Robot de pază

Introducere

Proiectul realizează un robot pe două roți care își păstrează singur echilibrul cu ajutorul celor două motoare și al modulului cu accelerometru și giroscop. Robotul este controlat de către utilizator prin intermediul unei aplicații de pe telefon prin apăsarea unor butoane pe fiecare direcție de mișcare sau poate fi programat să se plimbe într-o zonă în față și în spate pe o anumită durată de timp. Conține un senzor de mișcare ce declanșează o alarmă la o detecție.

Ideea a pornit de la dorința de a crea un robot controlat remote. Ideea de autobalansare mi se pare una interesantă și un plus față roboții normali pe 4 roți oferind o mișcare mai fluidă decât aceștia.

Scopul ales este de a oferi o soluție de securitate însă robotul poate fi folosit și în alte scopuri cum ar fi explorare prin atașarea unei camere sau ca robot de transport deoarece este capabil să păstreze obiecte în echilibru.

Descriere generală

Robotul va utiliza pentru mișcare două motoare stepper conectate la Arduino prin intermediul a două drivere. Motoarele vor fi controlate prin PWM atât prin input de la utilizator, cât și automat prin inputul primit de la accelerometru și giroscop pentru echilibrare.

Inputul utilizatorului va fi transmis din aplicația de pe telefon prin intermediul modulului de Bluetooth. Utilizatorul va avea posibilitatea de a controla mișcarea robotului in orice direcție și rotația acestuia, sau setarea unui timp pe care robotul se va deplasa înainte și inapoi.

Accelerometrul si giroscopul vor transmite catre Arduino viteze si unghiul de inclinare și în funcție de acestea se vor controla motoarele pentru a realiza echilibrarea. De exemplu, dacă robotul se înclină în față, se va deplasa puțin tot în față pentru a se balansa.

Microcontrollerul va comunica de asemenea cu senzorul de miscare cu infraroșu. Acesta ca transmite permanent date și la detectarea unei mișcări, plăcuța va declanșa o alarma prin intermediul unui buzzer și va trimite o notificare pe telefon tot prin Bluetooth.

Hardware Design

Lista de piese:

- arduino uno
- Modul Accelerometru și Giroscop cu 3 Axe MPU6050
- Buzzer Activ cu Sunet Continuu SFM-27 DC 3-24V
- 2x Motor Pas cu Pas 17HS4401 (1.7 A, 40 N·cm)
- 2x Driver pentru Motoare Pas cu Pas A4988
- Modul Senzor PIR HC-SR501
- Modul Bluetooth Master Slave HC-05
- 2x roti
- fire de legatura
- breadboard
- Acumulator LiPo Gens Ace, tensiune 11.1V, capacitate 700mAh
- o rezistenta 1k
- o rezistenta 2k

Componenta principala este microcontrollerul arduino uno. Dupa ce am asamblat robotul fizic, pentru a controla motoarele le-am legat la 2 drivere deoarece acestea necesita o tensiune de alimentare mai mare decat arduino, deci motoarele vor fi controlate prin intermediul driverelor si sunt alimentate de acumulatorul LiPo. Giroscopul a fost conectat la placuta prin I2C si un pin de intreruperi pentru a semnala cand are date de citit. In plus am conectat si senzorul de miscare pe pinul 3, iar la citirea pe acest pin se va pune pe high pinul 13 unde este legat buzzerul. Modulul de bluetooth este configurat in modul slave si comunica cu arduino pe interfata seriala UART.

Initial conectasem modulul de bluetooth la pinii 0 si 1 de la arduino, dar asta ingreuna debug-ul cu printurile pe Serial si incarcarea codului. Apoi au fost mutati pe pinii 2 si 3 dar dintr-un anumit motiv modulul nu voia sa primeasca date de pe pinul 3 deci in final a ramas pe 5 si 6.

Am adaugat si un divizor de tensiune dupa ce am stricat Rx de la un modul de bluetooth pentru ca nu am vazut ca primeste 3V3.

Initial legasem senzorul de miscare la pinul 7, dar l-am mutat pe 2 pentru a putea genera intreruperi.

Software Design

Mediul de dezvoltare folosit pentru cod a fost Arduino IDE. Pentru dezvoltarea aplicatiei pe telefon am folosit MIT App Inventor.

Pentru balansarea robotului am folosit algoritmul PID, ce calculeaza puterea data motoarelor folosindu-se de 3 constante:

- Kp = raspunsul proportional cu eroarea
- Kd = proportional cu derivata erorii, aproximata cu diferenta dintre eroarea curenta si cea precedenta impartita la timpul la care se face citirea de la mpu si recalcularea (adica 5ms)
- Ki = termenul integral, genereaza raspuns bazat pe acumularea erorii, este suma erorilor precedente

Citirea de la mpu si calcularea raspunsului se realizeaza la 5ms, logica fiind implementata cu ajutorul unui timer care seteaza un flag de update intr-o intrerupere la numararea de 5ms.

Pentru partea de detectare a miscarii, senzorul de miscare este legat la pinul 2 al uc si va genera o intrerupere externa pe rising edge la detectarea unei miscari pentru a evita polling-ul. In RTI se pune pe high pinul de alimentare al buzzer-ului, daca acesta a fost alimentat din aplicatie.

Pentru comunicarea cu aplicatia de pe telefon prin Bluetooth, am realizat o interfata seriala pe pinii 5 si 6. Din aplicatie se pot trimite comenzi de Up, Down, Left, Right care ar controla miscarea robotului (daca ar fi fost implementata) si de asemenea poate fi activat buzzerul pentru a emite un sunet de o secunda la detectarea unei miscari. De asemenea la detectarea unei miscari programul va trimite telefonului un caracter prin bluetooth si se va primi o notificare.

Biblioteci folosite:

- I2Cdev.h
- MPU6050.h
- math.h
- AccelStepper.h
- SoftwareSerial.h

Functii implementate:

- handleInterrupt() - intreruperea externa pentru senzorul de miscare care activeaza buzzerul
- init_PID() - initializeaza timerul de 5ms pentru citirea de la senzor si update-ul motoarelor
- setup() - initializeaza intreruperile, interfetele seriale, directia pinilor si calibreaza modulul mpu
- ISR(TIMER1_COMPA_vect) - intreruperea pentru update, seteaza un flag de update
- loop() - logica principala a programului. Se verifica daca avem ceva de citit de pe interfata bluetooth sau daca s-a detectat miscare si trebuie trimisa o notificare. Apoi daca flagul de update a fost setat, se citesc datele de la senzor si se calculeaza noua viteza a motoarelor
- readSensors() - citeste date de la modulul mpu si calculeaza inclinatia robotului pe baza accelerometrului si giroscopului
- calculatePID() - calculeaza pe baza algoritmului PID noua putere a motoarelor

Tot codul se poate gasi in arhiva de mai jos

Rezultate Obţinute

Din pacate, dupa o saptamana de nopti nedormite, nu s-a obtinut un robot foarte precis. Pe suprafete cum ar fi un covor reuseste sa isi pastreze echilibrul, dar pe suprafete mai netede nu reuseste mai mult de 1-2 secunde. Din acest motiv nu am mai implementat controlul miscarii deoarece nu era posibil. Acest rezultat nefericit se datoreaza probabil in cea mai mare parte constructiei corpului robotului, care ar fi trebuit sa aiba o inaltime mai mare, dimensiunii rotilor care sunt prea mici. O constructie mai buna impreuna cu o ajustare mai atenta a constantelor ar fi putut duce la rezultate mai bune. Am atasat un video cu un demo:

Concluzii

Un proiect dificil, care nu functioneaza din pacate la capacitate maxima. Mai multa atentie in constructia corpului robotului ar fi fost folositoare. Dar, pe partea de electronica si programare pot spune ca am invatat destul de multe lucruri de-a lungul implementarii si mi-am antrenat experienta (si rabdarea) la rezolvarea bugurilor de hardware/software low level.

Download

robotdepaza1.zip

Jurnal

Nu am tinut un jurnal propriu zis in timpul implementarii proiectului, dar s-au intamplat atatea rele incat trebuie documentate:

- Am primit placuta stricata, prima seara de lucru inceputa cu speranta si entuziasm a fost din pacate terminata rapid
- S-a rupt un fir de la legatura cu modulul MPU si pana a fost descoperit acest lucru s-a pierdut mult timp si nervi
- La inceputul unei noi zile de lucru cu entuziasm, dupa 10 minute s-a rupt firul de la plusul bateriei si nu am mai avut unde sa il prind in ziua aceea. Macar s-a umplut timpul cu implementarea aplicatiei.
- Pinul Rx de la modulul bluetooth s-a stricat, suspectez ca este pentru ca nu am observat ca se alimenteaza la 3V3 si l-am legat direct la un pin arduino care da 5V
- Noul modul de bluetooth, venit direct din magazin cu folie, tot nu dorea sa primeasca date. Dupa ore de debug si intrebari de care sunt sansele sa primesti doua componente defecte, am mutat modulul pe pinii 5 si 6 si a functionat (poate s-a stricat pinul 3 de la arduino?)
- Pentru a pune capac tuturor relelor laptopul pe care lucram a decedat brusc, doctorii stabilind cauza ca fiind o placa de baza stricata. Din fericire am reusit sa recuperez codul, dar s-a pierdut o alta zi de lucru.

Bloopers:

Bibliografie/Resurse

- https://www.instructables.com/Arduino-Self-Balancing-Robot-1/
- ds-a4988.pdf
- ds-hc05.pdf
- ds-pir.pdf
- ds-mpu-6050.pdf
- https://www.youtube.com/watch?v=aQcJ4uHdQEA

Export to PDF