

Ryobo - Computer Vision & Object Following

Introducere

Ryobo: Un robot care urmareste obiecte si detecteaza culori, utilizand computer vision

Urmărirea obiectului se realizează în 2 moduri:

1. la distanța fixă față de obiect (mobilitate în spațiu);
2. pe 2 axe (sus-jos și stânga-dreapta) de mișcare a senzorului optic folosit, urmărind centrarea obiectului în câmpul surprins de camera.

Ideea & inspirație

1. Camera PixyCam 2.1 este folosită în multe aplicații de robotica, datorită programelor sale deja implementate, cum ar fi line tracking sau color detection.
2. O sursă de inspirație este surprinsă chiar în documentația oficială a PixyCam (<https://pixycam.com/projects/>), deoarece, pentru a exemplifica funcționalitățile camerei, sunt date câteva exemple de proiecte.
3. Am utilizat PixyCam 2.1 și pentru alte proiecte. De exemplu, funcția de line tracking este utilă pentru urmărirea unui traseu predefinit format din 2 linii de contrast. Liniile sunt convertite în vectori, iar apoi, datele obținute pot fi procesate astfel încât să se obțină traiectoria optimă pe care un robot sau o mașină autonomă/cu funcționalități mai complexe de self driving o poate urmări.

Scopul proiectului

În general, roboții de urmărire a liniilor și roboții de urmărire a obiectelor pot fi utili într-o gamă largă de aplicații, de la automatizare industrială (manipularea materialelor), la educație și divertisment (ghidarea vizitatorilor în muzee și în alte atracții turistice), dar chiar și supraveghere sau securitate (urmărirea vehiculelor). Algoritmii implementați pot fi folosiți și în alte sisteme cum ar fi drone, calculatoare de bord etc. Pe măsură ce tehnologia avansează, utilitatea și aplicațiile potențiale ale acestora sunt susceptibile să se extindă și mai mult.

Descriere generală

Schema bloc a proiectului oferă o idee generală despre componente, modul în care acestea vor acționa, dar și modul în care comunică între ele. Camera este modalitatea în care Ryobo interacționează cu exteriorul și primește inputul. PixyCam 2.1 comunică cu ATmega328P prin SPI. Odată ce un obiect este detectat, iar signatura sa se potrivește cu cea a obiectului urmărit, microcontrollerul procesează datele și comandă motoarele. Cele 2 servomotoare sunt comandate prin pulse width modulation direct (PWM) de la pinii ATmega328P. Micromotoarele DC 30:1 sunt comandate de driverul specializat DRV8833, tot prin PWM.

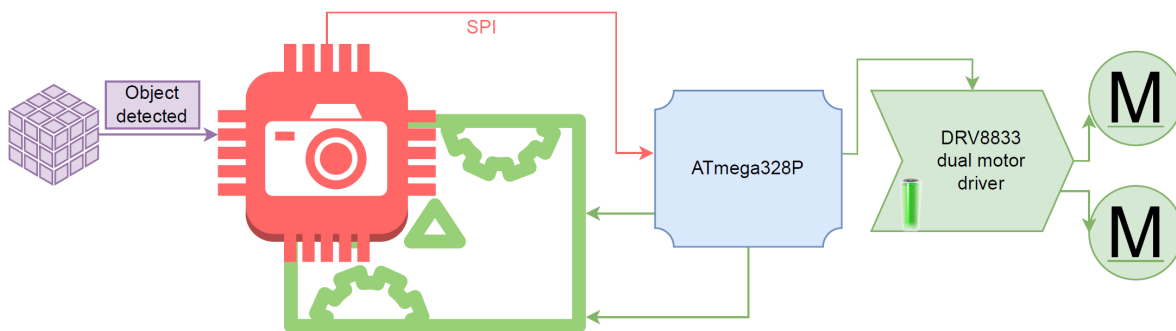
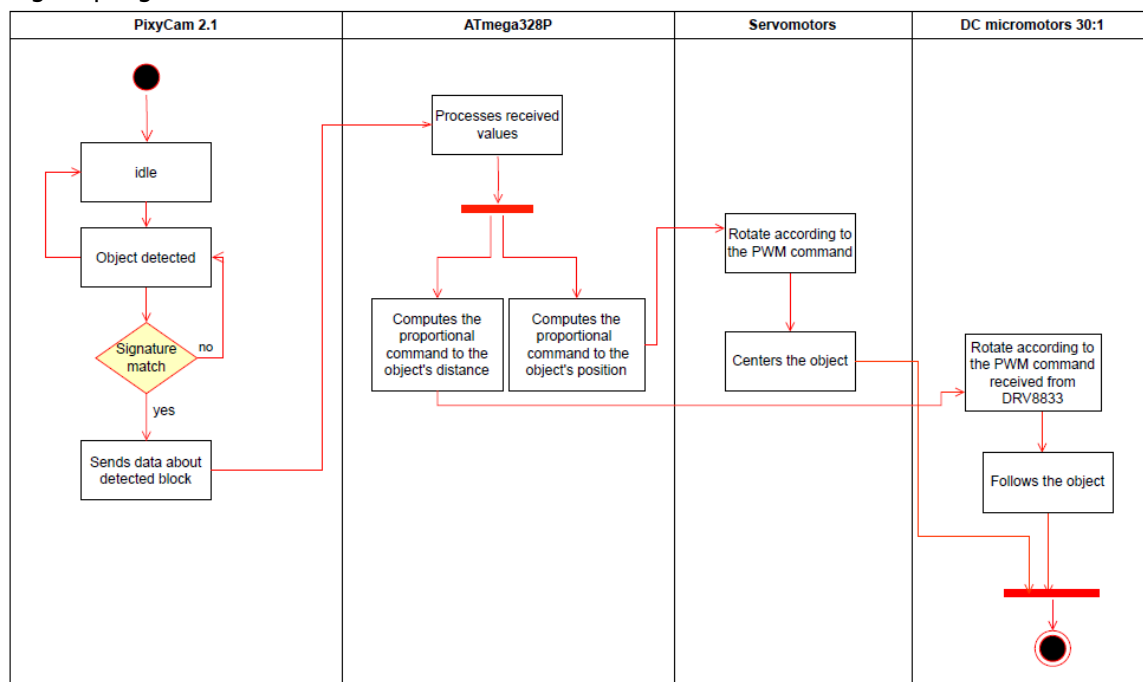


Diagrama de activitati pentru Ryobo contine principalele functionalitati, dar si ordinea executarii si logica programului:



Functionalitati

- Folosirea outputului de la camera Pixy Cam 2.1 pentru a urmari obiecte. Outputul camerei este generat pe baza contrastului dintre culori: o culoare puternica folosita pentru obiectul detectat, fata de fundal.
- Datele primite de la camera sunt procesate prin intermediul microcontrollerului ATmega328P si folosite in algoritmul pentru urmarirea obiectului detectat.
- Cu ajutorul celor 2 motoare Pololu, Ryobo se deplaseaza pentru a mentine o distanta fixa fata de obiectul detectat. Rotile din cauciuc sunt atasate la axurile motoarelor, pentru o aderenta mai buna cu mai multe suprafete de contact.
- Servomotoarele ii ofera mobilitate camerei, miscand suportul pe care aceasta se afla. Astfel, camera este permanent orientata catre obiectul detectat.

Hardware Design

Componente

Piesele folosite pentru a asambla robotul:

- Pixy Cam 2.1 (<https://pixycam.com/>) este o cameră de computer vision dezvoltată de către Charmed Labs. Aceasta poate detecta obiecte bazate pe culori, urmărindu-le la o rată de până la 60 de cadre pe secundă. Camera este capabilă să detecteze până la 7 culori diferite și poate comunica cu un microcontroller prin intermediul unui protocol I2C, SPI sau UART. De asemenea, camera este echipată cu un procesor dual-core NXP LPC4330, care permite procesarea rapidă a datelor în timp real.

- Microcontrollerul ATmega328P (

https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf) de pe placa de dezvoltare Arduino Uno, capabil să controleze diverse dispozitive electronice (de exemplu, servomotoarele și motoarele de curent continuu) și senzori (de exemplu, Pixy Cam 2.1, putând să comunice prin intermediul protocoalelor I2C, SPI și UART). Placa include o serie de pini de intrare/ieșire, precum și o serie de componente auxiliare, cum ar fi un regulator de tensiune și un cristal.

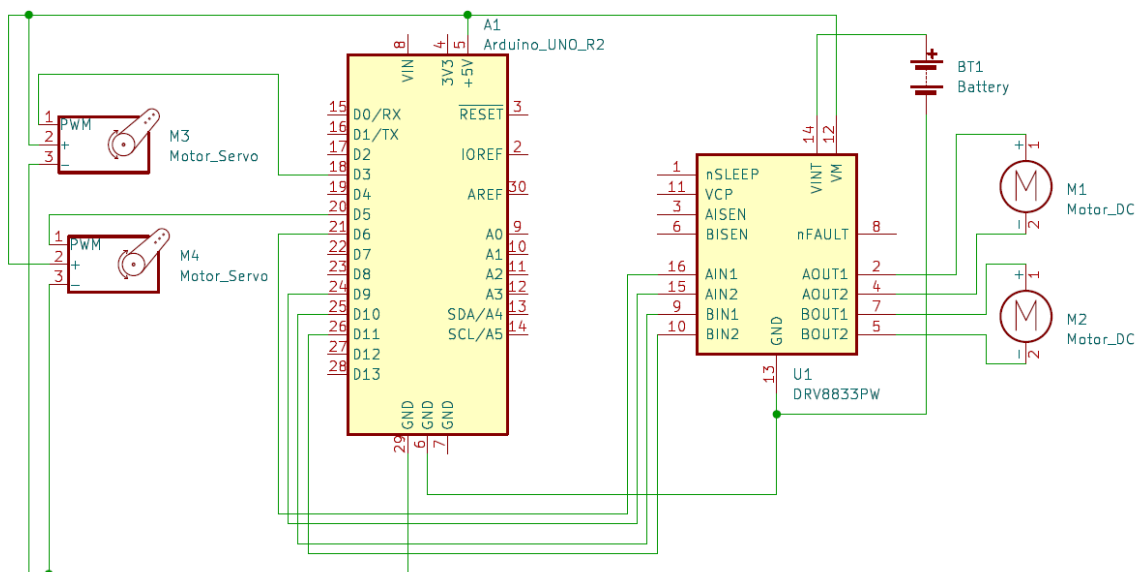
- Pololu 30:1 Micro Metal Gearmotor MP (

<https://www.tme.eu/ro/details/pololu-2364/motoare-dc/pololu/30-1-micro-metal-gearmotor-mp/>) sunt motoare cu perii de curent continuu cu o transmisie de 30:1, utilizate de Ryobo pentru a se deplasa. Acestea sunt mici și compacte, fiind potrivite pentru aplicații embedded. Motoarele sunt disponibile cu sau fără encoder și au o tensiune nominală de 6V.

- DRV8833 (<https://www.pololu.com/product/2130>) este driverul de motoare pe care îl folosesc pentru a controla cele 2 motoare DC, folosind comanda în PWM primită de la microcontroller

- SG90 servo motor (http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf) sunt motoare ușor de controlat care ajustează poziția camerei, prin mișcarea suportului. Comanda pe care o primesc reprezintă unghiul la care trebuie să se rotească pentru a orienta camera către obiectul detectat.

În schema de mai jos care conține designul hardware al robotului (componente, conexiuni), camera nu este surprinsă în imagine. Aceasta comunică cu microcontrollerul prin protocolul de comunicare SPI, datele fiind transmise atât de la ATmega328P către PixyCam, dar și invers.



Software Design

PlatformIO si C/C++

Am ales sa folosesc PlatformIO care este o platformă de dezvoltare software deschisă și gratuită, special concepută pentru proiecte embedded. Aceasta oferă un mediu integrat de dezvoltare (IDE) și un sistem de gestionare a bibliotecilor, care face ușor pentru dezvoltatorii de software să creeze și să gestioneze proiecte embedded. PlatformIO poate fi utilizat cu o varietate de platforme hardware, inclusiv Arduino.

Printre caracteristicile cheie ale PlatformIO se numără:

1. Suport pentru peste 1.000 de platforme hardware diferite
2. Suport pentru multiple limbaje de programare, inclusiv C/C++ (limbajul ales pentru Ryobo)
3. Integrare cu mai multe IDE-uri populare, cum ar fi Visual Studio Code, Atom și Eclipse
4. Posibilitatea de a instala și gestiona biblioteci de software folosind o singură comandă
5. Capacitatea de a genera automat fișiere de configurație pentru proiecte, ceea ce reduce timpul necesar pentru configurarea inițială a proiectelor
6. Suport pentru testare unitară și integrare continuă

În general, PlatformIO este o platformă puternică și flexibilă de dezvoltare software pentru proiecte embedded, care își găsește utilitatea într-o gamă largă de aplicații, de la proiecte hobbyist până la dezvoltarea de produse comerciale.

Librarii

1. Pixy2.h si Pixy2CCC.h sunt bibliotecile folosite pana acum pentru a apela metode specifice camerei (de exemplu: getBlocks(), care returneaza valorile intre care este incadrat obiectul detectat). Sunt folosite si obiecte specifice camerei: signature, m_x, m_y etc. De asemenea, pentru o vizibilitate mai mare si o performanta mai buna in medii in care lumina nu este atat de buna, am folosit metoda setLamp() pentru a aprinde ledurile de la camera.
2. SPI.h pentru comunicatie intre microcontroller si camera.
3. Servo.h pentru crearea obiectelor de tip Servo, folosirea metodelor attach() si write(). Am testat functionarea in 2 moduri: atat cu biblioteca de Servo, cat si fara. La final, am ales implementarea prin care obtineam rezultatele mai bune.

Algoritmi

Prima etapa a algoritmilor a fost sa implementez pe microcontroller un program care modifica unghiul servomotoarelor in functie de pozitia obiectului fata de centrul frameului. Pentru a obtine comanda in PWM, am calculat pozitia cu ajutorul coordonatelor x si y furnizate de camera, iar apoi am aplicat un algoritm care se bazeaza pe controlul proportional. Acesta este primul pas si minimul pentru a indeplini sarcina de urmarire.

Al doilea pas ar fi comandarea motoarelor DC sa porneasca atunci cand inputul de la camera este identic cu proprietatile obiectului cautat. Comanda se realizeaza prin PWM, semnal care ajunge la driver si dicteaza directia de deplasare, in functie de intrarea folosita din driver.

Al treilea pas este urmarirea in spatiu se realizeaza in functie de dimensiunea identificata prin camera a obiectului. Sunt analizati parametrii primiti (atat width, cat si height). Daca obiectul este identificat si este relativ departe (parametrii width si height sunt mici), motoarele pornesc. Altfel, daca obiectul nu este identificat sau daca acesta este foarte aproape de camera (parametrii width si height depasesc un anumit prag setat), motoarele se opresc.

O etapa de rafinare a urmaririi este reprezentata de virajele facute de motoarele DC. Aceasta problema este putin mai dificila si se bazeaza pe testare si setarea anumitor prioritati in program, astfel incat functionarea robotului sa semene cu un FSM (finite-state machine). In cazul de fata, am prioritizat centrarea obiectului pe axa Ox, iar apoi este realizat virajul in spatiu, in functie de comanda (unghiul) servomotorului care realizeaza centrarea pe Ox.

Rezultate Obținute

1. Testarea unitara a servomotoarelor a demonstrat necesitatea unui algoritm de tip PID, mai ales a componentei P, deoarece chiar daca si fara componenta proportionala, urmarirea era realizata, procesul nu era unul cu performante mari in ceea ce priveste urmarirea referintei si stabilizarea. Servomotoarele tindeau sa se reintoarca la pozitia neutra dupa fiecare modificare a unghiului.
2. Dupa adaugarea a doi factori Kp diferiti in functie de plan (pe Ox am uitizat 0.4, iar pe Oy, 0.5), am obtinut o urmarire vizibil imbunatatita.
3. Dupa ce am testat functionalitatea de centrare, am implementat deplasarea in spatiu. Cei mai importanti parametri de modificat au fost dimensiunile width (80) si height (60) ale obiectului vazut de camera.



Concluzii

Ryobo este un prototip in ceea ce priveste conceptul de visual servoing. In acest proiect am dorit sa punctez cateva notiuni mai provocatoare, din mai multe puncte de vedere:

1. **Control**: Pentru a face robotul sa functioneze, am avut nevoie de prioritizarea uneia dintre urmariri, astfel incat sa creez o activitate cat mai determinista. Am folosit un algoritm proportional de control pentru motoare, cu factori alesi experimental.
2. **Machine Learning**: Camera ruleaza un algoritm de detectie bazat pe hue-ul si contrastul din imaginile capturate.
3. **“Economie” in ceea ce priveste componentele**: Am folosit toate cele 6 canale de PWM

disponibile pe ATmega328P si rezultatul este satisfactor, chiar daca pot fi aduse imbunatatiri: cresterea vitezei motoarelor, multithreading etc. Nu am folosit un senzor de distanta pentru a identifica apropierea obiectului, ci am prelucrat software datele de la camera.

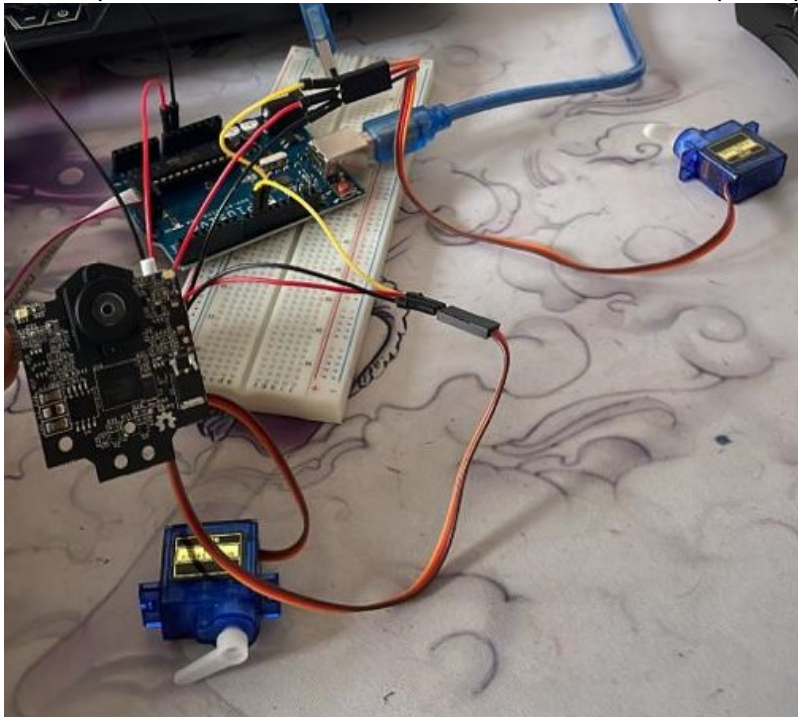
4. **Utilitatea:** Algoritmii de acest gen nu trebuie sa isi gaseasca sfarsitul in implementarea pe roboti miniaturali, ci, odata ce tehnologia avanseaza, pot fi folositi pe drone, in mediul industrial sau pentru securitate.

Download

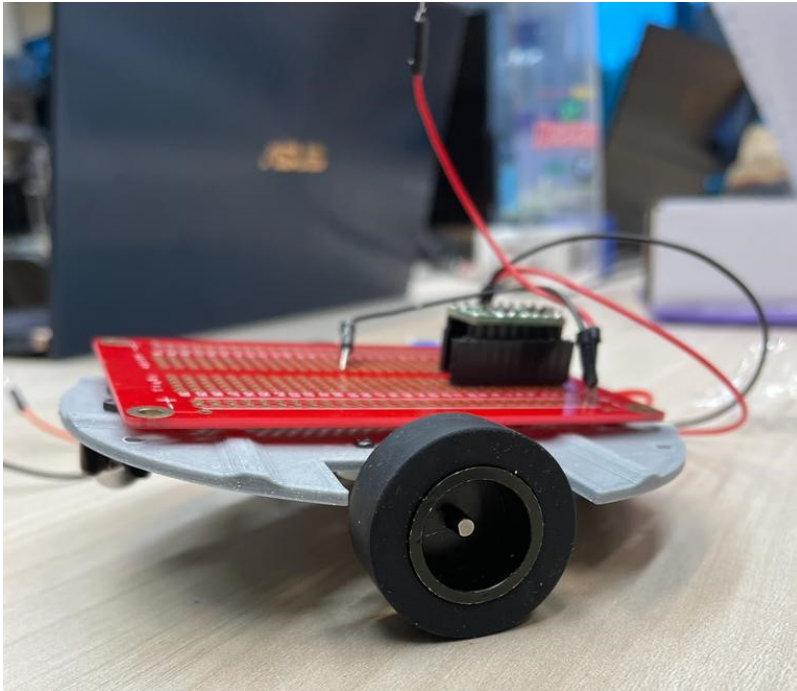
1. Proiectul detaliat poate fi descarcat de aici (inclusiv readme):
<https://gitlab.com/smp2023/331ab/ryobo.git>

Jurnal

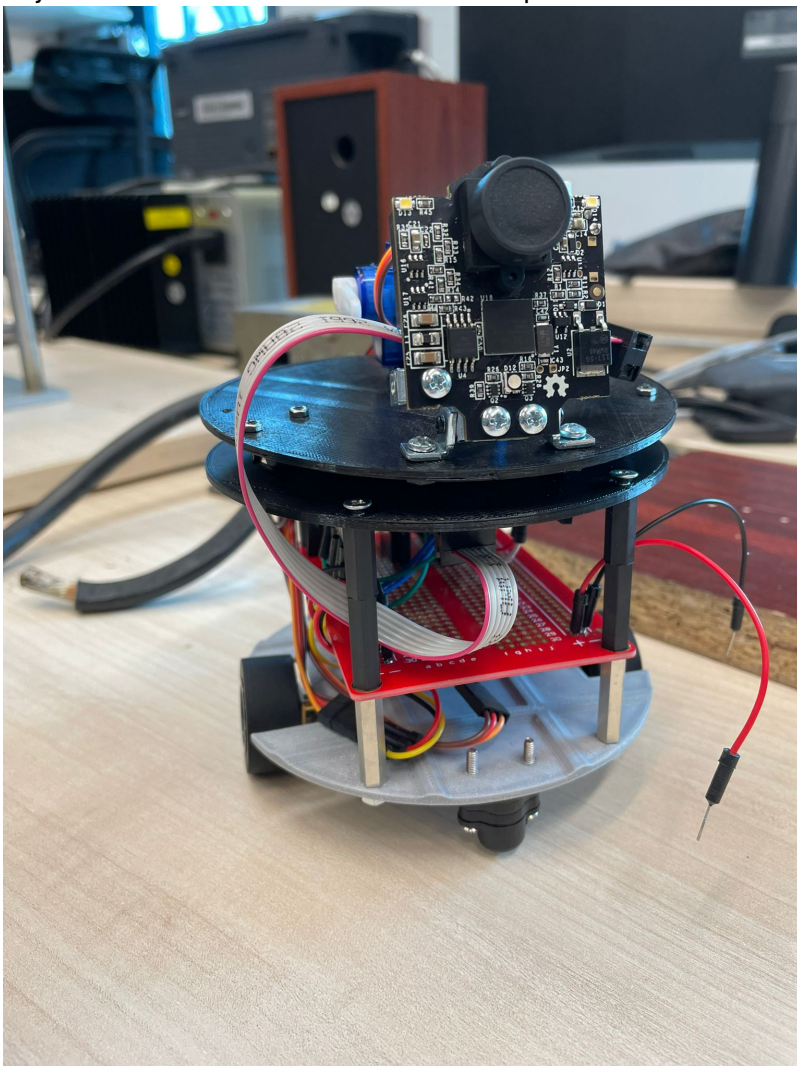
Primul pas a fost controlul servo-urilor in functie de inputul prelucrat de la camera:



Am inceput sa printez piesele din corpul robotului. Aici este primul body:



Pentru a oferi stabilitate tuturor componentelor din interior, inclusiv placii Arduino Uno, am printat si body-ul din mijloc de care este prinsa placa, avand si bratul de servo fixat deasupra. Al treilea body este cel mobil. Pentru a-i asigura stabilitate am folosit 3 ball casters care aluneca pe body-ul din mijloc atunci cand servomotorul misca platforma.



Am identificat existenta unei probleme legate de miscarea camerei in planul Oy. Suportul de care urma sa fie prins bratul servomotorului era la baza camerei. Orice mica ajustare a servomotorului ar fi

miscat camera destul de puternic. Pentru o centrare mai buna, am distantat camera mai mult de platforma mobila si am prins bratul servomotorului la mijlocul sau, nu la baza.



Bibliografie/Resurse

Proiecte similare:

- (<https://cdn-learn.adafruit.com/downloads/pdf/pixy-pet-robot-color-vision-follower-using-pixycam.pdf>)
→ Proiectul Pixy Pet Robot Color Vision Follower este un robot care urmărește un obiect colorat cu ajutorul camerei Pixy Cam 2. Robotul este capabil să detecteze și să urmărească obiecte cu o rată de până la 50 de cadre pe secundă și este controlat de un microcontroller Arduino Leonardo. Față de proiectul meu, acest proiect are mai multe microcontrollere performante, ceea ce face proiectul mai

ușor de implementat, deoarece fiecare poate prelua o funcție specifică. În cadrul proiectului meu, controlul motoarelor va fi mai dificil, deoarece acestea vor fi controlate prin intermediul unei mașini de stare care rulează în bucla infinită de pe ATmega328P.

- (https://youtu.be/DV4YK_Kk5IY) → În acest proiect este prezentat un exemplu de detectare a unei mingi colorate care se mișcă rapid și imprevizibil. Camera așezată pe un suport se orientează după obiectul detectat, fiind mișcată de două servomotoare. Acestea sunt comandate de ATmega328P. Acest proiect nu include o componentă de urmărire la o distanță fixă a obiectului. Nu este un corp robotic complet, ci doar un suport acționat de două servomotoare.

- (https://youtu.be/w_krOCBk1DE) → În acest proiect este construit un robot cu 4 motoare și 4 roți comandate printr-o [punte H L298N](https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf). Camera funcționează ca un senzor pentru a urmări obiectul detectat la o distanță fixă. Acest proiect nu implementează mobilitatea camerei pe axa orizontală și verticală, astfel încât camera să fie mereu orientată către obiect. Ryobo are o complexitate mai mare, deoarece sincronizează toate mișcărilor motoarelor și urmărirea devine mai dificilă.

Biblioteca folosita si descrierea functiilor:

https://docs.pixycam.com/wiki/doku.php?id=wiki:v2:ccc_api

First code Pixy info: <https://sites.google.com/view/jumpinkit/pixy-proyect>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/gpatru/483>



Last update: **2023/05/26 08:57**