

Saber

Introducere

Aceasta pagina documentează realizarea unui robot mini-sumo.

Autor : Darmaz Andrei-Sebastian 332AC

Descriere generală

Ca și în artele marțiale tradiționale japoneze, roboții încearcă să își împingă oponentul în afara ringului.
Reguli:

- Dimensiuni maxime: robotul trebuie să încapă într-un pătrat 10 cm x 10 cm (100 x 100 mm).
- Masa maximă a robotului nu trebuie să depășească 0,50 kg (500,00 gr.).
- Robotul după start se poate extinde. Nu este permis ca robotul să se separe în 2 sau mai multe bucăți. El trebuie să rămână un singur robot centralizat. Roboții care nu vor respecta aceste reguli vor pierde meciul.
- Șuruburi, piulițe, șaibe sau alte părți cu masa sub 5 gr, care se desprind, nu vor determina pierderea meciului.
- Toți roboții trebuie să fie autonomi. Poate fi implementat orice sistem de control, atâta timp cât acesta este în interiorul robotului și nu interacționează cu un sistem de control exterior (om, mașină sau oricare altul).

Schema bloc:



Hardware Design

Lista de componente: Microcontroller:

- Raspberry Pi® RP2040

Senzori:

- TFMini Plus Micro LiDAR Module
- Sharp GP2Y0D810Z0F
- IMU LSM6DSOXTR (6-axis)
- MICROPHONE MP34DT05

Conectivitate:

- Wi-Fi Nina W102 uBlox module
- BLUETOOTH® Nina W102 uBlox module
- SECURE ELEMENT ATECC608A-MAHDA-T Crypto IC

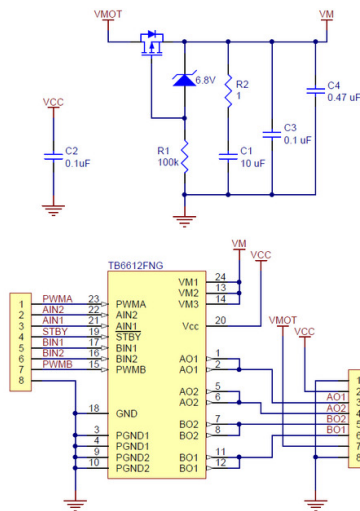
Alimentare:

- LiPo Gens Ace 7V6 300mA 75C
- Step down 3V3 DFR0570
- LM7805

Mecanice:

- Pololu 10:1 Micro Metal Gearmotor HPCB 6V
- Solarbotics RW2i Wheel
- Pololu TB6612FNG

Șasiul robotului a fost realizat manual din tabla de oțel de 1mm. Pe vârful acestuia, plug, voi lipi o



lama din titan.

Software Design

Mediu de dezvoltare folosit a fost Arduino IDE.

Biblioteci folosite : MBED_RP2040_PWM, TFMPlus, WiFinINA.

Deoarece am utilizat microcontrollerul ARM RP2040 care dispune de doua coruri codul este împărțit astfel :

Primul core :

- initializeaza pinii ca INPUT/OUTPUT
- initializeaza comunicația UART cu senzorul LiDAR
- acționează motoarele generând semnale PWM pure hardware. Prin urmare semnalele nu sunt întrerupte de eventuale blocaje din cod
- interpreteaza strategia inițiala a robotului prin citirea dipswitch-ului

- selectează modul de cautare / atac al robotului pentru runda curentă
- se ocupa de întreruperile generate de TSOP

Al doilea core:

- Comunica cu senzorul LiDAR și transmite către core 1 distanța față de oponent
- se ocupa de întreruperile generate de QTR

Am implementat mai multe metode de cautare :

- robotul se învârtă până când găsește adversarul
- robotul se mișcă random până la marginea ringului după care se rotește înapoi spre centru
- robotul avansează încet spre centrul ringului

Fiecare căutarea are avantajul ei :

- rotirea conferă o viziune completă a ringului, dar face ca lama robotului să trepideze deci robotul este mai vulnerabil. În plus dacă acesta nu reacționează rapid la detecția adversarului va pierde timp până când va ajunge din nou cu fața la acesta.
- mișcarea către margine este o strategie clasică, robotul nu este staționar astfel nu este o țintă ușoară și acoperă o distanță destul de mare deci poate detecta adversarul relativ ușor. Insa, pentru ca robotul să se mișcă lama nu este fixată în ring deci se poate intra mai ușor sub ea.
- robotii mini sumo folosesc ultima strategie când sunt siguri de acuratețea senzorilor principali. Pentru ca avansarea către centru, dar și către adversar este lentă lama robotului nu tremură foarte mult deci sunt șanse cât mai mari să intre sub cea a adversarului și să îl ridice scizându-i aderența. Robotul este vulnerabil în laterale fiind destul de staționar. Acest defect poate fi evitat folosind senzorii laterali (sharp/laser omron) însă din experiență dacă adversarul a ajuns deja în lateral o întoarcere bruscă cu ± 90 va destabiliza robotul și acesta va pierde în continuare.

Aceste căutări sunt implementate în funcții diferite din Arduino și sunt selectate prin determinarea strategiei de pe dipswitch. Tot înaintea meciului pe dipswitch se selectează și atacul principal pe care robotul o să îl facă.

Atacul inițial oferă un avantaj robotului prin cunoașterea poziției față de adversar. Prin selectarea strategiei 0-16 îi comunicăm robotului în ce poziție să se miște. Astfel am implementat câteva strategii clasice folosite în meciuri de sumo :

- Rotire cu ± 90 față de poziția inițială - dacă robotul adversar se află în stânga/dreapta odată cu pornirea meciului vom ajunge cu fața către el fără să îl mai căutăm în ring.
- Arc stânga/dreapta - se folosește dacă credem că robotul adversar o să ajungă cu fața către noi. Prin descrierea unui arc vom putea ajunge în spatele/lateralul acestuia.
- 180 de grade - pentru meciurile spate - spate
- etc alte strategii asemănătoare cu cele menționate anterior.

Interacțiune cu senzorii:

- Citirea senzorului LiDAR TOF se face prin UART BR 115200 Fq 1MHZ (senzorul este capabil de BR mai mare, dar devine instabil). Acesta returnează distanța față de obiectul detectat (max 12m - puțin cam overkill), temperatura senzorului și acuratețea de care crede că are dovada. În cod folosesc doar distanța față de target. În alta versiune a codului am setat senzorul pe modul de detection response, mod în care generează o întrerupere de fiecare dată când un obiect intră în zona critică setată în registrii. Încă testez pentru a afla care versiune este mai bună.
- Senzorii QTR (de linie) generează o întrerupere high edge de fiecare dată când detectează culoarea

albă și astfel robotul se va îndepărta de margine.

- TSOP-ul generează o întrerupere când primește semnal de la telecomanda, în handler se citește instrucțiunea primită pentru a afla dacă robotul trebuie să pornească sau să se oprească.
- DipSwitch-ul este citit doar inițial și se face o adunare binară pe cei 4 pini pentru a stabili strategia. Puteam să folosesc un shift register sau un NE555 pentru a folosi mai puțini pini digitali, dar nu am dispus de spațiu pe PCB.

Interacțiunea cu motoarele:

- Fiecare motor are câte un drive asignat care rezistă la 2.4A peak.
- Motoarele sunt comandate pe 3 pini - 2 pini de direcție și unul de PWM
- Generarea de PWM se face pure hardware putând să se stabilească duty cycle și frecvența.

Posibile upgrades :

În viitorul nu foarte îndepărtat o să mai adaug citirea datelor de la accelerometrul de pe placă. Din puținele teste făcute până acum datele nu vin destul de rapid de la senzor pentru a-l implementa în codul core al robotului. Conexiune Bluetooth cu un device pentru a selecta strategia/desena atacul inițial - nu am implementat în core fiindcă conexiunea bluetooth nu este tot timpul stabilă, iar unele concursuri nu permit conexiuni externe în timpul meciurilor, dar poate să fie o funcție fun. Pornire/oprire folosind microfonul de pe board - am implementat antrenând un model ML cu TensorFlow, dar ocupă un core și nu este practic în timpul meciurilor. Alimentare cu 12V în loc de 8V pentru motoare - am testat deja totul era funcțional, însă nu voiam să ard motoarele fix înainte de prezentare :)

Rezultate Obținute

Am obținut încă un robot mini-sumo de adăugat la colecție. Pot spune că este cel mai avansat de până acum din punct de vedere mecanic, iar la partea software este tot timpul loc de îmbunătățire. Saber a reușit să îmi bată robotul anterior care avea în palmares un loc 8 la Robochallenge și loc 2 la Infomatrix. Deci o să așteptăm să vedem cum se descurcă la concursurile din viitor.

Tot ca rezultat aș putea spune că mi-am mai dezvoltat skillurile hardware și am învățat mai multe despre arhitectura ARM, eu fiind obișnuit numai cu AVR.

Concluzii

Mă bucur că am ales să fac PM și că mi-a dat impulsul de a începe un nou proiect. Robotul consider că este unul reușit, deși există tot timpul loc de îmbunătățiri.

Download

Jurnal

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

Bibliografie/Resurse

https://github.com/khoih-prog/RP2040_PWM#features

https://cdn.sparkfun.com/assets/1/4/2/1/9/TFmini_Plus_A02_Product_Manual_EN.pdf

<https://github.com/budryerson/TFMini-Plus>

<https://datasheets.raspberrypi.com/pico/raspberry-pi-pico-c-sdk.pdf>

<https://p1r.se/startmodule/implement-yourself/>

<https://docs.arduino.cc/hardware/nano-rp2040-connect>

<https://cdn-reichelt.de/documents/datenblatt/A300/ABX00053-DATASHEET.pdf>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/cristip/saber>



Last update: **2022/05/27 20:09**