

UNIVERSITATEA “POLITEHNICA” DIN TIMISOARA
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL AUTOMATICĂȘI INFORMATICĂ

Robot Tamagotchi

Proiect sincretic

Anul III, AIA, 2024-2025

Vernichescu Cristian

Mihaescu Cristina-Maria

Introducere

În era tehnologiei avansate, domeniul roboticii a devenit unul dintre cele mai fascinante și inovatoare domenii, având aplicații în diverse industrii, de la medicină și agricultură la logistică și explorare spațială. Dezvoltarea roboților autonomi, capabili să perceapă mediul înconjurător și să ia decizii în timp real, reprezintă un pas esențial spre integrarea lor completă în viața cotidiană.

Proiectul de față se încadrează în acest context, explorând utilizarea unui robot autonom pentru o sarcină specifică: detectarea și urmărirea culorii verzi, precum și evitarea culorii roșii. Prin utilizarea tehnologiilor moderne precum ROS2, Ubuntu și Python, proiectul demonstrează cum aceste instrumente pot fi combinate pentru a dezvolta un sistem robotizat capabil să interpreteze informații vizuale și să răspundă în mod adaptiv la acestea.

Prezentarea temei

Tema acestui proiect explorează dezvoltarea unui robot autonom capabil să interpreteze date vizuale și să ia decizii bazate pe culoare. Mai specific, robotul a fost programat pentru a detecta și urmări culoarea verde, interpretând-o ca un obiectiv de interes, și pentru a evita culoarea roșie, considerând-o o zonă de pericol.

Procesarea imaginilor:

Implementată cu ajutorul bibliotecii OpenCV, permite robotului să interpreteze mediul vizual capturat de o cameră. Imaginea este analizată pentru a identifica regiunile corespunzătoare culorilor verde și roșu.

Controlul mișcării robotului:

Prin intermediul ROS2, comenzile de mișcare sunt generate și transmise către sistemul de navigație al robotului, în funcție de poziția culorii detectate în câmpul vizual.

Mediul de dezvoltare:

Sistemul de operare Ubuntu și limbajul de programare Python au fost utilizate pentru a crea un mediu flexibil și eficient, capabil să integreze componentele hardware și software ale robotului.

Tehnologii utilizate

Pentru realizarea acestui proiect, au fost utilizate tehnologii avansate care asigură o integrare eficientă între componentele hardware și software ale robotului.

Robotul utilizat este un sistem autonom echipat cu:

Cameră video pentru capturarea imaginilor din mediul înconjurător.

Unitate de procesare care rulează pe un dispozitiv bazat pe Ubuntu.

Motoare și senzori controlate prin intermediul platformei ROS2, care asigură mișcarea robotului în funcție de datele primite.

Robotul este proiectat să proceseze informațiile vizuale în timp real și să reacționeze în funcție de detecția culorilor verde și roșu.

Mediile de programare

Ubuntu

- Sistem de operare open-source utilizat ca platformă de bază pentru dezvoltarea aplicației robotice.
- Asigură compatibilitatea și stabilitatea necesară pentru funcționarea eficientă a ROS2 și altor biblioteci.

ROS2 (Robot Operating System 2)

- Cadrul software utilizat pentru gestionarea comunicației între componentele robotului.
- Oferă un sistem modular bazat pe noduri, care permite:
 - Capturarea imaginilor de la cameră.
 - Transmiterea comenzilor de mișcare către motoare.
 - Integrarea altor funcționalități avansate.

Python

- Limbajul de programare principal utilizat pentru implementarea algoritmilor de procesare a imaginilor și control al mișcării.
- Biblioteca OpenCV este utilizată pentru:
 - Detectarea culorilor verde și roșu.
 - Calcularea poziției acestora în câmpul vizual.
- Biblioteca ROS2 pentru Python gestionează comunicarea între noduri și publicarea comenzilor robotului.

Schema bloc generală

Fluxul principal al aplicației:

- Capturarea imaginilor
- Procesarea imaginilor
- Decizia comportamentală
- Publicarea comenzilor de mișcare

Ierarhia generală

- Senzor de intrare (Camera)
- Nod Python – Procesarea imaginilor
- Nod Python – Controlul mișcării
- Motoare și unitatea de mișcare: interpretează comenzile și controlează motoarele robotului.

Testare si punere in functiune

Pentru testarea și punerea în funcțiune a proiectului, procesul a inclus verificarea funcționării individuale și integrate a componentelor. În prima etapă, nodurile de procesare a imaginilor și de control al mișcării au fost testate separat, utilizând imagini de test și monitorizând comenzile generate. Ulterior, sistemul complet a fost verificat în mediu controlat, robotul fiind plasat într-un spațiu cu obiecte colorate pentru a valida detectarea culorii verzi, evitarea culorii roșii și reacțiile la condiții variabile, cum ar fi schimbările de lumină.

Pentru instalare, proiectul a fost configurat pe un dispozitiv cu Ubuntu și ROS2. Codul sursă a fost descărcat și compilat, iar parametrii specifici au fost ajustați în funcție de caracteristicile camerei și limitele culorilor. După construirea proiectului, toate nodurile au fost lansate utilizând un fișier de lansare dedicat. Procesul a demonstrat o integrare

eficientă între componentele software și hardware, oferind o bază stabilă pentru extinderea funcționalităților robotului.

Concluzie

Proiectul demonstrează puterea integrării tehnologiilor moderne, precum ROS2, Ubuntu și Python, în dezvoltarea unui robot autonom capabil să interacționeze cu mediul înconjurător. Prin combinarea procesării imaginilor și a algoritmilor de control, robotul reușește să interpreteze datele vizuale și să ia decizii în timp real, cum ar fi deplasarea către culoarea verde și evitarea culorii roșii.

Această realizare evidențiază importanța unei abordări modulare și a utilizării unui ecosistem software robust, permițând nu doar îndeplinirea sarcinilor propuse, ci și extinderea funcționalităților în viitor. Proiectul servește drept exemplu de cum tehnologia poate fi folosită pentru a dezvolta soluții eficiente și scalabile în domeniul roboticii.

Bibliografie

- Quigley, M., Gerkey, B., & Smart, W. D. (2015). *Programming Robots with ROS: A Practical Introduction to the Robot Operating System*. O'Reilly Media
- Molloy, D. (2016). *Exploring Raspberry Pi: Interfacing to the Real World with Embedded Linux*. Wiley
- Open Robotics. *ROS 2 Documentation: Foxy*: <https://docs.ros.org/en/foxy/>
- OpenCV. *OpenCV Documentation*: <https://docs.opencv.org/>
- Ubuntu. *Getting Started with Ubuntu 20.04*: <https://ubuntu.com/tutorials>
- ROS2 Tutorials. *Using ROS2 for Beginners*: <https://index.ros.org/doc/ros2/Tutorials/>