistent virtual ce face parte din pachetul Apple Inc.'s iOS);
- 2) Stocarea informației procesate de Siri în contul Google, folosind protocolul IMAP;
- 3) Citirea informației stocate în contul Google utilizând SiriControl (un framework python open source) instalat pe platforma Raspberry Pi;
- 4) Folosirea informației primite, ulterior preluate de către platforma XMC 1100 Boot Kit, setarea vitezei motorului și afișarea treptei de viteză prin intermediul LED-urilor.



Schema proiectului:



Elementul de noutate în cadrul proiectului este reprezentat de utilizarea asistentului Siri disponibil pe produsele Apple ce permite controlul vocal al sistemului.

Practic, utilizatorul va rosti către dispozitiv, în cazul nostru, un smartphone, una din comenzile cheie ce vor avea ca efect rezultatul dorit: “zero”, “one”, “two”, “three”, “four” sau “five”. În continuare, se va crea o “notă” ce va conține comenzile rostite. Aceasta poate fi citită și din contul de Gmail care reprezintă punctul comun dintre dispozitiv și platforma Raspberry Pi, acolo unde informația este citită, interpretată, procesată și mai apoi trimisă mai departe.

Semnalul PWM de control este generat de către platforma de dezvoltare XMC 1100 BootKit, iar indicațiile pentru treapta de viteză sunt transmise de către Raspberry Pi prin intermediul comunicației seriale UART sau prin intermediul unei codificări binare a treptei de viteză. După decodificarea mesajului primit, factorul de umplere se reglează în concordanță cu treptele de viteză indicate în cele ce urmează.

Pentru controlul motorului s-a folosit un driver de motor fără perii (Electronic Speed Controller). Acesta primește ca intrare un semnal de tipul PWM cu o frecvență care de obicei este de 50 de Hertz, având o ieșire trifazată prin intermediul căreia controlează cele trei terminale ale motorului brushless.

După standardul folosit în dispozitivele radiotelecomandate, frecvența folosită este de 50 Hz (o perioadă de 20 de milisecunde sau 20000 microsecunde). De asemenea, pentru controlul motorului, perioada în care semnalul PWM stă în valoarea “1” logic este între 1000 și 2000 de microsecunde. Astfel, pentru o viteză maximă se aplică o perioadă pe “1” logic de 2000 microsecunde, iar pentru o turație minimă (“0” – oprire), valoarea corespunzătoare este de 1000 de microsecunde.

În urma unui calcul simplu, rezultă că pentru turația minimă este necesar un factor de umplere de $1000/20000 = 5\%$, iar pentru turația maximă este nevoie de un factor de umplere de $2000/20000 = 10\%$ (factor de umplere pentru semnalul PWM de intrare în driver).

La prima folosire a sistemului în ansamblu este necesară calibrarea driverului în funcție de generatorul de semnal de intrare – determinarea punctelor de turație minimă și maximă. Astfel, driverul folosit în cadrul acestui proiect are următoarea procedură de calibrare inițială:

1. La alimentarea driverului cu energie electrică, semnalul de intrare trebuie să fie la factorul de umplere ce va corespunde turației maxime.
2. După câteva secunde, prin intermediul unui feedback sonor, se va confirma setarea valorii maxime și se va aștepta valoarea minimă, printr-o variație mare a factorului de umplere a semnalului de intrare.
3. La apariția unui factor de umplere mai mic, tot prin intermediul unui sunet se va confirma finalizarea calibrării. În acest moment, viteza setată este cea minimă, iar driverul este pregătit pentru a învârti motorul pentru un semnal valid (între valoarea minimă și maximă a factorului de umplere abia setate).

Observații:

- Această setare presupune respectarea intervalului de 5% – 10% a factorului de umplere. În cazul unei valori în afara acestui interval, driverul nu va recunoaște semnalul ca fiind unul valid, iar procesul de calibrare poate să nu ofere rezultatele așteptate.
- Valorile actuale setate pentru minim și maxim pot să nu fie exact 5% și 10% și vor fi diferite de la telecomandă la telecomandă. Din acest motiv este necesară calibrarea de fiecare dată când se schimbă driverul sau telecomanda, pentru ca actuatorul să fie capabil să controleze întreaga gamă de viteză a motorului.

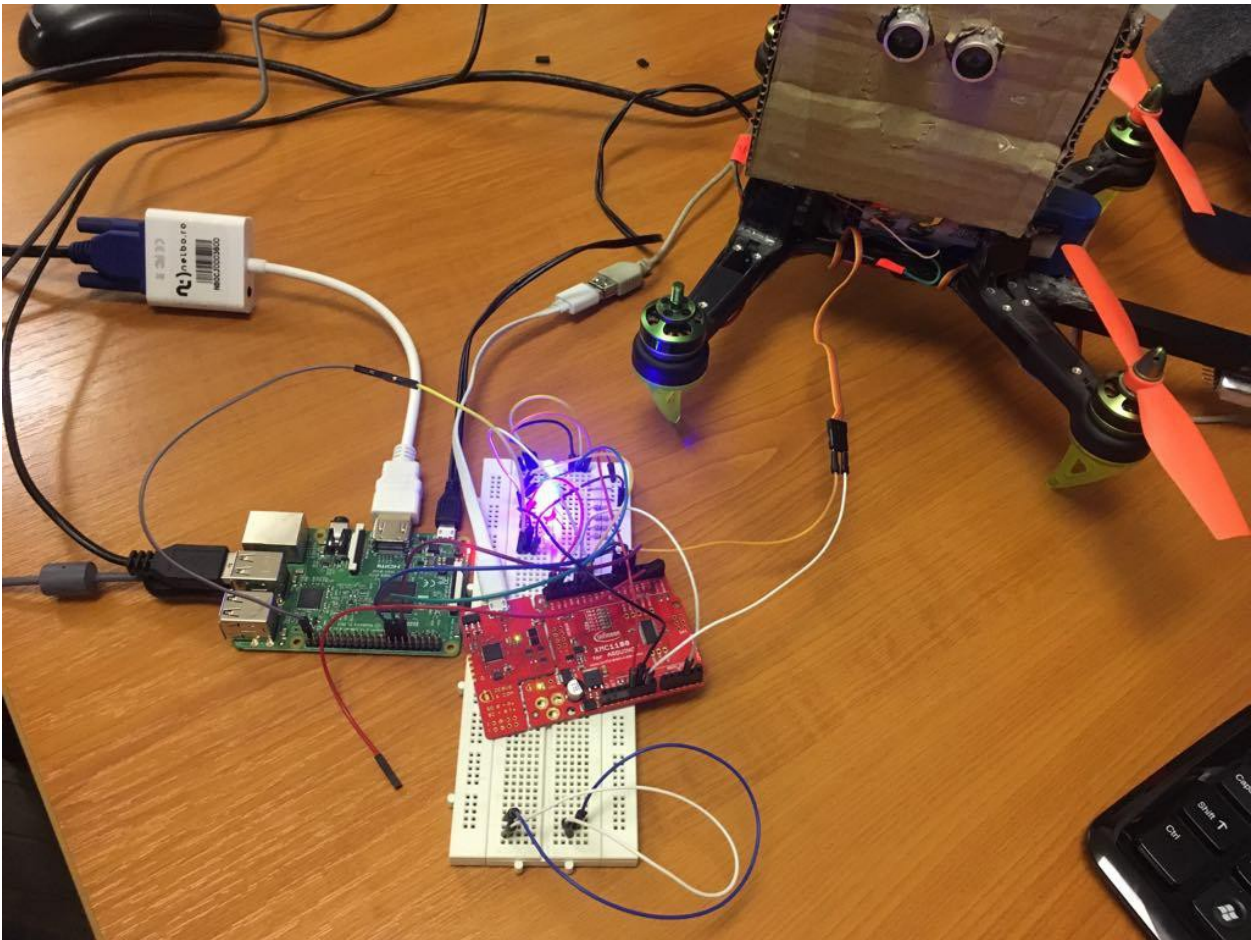
În cazul acestui proiect, calibrarea s-a realizat aplicând un factor de umplere de 5% pentru valoarea minimă și de 7.4% pentru turația maximă.

Cum controlul se realizează în 5 trepte, plus modul “oprit”, valorile pentru factorul de umplere a semnalului PWM de control vor varia în concordanță: 5% - oprit, 5.5% - viteza 1, 6% - viteza 2, 6.5% - viteza 3, 7% - viteza 4, 7.4% - viteza 5 (considerata maximă).

Cuvant	Reprezentare	Factor de umplere	Viteza	Display
"zero"	0	5%	Motor oprit	0 LEDs ON
"one"	1	5.5%	0.2*Viteza_maxima	1 LED ON
"two"	2	6.0%	0.4*Viteza_maxima	2 LEDs ON
"three"	3	6.5%	0.6*Viteza_maxima	3 LEDs ON
"four"	4	7.0%	0.8*Viteza_maxima	4 LEDs ON
"five"	5	7.4%	Viteza_maxima	5 LEDs ON

Pentru semnalul PWM s-a utilizat un modul standard, componentă a mediului de dezvoltare DAVE, acesta fiind configurat după necesitățile aplicației curente.

Rezultate:



Bibliografie:

- <https://www.raspberrypi.org/magpi/siricontrol-control-raspberry-pi-siri/>
 - <https://github.com/theraspberryguy/SiriControl-System>
-