**UNIVERSITATEA „POLITEHNICA” din TIMIȘOARA**

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**

**SPECIALIZAREA AUTOMATICĂ**

Profesor coordonator: Autori:

Ș.l. dr. ing. Sorin NANU Silviu-Dumitru BĂTRÎNUȚ

Daniel-Pavel NEAMȚIU

TIMIȘOARA

2019

Cuprins

[1. INTRODUCERE 4](#_Toc4147021)

[1.1 Contextul aplicației 4](#_Toc4147022)

[1.2 Prezentarea temei 4](#_Toc4147023)

[2. STADIUL ACTUAL AL DEZVOLTARILOR IN DOMENIU 4](#_Toc4147024)

[3. ASPECTE TEORETICE 4](#_Toc4147025)

[3.1 Echipamente hardware 4](#_Toc4147026)

[3.1.1 Arduino Uno 4](#_Toc4147027)

[3.1.2 Pixy CMUcam5 5](#_Toc4147028)

[3.1.3 Biped BRAT kit 6](#_Toc4147029)

[3.2 Tehnologii software 6](#_Toc4147030)

[3.2.1 Arduino IDE 6](#_Toc4147031)

[3.2.2 Limbajul C++ 7](#_Toc4147032)

[3.2.3 PixyMon 7](#_Toc4147033)

[3.2.4 GitHub 8](#_Toc4147034)

[3.2.5 LynxTerm 8](#_Toc4147035)

[4. Arhitectura sistemului 8](#_Toc4147036)

[4.1 Arhitectura hardware şi mecanică 8](#_Toc4147037)

[4.2 Arhitectura software 9](#_Toc4147038)

[4.3 Funcțiile sistemului 9](#_Toc4147039)

[5. Implementare 9](#_Toc4147040)

[6. Rezultate și probleme întâmpinate 10](#_Toc4147041)

[6.1 Rezultate 10](#_Toc4147042)

[6.2 Probleme intampinate 10](#_Toc4147043)

[7. Concluzii și direcții de dezvoltare 10](#_Toc4147044)

[8. Bibliografie 10](#_Toc4147045)

1. INTRODUCERE

1.1 Contextul aplicației

În momentul actual tehnologia se schimbă într-un ritm accelerat. În fiecare an apar paradigme noi, iar inginerii și dezvoltatorii sunt implicați într-un proces de învățare continuă. Marile companii promovează o gamă tot mai variată de unelte și tehnologii.

Dacă până acuma roboții staționari au dominat industria, în ultimii ani s-a remarcat utilitatea roboților mobili. Aceștia, dotați cu o inteligență superioară, vor fi capabili să îndeplinească o multitudine de sarcini, spre deosebire de cei statici, care aveau doar o singură atribuție.

Utilitatea lor este vastă. Ei pot fi găsiți în marile depozite, pe câmpurile de luptă dar și la competiții, în scop recreativ. Noi am hotărât să dezvoltăm un robot din ultima categorie.

1.2 Prezentarea temei

Această lucrare își propune să livreze un robot inteligent de formă umanoidă. Rolul său este de a căuta un obiect de forma sferică într-un spațiu special amenajat, sa se deplaseze inspre el și de a șuta cu unul din membrele sale. Acest comportament este specific modului de lucru automat, dar robotul poate fi controlat si manual de catre utilizator cu ajutorul unei aplicatii.??aici zis ce fel de aplicatie??

Ne-am dorit sa realizam un robot usor de controlat, care sa fie autonom din punct de vedere a logicii si distractiv de urmarit.

2. STADIUL ACTUAL AL DEZVOLTARILOR IN DOMENIU

Robotul, ca definitie, este un dispozitiv reprogramabil care poate inlocui omul pentru anumite sarcini. Robotii umanoizi sunt acei roboti care au forma umana, in special membre, folosite pentru deplasare sau apucare de obiecte.

Termenul de robot a fost formulat pentru prima oara de catre scriitorul ceh de science-fiction Karel Čapek si folosit ulterior de Isaac Asimov. De la stadiul de fictiune robotii au evoluat la realitate pe la mijlocul secolulu 20. Primul robot umanoid a fost completat in 1972 de catre specialistii de la Universitatea Waseda din Japonia, acesta avand capacitatea de a se deplasa.

In ultimii ani programarea robotilor mobili bipezi s-a dezvoltat foarte mult. Marile institute tehnice au investit multi bani pentru realizarea unor roboti cat mai performanti si utili. Robotii actuali au utilitate variata, ca si:

* Asistarea la recuperarea dupa dezastre
* Ingrijirea unei locuinte
* Livrarea de bunuri sau produse
* Utilitate militara

Capacitatile acestor roboti includ:

* Mersul si alergatul pe diferite suprafete
* Mentinerea echilibrului
* Salturi
* Comunicarea cu alte dispozitive

3. ASPECTE TEORETICE

3.1 Echipamente hardware

Pentru elaborarea lucrării de licență, pe partea de hardware s-au folosit următoarele:

* Placa Arduino Uno
* Camera Pixy CMUcam5
* Biped BRAT kit

3.1.1 Arduino Uno

Arduino Uno este un microcontroller bazat pe chipul ATmega328P. Îi este ascociat un mediu de dezvoltare, Arduino IDE, unde se poate programa în limbajul C++. Acesta poate fi alimentat printr-o conexiune USB sau cu ajutorul unei surse externe. Voltajul recomandat de alimentare este situate între 7 și 12 V. La rândul lui, Arduino poate alimenta alte circuite prin pinii speciali de 5V, 3,3V și GND(Ground). Placa este echipată cu:

* Un set de 6 pini analogi și 14 digitali pentru a interfața cu alte circuite
* Capacitatea de a comunica serial(UART TTL, SPI sau I2C)
* 2 surse de întreruperi
* 6 surse de PWM pe 8 biți
* Un LED integrat
* Un bootloader intern

Ca și memorie, ATmega328P deține:

* 32 KB de Flash
* 2 KB de SRAM(Static Random-Access Memory)
* 1 KB de EEPROM(Electrically Erasable Programmable Read-Only Memory)

A circuit board

Description generated with very high confidence

Figura 1: Pinii atașați plăcii Arduino Uno

Ca și alte caractestici, clockul intern are frecvența de 16 MHz, generând un semnal de perioadă 62,5 nanosecunde.

Această placă este compatibilă cu o varietate mare de senzori, traductoare, dar și cu cele mai importante sisteme de operare, dând dovadă de flexibilitate și versatilitate. Luate în considerare toate aceste aspecte reiese că Arduino este ușor de folosit și poate fi utilizată în proiecte care variază de la uz casnic până la uz industrial.

3.1.2 Pixy CMUcam5

Pixy este o cameră cu software/firmware open-source folosită, în special, pentru detecția de obiecte. Printre caracteristicile sale se numără:

* O rată a cadrelor de 50 fps
* Compatibilitate cu plăcile din gama Arduino, Raspberry Pi, BeagleBone
* Compatibilitate cu sistemele de operare Windows, Linux, MacOS
* Capacitatea de comunicare prin SPI, I2C, UART, USB
* Programarea în limbajele C/C++ sau Python

Datorită dimensiunilor sale mici ( 2,1 cm pe 2 cm pe 1,4 cm), CMUcam5 poate fi integrată cu ușurință în majoritatea proiectelor. API-ul său facilitează utilizarea de către toate categoriile de programatori.

3**.1.3** Biped BRAT kit

Acest kit, creat de Lynxmotion, conține :

* Un controller SSC-32
* 6 servo-motoare Hitec HS-422
* Piese prefabricate din aluminiu și șuruburi de diferite dimensiuni

SSC-32(serial servo controller) este un microcontroller special folosit pentru a comanda cu precizie servo-motoare. Se bazează pe microchipul Atmega168-20PU. Deține următoarele capacități:

* Poate comanda până la 32 de motoare
* Poate comunica prin RS232 sau UART TTL
* Rezoluția motoarelor este de 0,09 grade
* Raza de acțiune a motoarelor de 180 de grade
* Un loc pentru memorie EEPROM

Motoarele HS-422 sunt niște servo-motoare cu modulare analogică ce pot fi alimentate cu 4,8V sau 6V. Cu cât este mai mare voltajul cu atât este mai rapidă mișcarea motorului. Ele sunt realizate din plastic și au dimensiuni și greutăți mici, pentru a fi ușor de încorporat în majoritatea proiectelor(4 cm pe 1,96 cm pe 3,66 cm și 45,3 grame). Aria lor de rotație este de 180 de grade în ambele sensuri.

3.2 Tehnologii software

3.2.1 Arduino IDE

Arduino IDE este un mediu de dezvoltare folosită pentru a scrie și încărca programe pe placa Arduino. Ca și capacități se pot număra următoarele:

* Este multiplatformă
* Deține un compilator de C și de C++
* Pot fi incluse numeroase librării
* Are o structură a codului ușor de înțeles și folosit

3.2.2 Limbajul C++

C++ este un limbaj de programare orientat pe obiecte, de nivel înalt dezvoltat de Bjarne Stroustrup ca și o extensie a limbajului C. Apărut pentru prima oară în 1985, C++ este unul dintre cele mai răspândite și de încredere limbaje, fiind folosit de o mare parte de dezvoltatori. În martie 2019, acesta s-a situat pe locul 4 ca și popularitate la nivel mondial.

El este un limbaj flexibil și performant, fiind folosit în diferite domenii, de la jocuri video până la telecomunicații. Poate fi folosit pe cele mai utilizate sisteme de operare(Widnows, Linux, MacOS).

3.2.3 PixyMon

Ca și funcționare, Pixy se bazează pe semnături de culoare. Poate stoca până la 7 semnături diferite și poate recunoaște sute de obiecte. Programatorul setează semnăturile obiectelor de interes. O semnătură poate fi compusă din mai multe culori, după cum se poate vedea în figura 2. În acest caz se detectează un obiect cu o semnătură compusă din 3 culori. Se poate observa interfața mediului PixyMon, compusă din bări de acțiune, imaginile date de cameră, opțiunile de setare de semnături și editare, dar și un spațiu pentru log-uri.



Figura 2: Detecția de obiecte cu ajutorul camerei Pixy

Cu ajutorul mediului PixyMon se pot vizualiza imaginile înregistrate de cameră în 3 moduri:

* Default Program(programul implicit de vizualizare de semnături)
* Raw video(program fără procesare;folosit pentru ajustarea parametrilor camerei, ca și focalizarea sau luminozitatea)
* Cooked video(raw video dar cu un strat de imagine procesată deasupra)

Imaginea utilă este împărțită într-o matrice de pixeli de 320 px pe 200 px, iar procesarea se face pe aceasta.

3.2.4 GitHub

GitHub este un serviciu de hosting și versionare pentru programe software. Acesta poate fi accesat atât de pe web, cât și printr-o aplicație desktop( Windows, Linux, MacOS). Fondat în 2008, GitHub este o unealtă utilă pentru gestionarea de proiecte și este cel mai folosit sistem de hosting din lume.

Este bazat pe sistemul Git, adăugând funcționalități noi, cum ar fi vizualizarea de documente, notificări pe mail sau un sistem de gestionare a erorilor și problemelor întâmpinate.

3.2.5 LynxTerm

LynxTerm este un software realizat special, de catre LynxMotion, pentru testarea functionalitatilor placii SSC-32. Acesta se prezinta sub forma unui terminal serial si a unor bari de actiune, de unde se pot da comenzi motoarelor. Aceasta aplicatie este utila pentru a determina buna functionare a motoarelor.

4. Arhitectura sistemului

4.1 Arhitectura hardware şi mecanică

Kitul BRAT a venit însoţit de un ghid de asamblare online. Cadrul mecanic al robotului a fost realizat prin îmbinarea pieselor prefabricate conform ghidului. Totodată, cu ajutorul acestui document am integrat motoarele şi placa SSC-32. Pentru alimentarea celor menţionate anterior am utilizat o sursă de curent continuu??de inserat modelu de sursa?? pentru a asigura suficientă tensiune şi intensitate electrică.??Eventual pus o poza si vazut ce trebuie pus aici si ce la implementare??

Am stablit şi un spaţiu de lucru pentru robot, unde să îşi desfăşoare mişcările.??eventual pun poza??. Arena are o formă dreptunghiulară, cu pereți pe toate părțile, având dimensiunile 84,5 cm pe 78 cm pe 4,4 cm, realizată din polistiren și de culoare verde, pentru a contrasta cu sfera de culoare roșie. Această hotărâre vine în ajutorul camerei, facilitând detecția. Rolul pereților este de a face ca mingea să ricoșeze din ei, astfel aceasta rămânând constant în interiorul arenei.

Motoarele sunt legate la pinii 0, 1, 2, 16, 17, 18 ai plăcii SSC-32 și corespund articulațiilor robotului conform listei de mai jos, considerându-se spatele robotului partea unde este fixat controllerul SSC-32.

* 16- glezna stângă
* 17- genunchiul stâng
* 18- șoldul stâng
* 0- glezna dreaptă
* 1- genunchiul drept
* 2- șoldul drept

Conexiunile dintre Arduino și SSC-32 sunt următoarele:

* GND la GND
* Tx-ul de la Arduino la Rx-ul de la SSC-32

A screenshot of a cell phone

Description generated with very high confidence

Figura 3: Protocoalele de comunicare folosite în interacțiunea dintre componente

4.2 Arhitectura software

Ca și software, am dezvoltat doar programul ce rulează pe Arduino. Rolul acestui cod este de a prelua datele venite de la camera Pixy, de a le procesa și de a transmite o comandă plăcii SSC-32 în funcție de acestea.

Preluarea de date și prelucrarea lor se face cu ajutorul librăriei Pixy. Prin intermediul API-ului se poate determina:

* Numărul de obiecte detectate
* Coordonatele x și y ale centrului fiecărui obiect
* Lungimea și lățimea fiecărui obiect

Iar prin metoda *print* se pot transmite toate aceste informații pe portul serial.

Pentru a comanda placa SSC-32 este nevoie de a transmite datele într-un anumit format, și anume: #x Py Tz, unde:

* x este pinul atașat motorului ce se dorește a fi comandat
* y este poziția la care trebuie să ajungă membrul comandat de motorul x; 750 ≤ y ≤ 2250 și corespunde unui unghi situat între [0°,180°]
* z este timpul în care să se facă mișcarea, în milisecunde; acest câmp este opțional

Exemplu: #18 P1500 T500

Există și variații ale acestei comenzi.Se poate enumera o listă de motoare și apoi să li se aplice tuturor o poziție și o durată. Exemplu: #18,#17,#16 P1450 T1000

4.3 Funcțiile sistemului

5. Implementare

6. Rezultate și probleme întâmpinate

6.1 Rezultate

6.2 Probleme intampinate

Pe parcursul dezvoltarii lucrarii ne-am lovit de unele dificultati, peste care am trecut cu bine, in final.Acestea sunt:

* Gestionarea cablurilor. Acestea nu erau fixate strans, iar miscarile robotului contribuiau la deconectarea lor. Totodata, ele afectau echilibrul robotului.
* Camera Pixy este sensibila la detectia de obiecte cand lumina ambientala este diferita. In unele cazuri se detectau mai multe obiecte desi doar unul era valid.
* Motoarele au diferite referinte.Teoretic, robotul trebuie sa stea drept cand toate motoarele sunt pe pozitia 1500, dar in practica acest lucru nu se intampla. A trebuit sa gasim pentru fiecare motor referinta sa. Totodata, sensurile de deplasare erau diferite. Daca pentru motorul de la soldul stang 2250 era pozitia maxima in fata, pentru motorul de la genunchiul stang 2250 era pozitia maxima in spate.
* Initial, robotul devia drastic de la traseu si se misca prea brusc, dezechilibrandu-se, iar noi a trebuit sa refacem tot algoritmul de pasit.
* La un moment dat s-a stricat laptopul pe care lucram si unde aveam toate instrumentele, a trebuit sa reinstalam tot si am fost inspirati sa gazduim pe Git toate lucrarile pe care le-am facut.
* A trebuit sa recalculam parametrii functiei de pas deoarece elementele de hardware ce le adaugam pe robot ii afectau echilibrul. Totodata, a trebuit sa optimizam aceasta functie si la schimbarea de suprafata de mers.
* Sursa nu furniza destul curent. Initial, cand robotul nu avea atatea constrangeri folosea maxim 1A, dar pe final ajungea sa consume pana si 1,9A.
* Am incercat sa utilizam alte 2 module de Bluetooth pentru comunicarea intre aplicatia mobile si robot, dar am esuat.
* Se transmiteau caractere ininteligibile de la Arduino la SSC-32 sau unele nu se transmiteau deloc.
* Placa SSC-32 a venit impreuna cu un jumper ce lega pinul Rx de GND, iar acest aspect ne impiedica buna desfasurare a testelor deoarece robotul nu raspundea la nicio comanda.

7. Concluzii și direcții de dezvoltare

7.1 Directii de dezvoltare

Robotul in cauza poate fi imbunatatit si optimizat prin urmatoarele:

* Montarea unei baterii, care sa genereze 6V de tensiune si 2A de curent, dar care sa aibe si dimensiuni reduse pentru a putea fi incorporata cu usurinta in angrenajul robotului

7.2 Concluzii

8. Bibliografie