Ciclocomputer

Autor

Teodora Argintaru

Introducere

Fiind pasionată de sporturi de anduranță, știu cât de important este pentru practicanții de un astfel de sport să aibă în orice moment dintr-un antrenament o idee clară asupra efortului depus până atunci. Astfel, proiectul pe care l-am ales constă într-un ciclocomputer, deoarece consider că un astfel de device este util oricărui ciclist.
Scopul principal al proiectului este familiarizarea cu plăcuța Arduino și modul în care diferiți senzori interacționează cu aceasta. Computerul de bicicletă va oferi informații în timp real despre viteză, timpul trecut de la începutul turei, precum și despre elevația din timpul acesteia.

Descriere generală

Ciclocomputerul va fi un device care se va monta pe bicicletă. Singura parte vizibilă pentru utilizator va fi ecranul LCD, unde vor fi afișate informațiile relevante. **Viteza roții** va fi calculată cu ajutorul unui senzor de efect Hall, și un magnet prins pe roata din față.
Elevația va fi calculată cu ajutorul modulului de accelerometru și giroscop.

Schema bloc

Hardware Design

Lista de componente

- Arduino UNO

- Modul LCD Nokia 5110
- Fire
- Modul senzor efect Hall (luat de la un device de kilometraj)
- Magnet
- Modul accelerometru și giroscop MPU6050

Schema electrică

Software Design

Pentru implementare, am folosit interfața I2C pentru comunicația cu modulul MPU6050 (alături de biblioteca MPU6050_tockn) și comunicația SPI (implementată in software în biblioteca *Adafruit_PCD8544.h*) cu modulul LCD Nokia 5110. Pentru senzorul de efect Hall, am testat mai întâi cu un multimetru polaritatea celor doi pini (pentru că senzorul făcea parte inițial dintr-un vitezometru) și l-am conectat la pinul digital 8, în modul INPUT_PULLUP.

Structura codului

- funcția **setup** inițializează pinul pentru senzor Hall, display-ul și comunicația I2C cu senzorul MPU6050
- funcția **loop** afișează de fiecare dată noile informații
- funcțiile **print_speed**, **print_elevation**, **print_time_elapsed** realizează afișarea efectivă
- funcția **calculate_elevation** adaugă la elevația curentă abs(sin(variație_OY)), în cazul în care variația trece de un prag setat
- funcția **calculate_speed_elevation** verifică dacă este prima dată când se întâlnește magnetul după o perioadă în care starea a fost HIGH (adică s-a făcut o rotație completă a roții din față) și dacă da, atunci calculează viteza curentă și elevația; dacă a trecut prea mult timp de când nu s-a mai întâlnit magnetul, atunci viteza este probabil 0
- funcția **calculate_time_elapsed** actualizează timpul trecut de la începerea turei

Rezultate obținute

Link demo proiect Cod sursă

Concluzii

În concluzie, proiectul la PM a reprezentat o provocare interesantă anul acesta. Cred că una dintre cele mai dificile părți la proiectul meu a fost calibrarea și poziționarea senzorilor (parte la care sunt destul de sigură că încă se mai pot aduce îmbunătățiri 😃). Proiectul reprezintă un punct de plecare bun pentru un device pe care chiar l-aș putea folosi atunci când merg cu bicicleta. Singurul dezavantaj este probabil dimensiunea componentelor, mult mai incomodă decât cea a unui ceas care ar avea aceleași funcții.

Bibliografie/Resurse

- https://github.com/tockn/MPU6050_tockn/blob/master/examples/GetAllData/GetAllData.ino
- https://create.arduino.cc/projecthub/muhammad-aqib/interfacing-nokia-5110-lcd-with-arduino-7bfcdd
- https://maker.pro/arduino/tutorial/how-to-interface-arduino-and-the-mpu-6050-sensor

ciclocomputer.pdf