Samobalansirajući robot na 2 kotača

Blažeka, Valentina

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Polytechnic of Međimurje in Čakovec / Međimursko veleučilište u Čakovcu**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:110:324411

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-07-17**

Repository / Repozitorij:

Polytechnic of Međimurje in Čakovec Repository - Polytechnic of Međimurje Undergraduate and Graduate Theses Repository

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI PRIJEDIPLOMSKI STUDIJ RAČUNARSTVO

Valentina Blažeka

SAMOBALANSIRAJUĆI ROBOT NA 2 KOTAČA

ZAVRŠNI RAD

Čakovec, srpanj 2023.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI PRIJEDIPLOMSKI STUDIJ RAČUNARSTVO

Valentina Blažeka

SAMOBALANSIRAJUĆI ROBOT NA 2 KOTAČA
SELF-BALANCING ROBOT ON 2 WHEELS

ZAVRŠNI RAD

Mentor:
Jurica Trstenjak, v. pred.

Čakovec, srpanj 2023.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
ODBOR ZA ZAVRŠNI RAD

Čakovec, 26. siječnja 2023.

država: **Republika Hrvatska**
Predmet: **Digitalni elektronički sklopovi**

ZAVRŠNI ZADATAK br. 2022-RAČ-R-23

Pristupnik: **Valentina Blažeka (0313025606)**
Studij: Redoviti preddiplomski stručni studij Računarstvo
Smjer: Programsko inženjerstvo

Zadatak: **Samobalansirajući robot na 2 kotača**

Opis zadatka:

Potrebno je izraditi model (maketu) samo balansirajućeg robota. Upravljački dio će biti baziran na Arduino razvojnom okruženju uz korištenje žiroskopa (akceleratora), dok će se za kretanje robota koristiti par istosmjernih motora (opcionalno: koračni motori / step motori). Potrebno je, uz upotrebu Bluetooth modula (dodatka) kontrolirati gibanje robota putem pametnog uređaja. Za sve dijelove upravljačkog sustava je potrebno napisati odgovarajući programski kôd. Samo balansirajući robot će biti napajan pomoću Li-Ion baterijskog modula.

Rok za predaju rada: 20. rujna 2023.

Mentor:

Jurica Trstenjak, v. pred.

Predsjednik povjerenstva za završni ispit:

Sažetak

Cilj je završnoga rada izrada i objašnjenje rada samobalansirajućega robota. Robotom se mora upravljati pomoću izrađene mobilne aplikacije. Komunikaciju između mobitela i robota omogućava bluetooth tehnologija. Glavno je svojstvo samobalansirajućega robota održavanje ravnoteže te brza reakcija i odgovor na vanjske sile. Robot se suprotstavlja sklonosti naginjanju. Robot se mora stalno kretati kako bi održao stabilan položaj. Ako na njega djeluje neka sila, on se mora ponovno postaviti na uravnotežen položaj. Žiroskop je glavna komponenta robota koja može očitati položaj samobalansirajućega robota, a pri tome se upotrebljava princip obrnutoga njihala[1] kako bi se zadržala stabilnost i uravnoteženost. U ovome je radu detaljnije objašnjen princip obrnutoga njihala te načelo PID[2] algoritma. Postoji puno mogućnosti za odabir komponenti, zato je potrebno odabrati odgovarajuće i najprikladnije komponente. Primjer bi bio odabir motora koji će se upotrebljavati. Prilikom izrade robota pozornost se pridaje i odabiru baterija te njihovu položaju. Bitno je odabrati baterije koje pružaju dovoljno snage i koje su dosta izdržljive kako bi mogle održavati ravnotežu robota tijekom duljega razdoblja. Potrebno je razmotriti težinu baterija i to kako utječu na centar gravitacije robota jer su za ovaj rad upotrijebljene četiri baterije. Važna je veličina i prianjanje kotača, vrsta pokretača motora, raspored komponenti uz održavanje težišta i drugo. U procesu izrade samobalansirajućega robota, Arduino IDE paket [1] upotrebljava se za programiranje mikrokontrolera koji upravlja robotom. Osim toga, MIT App Inventor [2] također se upotrebljava za upravljanje robotom pomoću mobilnoga uređaja. MIT App Inventor omogućava jednostavnu izradu mobilnih aplikacija bez složenoga programiranja jer upotrebljava kôd već napisan u blokovima. Za izradu sheme spajanja komponenti upotrijebljen je Fritzing [3]. Tijekom projektiranja robota bilo je važno da materijalni troškovi nisu previsoki. Cilj je ovoga rada i prikazati način na koji je izrađen model samobalansirajućega robota, od odabira komponenti do programiranja i testiranja. Jedan od izazova s kojim se robot suočava može biti osjetljivost na promjene u okruženju, poput neravnoga terena. U tome slučaju robot mora dosta brzo reagirati i prilagoditi se terenu.

Ključne riječi: *samobalansirajući robot, Bluetooth, žiroskop, obrnuto njihalo, PID, Arduino*

[1] Obrnuto njihalo primjer je nestabilnoga sustava koji nije u ravnoteži, a čije je središte mase iznad točke zakretanja. [4]

[2] Prema [5], kontrolni signal PID kontrolera dobiva se zbrajanjem triju komponenti. Prva je proporcionalna vrijednosti signala pogreške, druga je integral signala pogreške, a treća je njegova derivacija.

Sadržaj

1. Uvod

Samobalansirajući roboti dizajnirani su tako da samostalno održavaju ravnotežu, a pri tome se koriste različitim senzorima i kontrolnim algoritmima. Njihova sposobnost održavanja stabilnosti omogućava im da se kreću po različitim terenima, bez obzira na neravnine ili promjene u okolišu.

Jedan je od popularnijih primjera samobalansirajućega robota Segway[3]. Segway je napravljen tako da se balansira na jednome mjestu i ostane u ravnoteži. Osoba se naginje naprijed ako želi da se Segway počinje pomicati unaprijed. Tako se narušava ravnoteža. Segway se mora pomaknuti naprijed kako bi povratio ravnotežu [6]. Projektirana inačica robota manje je složena te se robotom upravlja pomoću pametnoga uređaja.

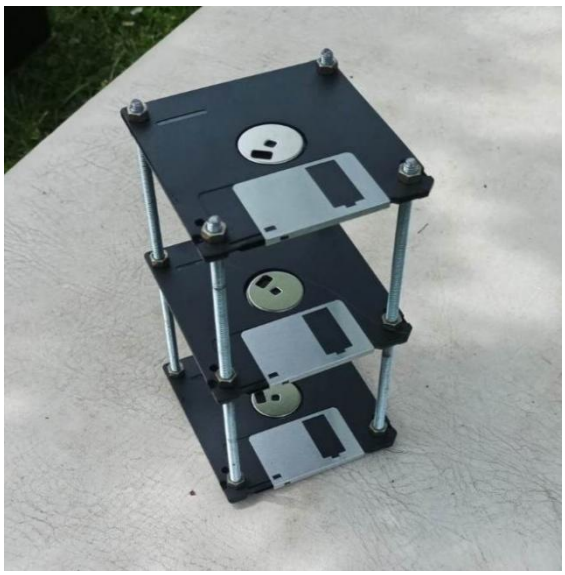
Cilj je ovoga rada napraviti model samobalansirajućega robota uz minimalne materijalne troškove. Potrebno je odabrati odgovarajuće komponente i optimizirati dizajn robota kako bi se postigla stabilnost i učinkovitost uz minimalne resurse. Tijekom izrade modela, velika pozornost posvećena je odabiru odgovarajućega motora, baterije, senzora i ostalih komponenti kako bi se postigao željeni rezultat. Glavni su koraci potrebni za izradu robota mehanički dijelovi, elektroničke komponente i programiranje robota. U radu su objašnjene sve komponente potrebne za rad samobalansirajućega robota te je objašnjena njihova svrha i princip na kojemu rade. Mogućnosti za odabir motora bile su upotreba koračnih step motora ili istosmjernih motora s četkicama. Motori s četkicama imaju mehanički pogon, a motori bez četkica elektronički pogon. Istosmjerni motori jeftini su za kupnju, pouzdani su i mogu se kontrolirati. Nakon objašnjenja sustava, materijala i metoda, prikazan je projektirani robot. Robot je testiran svaki put prilikom spremanja i slanja koda na Arduino pločicu.

[3] Električno vozilo koje se sastoji od dvaju kotača postavljenih jedan pokraj drugoga ispod platforme na kojoj vozač stoji. Upravljano je načinom na koji vozač raspoređuje njihovu težinu. [6]

2. Komponente robota

2.1. Mehanički dio robota

Prilikom projektiranja samobalansirajućega robota, odlučeno je da se za bazu upotrebljavaju samo tri floppy diskete, bez dodavanja vanjskoga kućišta. Ta odluka proizlazi iz potrebe da se smanji ukupna težina robota i omogući bolje održavanje ravnoteže. Floppy diskete kao baze imaju nekoliko prednosti. Diskete su dosta lagane i kompaktne, što je značajno smanjilo težinu robota. To je ključno za postizanje samobalansirajuće funkcionalnosti jer manja ukupna težina olakšava brže i preciznije reakcije na vanjske sile. Također, diskete su čvrste i povezane su vijcima i maticama kako bi se osigurala stabilnost i sigurnost. Ta jednostavna i učinkovita metoda spajanja omogućava stabilnu platformu za postavljanje ostalih komponenti robota poput senzora, mikrokontrolera i motora. Odabir floppy disketa kao baze također nudi praktičnost u smislu dostupnosti i troškovne učinkovitosti. Diskete su široko rasprostranjene i lako dostupne komponente, a njihova upotreba u ovome kontekstu smanjuje potrebu za skupim ili složenim materijalima za izradu vanjskoga kućišta. Iako floppy diskete pružaju stabilnu bazu za komponente robota, bilo je potrebno pozorno prilagoditi raspored komponenti kako bi se održala ravnoteža i centar gravitacije. Prikaz baze robota može se vidjeti na Slici 1.



Slika 1. Baza robota

Izvor: Autor

2.2. Istosmjerni motori

U ovome radu upotrijebljena su dva istosmjerna (engl. *DC - Direct current*) motora s reduktorom (engl. *Gear motor*). Istosmjerni motor s reduktorom jest vrsta električnoga motora koji upotrebljava istosmjernu struju za proizvodnju rotacijskoga gibanja, zupčanici mu omogućavaju povećanje okretnoga momenta i smanjenje brzine [8]. U robotici, ti se motori upotrebljavaju zbog svojega velikog okretnog momenta i niske brzine, što omogućava precizno pozicioniranje i kontrolu. I DC motori s reduktorima i koračni motori imaju svoje prednosti i nedostatke. Prednost je istosmjernih motora ta što s reduktorom mogu raditi stalno, što ih čini prikladnima za primjene koje zahtijevaju stalno kretanje [9].

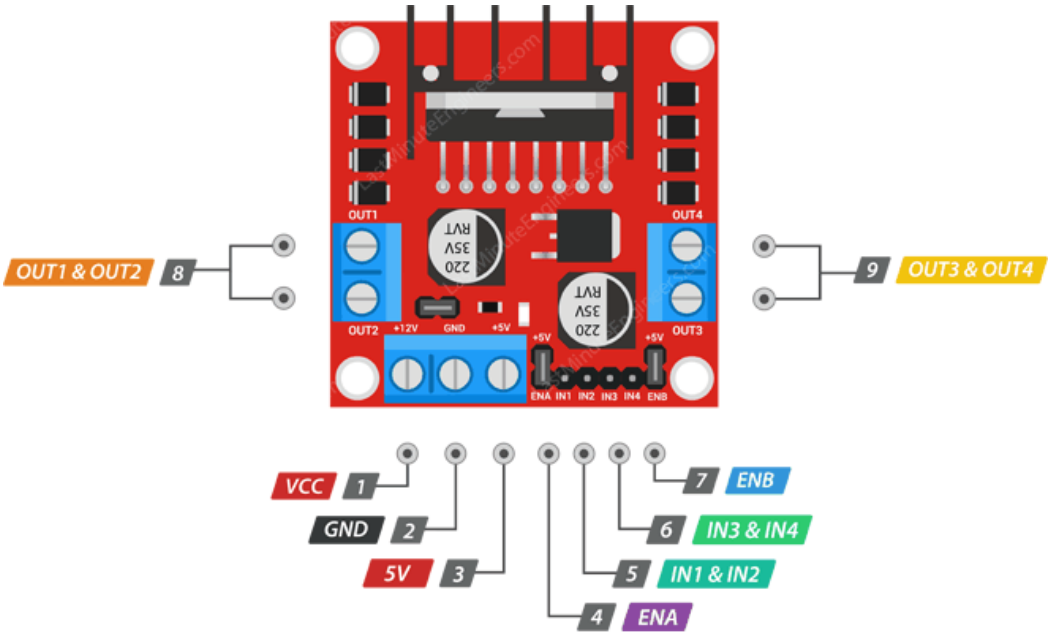
Za ovaj rad upotrijebljeni su motori s reduktorom zbog svoje dostupnosti i cijene, a osigurali su dobru brzinu za kontrolu motora. Smanjenje brzine u motoru može povećati izlazni zakretni moment, omogućavajući robotu da se uravnoteži i napravi male prilagodbe kako bi održao stabilnost. Precizno pozicioniranje i kontrola bitno je za balansiranje robota. Na Slici 2. vidi se DC motor s reduktorom.

Slika 2. DC motor s reduktorom

Izvor: https://m.media-amazon.com/images/I/61l7XUgeN-L._SX522_.jpg (3. 6. 2023.)

2.3. L298N kontroler motora

Najlakši je i najjeftiniji način upravljanja koračnim motorima upotreba motornoga kontrolera L298N. L298N Dual H Bridge Motor Driver jest kontroler motora koji se obično upotrebljava za kontrolu brzine i smjera motora [10]. Čip L298N na sebi ima dva H-mosta[4], koji su elektronički sklopovi koji omogućavaju kontrolu smjera i brzine istosmjernih motora. Ta je funkcionalnost bitna za upravljanje samobalansirajućim robotom s dvama kotačima jer omogućava preciznu kontrolu pokreta i ravnoteže. Prikaz i opis pinova na L298N kontroleru motora vidi se na Slici 3.



Slika 3. L298N Dual H Bridge Motor Driver

Izvor: https://lastminuteengineers.b-cdn.net/wp-content/uploads/arduino/L298N-Motor-Driver-Module-Pinout.png (3. 6. 2023.)

[4] H-most (engl. *bridge*) se sastoji od četiriju sklopki raspoređenih u obliku slova H, s motorom u sredini [10].

Tablica 1. Objašnjenje pinova na L298N Dual H Bridge Motor Driver

Naziv pinova	**Opis pinova**
IN1 & IN2	ulazni pinovi motora A Upotrebljava se za kontrolu smjera vrtnje motora A.
IN3 & IN4	ulazni pinovi motora B Upotrebljava se za kontrolu smjera vrtnje motora B.
ENA	Omogućava PWM [5]signal za motor A.
ENB	Omogućava PWM signal za motor B.
OUT1 & OUT2	izlazni pinovi motora A
OUT3 & OUT4	izlazni pinovi motora B
12V	12 V ulaz iz istosmjernoga izvora napajanja
5V	Napaja sklopove sklopne logičke sklopke unutar L298N.
GND	uzemljenje

Izvor: https://components101.com/modules/l293n-motor-driver-module (4. 6. 2023.)

[5] PWM- (engl. *Pulse-width modulation*) modulacija širine impulse; Prema [14], upotrebljava se za upravljanje analognim uređajevima pomoću digitalnoga signala.

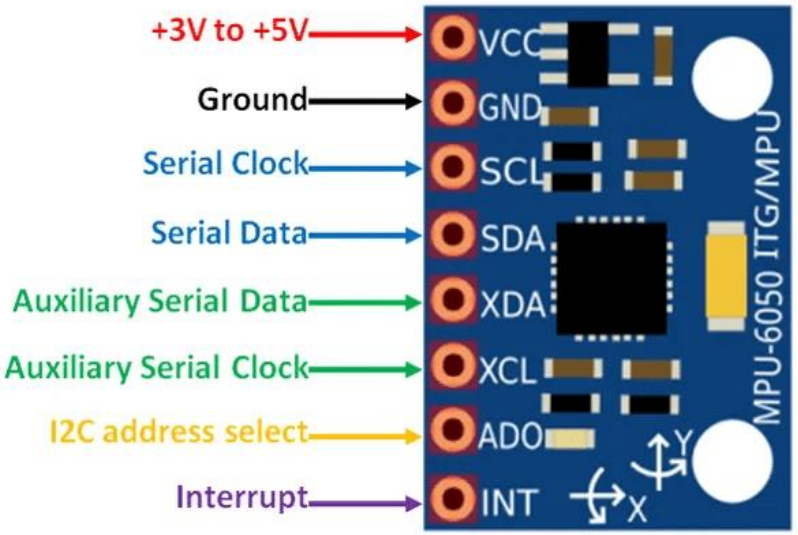
2.4. Senzori

Prema Hrvatskoj enciklopediji, senzor ili mjerno osjetilo jest dio mjernoga sustava. To je uređaj koji vraća izlazni signal, otkriva promjene koje se događaju u njegovoj okolini i šalje informacije o tome drugim elektroničkim uređajima. Većina mjernih osjetila pretvara mjerenu veličinu u električki mjerljiv signal. Kod samobalansirajućega robota upotrebljavat će se akcelerometar i žiroskop.

2.4.1. MPU6050

Najbolji je izbor akcelerometra i žiroskopa za robota MPU6050. MPU6050 je senzor koji u sebi ima 3 osi za žiroskop i 3 osi za akcelerometar koji služi za određivanje položaja robota. Daje tri vrijednosti iz akcelerometra i tri iz žiroskopa. Pomaže u mjerenju brzine, orijentacije, ubrzanja, pomaka i drugih značajki. [11]

Akcelerometar mjeri ubrzanje koje objekt doživljava duž jedne, dviju ili triju osi te pretvara tu informaciju u električni signal. Prema [11], žiroskop primijenjen na MPU6050 koristi se tehnologijom mikroelektromehaničkih sustava (engl. *MEMS*) za stvaranje minijaturnoga rotora koji se okreće na čipu. On mjeri kutnu brzinu oko triju osi. Kutni moment rotora koji se okreće stvara silu koja se odupire promjenama orijentacije, što žiroskopu omogućava mjerenje i održavanje orijentacije. Prikaz MPU6050 i pinovi vide se na Slici 4.



Slika 4. Prikaz pinova na MPU6050

Izvor: https://components101.com/sites/default/files/component_pin/MPU6050-Pinout.png (3. 6. 2023.)

Tablica 2. Objašnjenje pinova na MPU6050

Naziv pinova	**Opis pinova**
Vcc	Osigurava napajanje za modul, može biti od +3V do +5V.
Ground	uzemljenje
Serial Clock (SCL)	Upotrebljava se za davanje taktnoga impulsa za I2C[6] komunikaciju.
Serial Data (SDA)	Upotrebljava se za prijenos podataka putem I2C komunikacije.
Auxiliary Serial Data (XDA)	Može se upotrebljavati za povezivanje drugih I2C modula s MPU6050.
Auxiliary Serial Clock (XCL)	Može se upotrebljavati za povezivanje drugih I2C modula s MPU6050.
AD0	Ako više od jednoga MPU6050 upotrebljava jedan MCU (engl. *Microcontroller*), tada se taj pin može upotrebljavati za promjenu adrese.
Interrupt (INT)	Pin prekida koji označava da su podatci dostupni MCU-u za čitanje.

Izvor: https://components101.com/sensors/mpu6050-module (4. 6. 2023.)

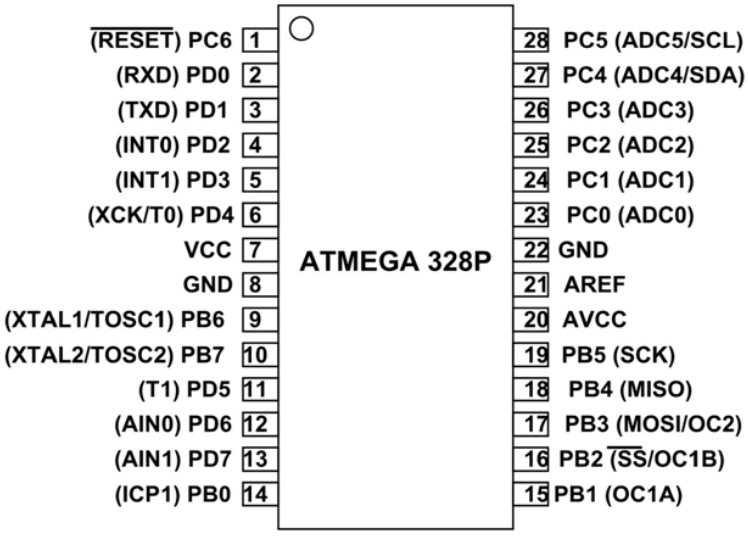
2.5. Arduino Uno

Arduino Uno jest mikrokontroler temeljen na arhitekturi ATmega328P. ATmega328P je 8-bitni mikrokontroler koji ima 32 KB flash ROM memorije [12], a P označava "Picopower", što omogućava da čip radi uz vrlo nisku potrošnju energije. Na Slici 6. vidi se izgled pločice. U nastavku su navedene prednosti Arduino Uno pločice:

- veličina i cijena; Arduino Uno dosta je kompaktan mikrokontroler s malom formom čimbenika, što ga čini pogodnim za ugradnju u različite uređaje i robote. Također, cijena je pločice dosta pristupačna;

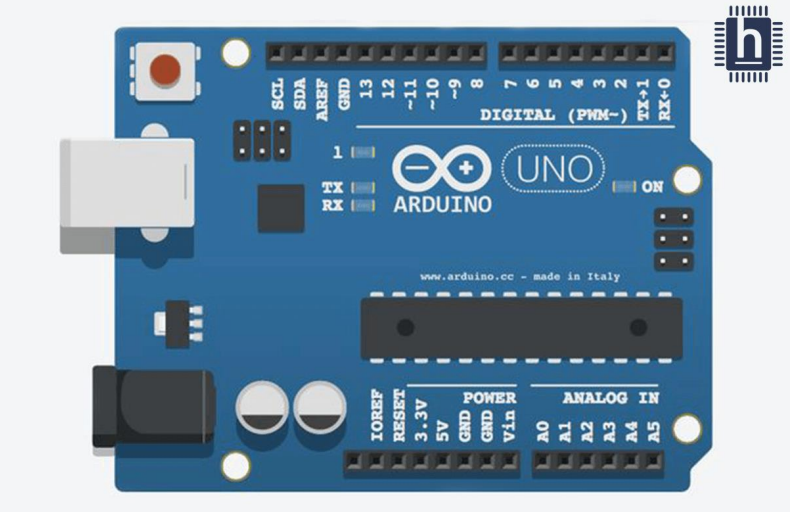
[6] I2C komunikacija - kratica za Inter-Integrirani krug (engl. *Inter-Integrated Circuit*), a to je komunikacijski protokol koji se obično upotrebljava u elektroničkim uređajima za omogućavanje komunikacije između različitih komponenti ili modula.

- vrsta i količina I/O priključaka: Arduino Uno ima dovoljan broj digitalnih i analognih ulaza/izlaza (I/O) kako bi se ostali uređaji mogli spojiti; ti priključci omogućavaju fleksibilnost i mogućnost proširenja funkcionalnosti robota;
- niska potrošnja energije: ATmega328P čip na Arduino Unu ima „Picopower“ tehnologiju koja omogućava rad s vrlo niskom potrošnjom energije. To je važno za samobalansirajućega robota koji bi mogao raditi na baterijsko napajanje, omogućavajući dulje trajanje baterije i produženu autonomiju.



Slika 5. Prikaz nožica na ATMega 328P

Izvor: https://components101.com/sites/default/files/component_pin/ATMega328P-Pinout.png (3. 6. 2023.)



Slika 6. Prikaz Arduino Uno pločice

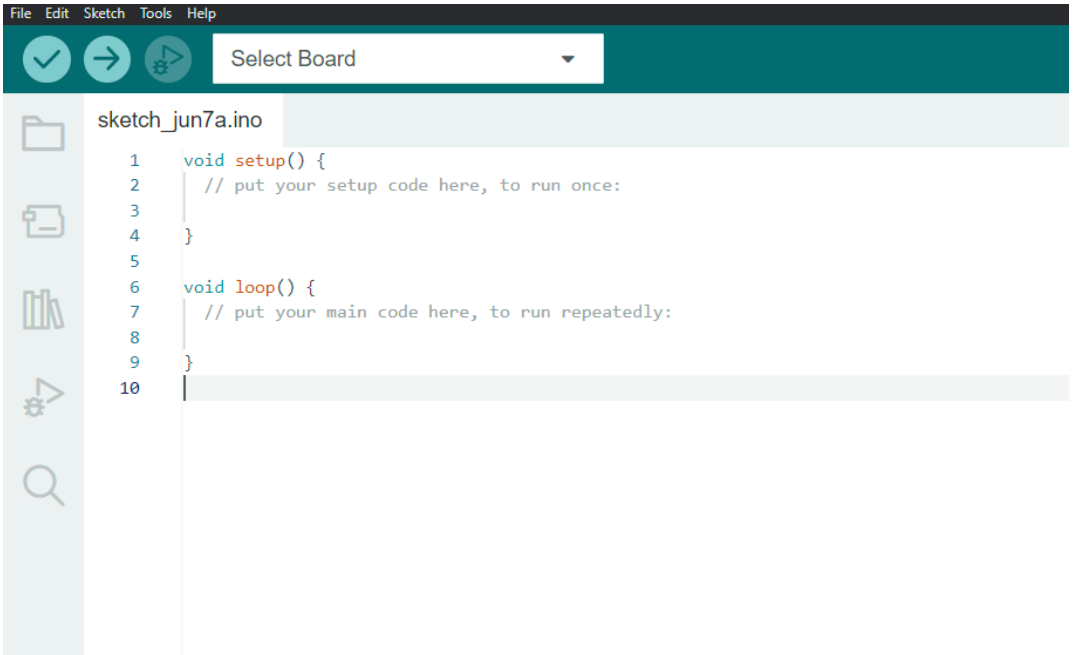
Izvor: https://blog-c7ff.kxcdn.com/blog/wp-content/uploads/2017/02/Blog-Featured-Image.png (3. 6. 2023.)

Karakteristike Arduino Uno nalaze se u nastavku [12]:

- digitalni I/O pinovi: 14 (6 su PMW izlazi)
- analog ulazni pinovi: 6
- radni napon 5 V
- flash memorija: 32 KB
- SRAM: 2 KB
- EEPROM: 1 KB.

2.5.1. Arduino IDE

Prema [1], Arduino integrirano razvojno okruženje (engl. *Arduino Integrated Development Environment- IDE*) jest softverska aplikacija za razvoj, uređivanje i učitavanje kôda na Arduino ploče. Upotrebljava za programiranje i rad s Arduino mikrokontrolerima. Arduino IDE koristi se programskim jezikom temeljenim na C i C++. U program se mogu dodavati različite funkcije i biblioteke za upravljanje radom pločice i ostalih komponenti. Programski se kôd kompajlira (engl. *compile*) i učitava na Arduino ploču pomoću USB veze. Kada se robot priključi na računalo pomoću USB-a, također se može pratiti balansiranje robota pomoću serijskoga monitora i serijskoga crtača. Sljedeća slika prikazuje korisničko sučelje Arduino IDE.



Slika 7. Izgled korisničkoga sučelja na Arduino IDE

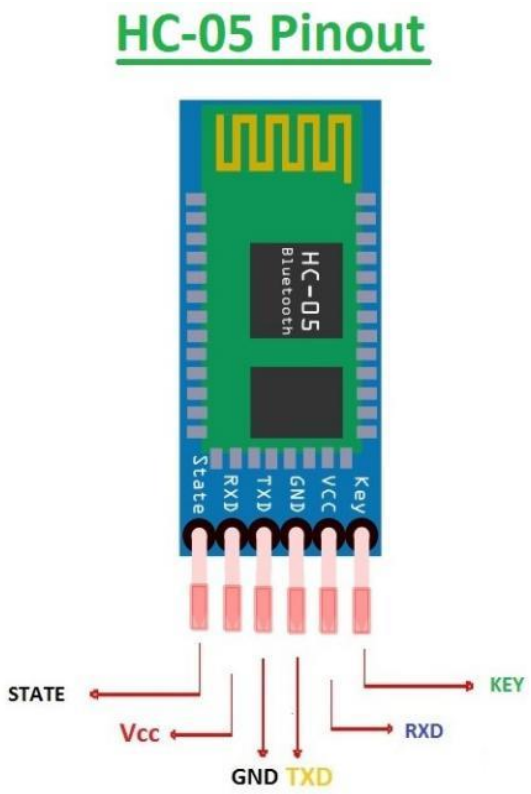
Izvor: Autor

2.6. HC-05

Za ovaj rad upotrijebljen je HC-05 bluetooth modul. Modul je potreban za komunikaciju između mobilnoga uređaja i robota. Prema Hrvatskoj enciklopediji, bluetooth je bežična tehnologija koja omogućava jednostavno povezivanje i prijenos podataka među različitim elektroničkim uređajima na kratkim udaljenostima. U ovome je radu modul upotrijebljen za upravljanje kretnje robota. Izrađena je mobilna aplikacija koja može upravljati robotom. Na Slici 8. prikazan je HC-05.

U nastavku se nalaze karakteristike HC-05 [13]:

- HC-05 modul upotrebljava Bluetooth verziju 2.0
- radi na frekvenciji od 2,4 GHz
- HC-05 modul obično radi na radnome naponu od 3,3 V, ali može podnijeti naponski opseg od 3,6 V do 6 V
- modul podržava snagu prijenosa do 4 dBm.



Slika 8. Prikaz pinova na HC-05

Izvor: https://images.theengineeringprojects.com/image/webp/2019/10/HC-05-Bluetooth-Module-Pinout-Datasheet-Features-Applications-2.jpg.webp?ssl=1 (4. 6. 2023.)

Tablica 3. Objašnjenje pinova na HC-05

Naziv pinova	**Opis pinova**
Enable / Key	Taj se pin upotrebljava za prebacivanje između podatkovnoga i naredbenoga načina.
Vcc	Upotrebljava se za napajanje Bluetooth modula. Daje se 5 V/3,3 V na taj pin.
Ground	uzemljenje
TX – Transmitter	Prenosi serijske podatke. Sve primljeno Bluetoothom taj će pin poslati kao serijske podatke.
RX – Receiver	Primanje serijskih podataka. Svaki serijski podatak poslan tomu pinu bit će emitiran Bluetoothom.
State	Pin stanja spojen je na LED na ploči, može se upotrebljavati kao povratna informacija za provjeru radi li Bluetooth ispravno.

Izvor: https://components101.com/wireless/hc-05-bluetooth-module (4. 6. 2023.)

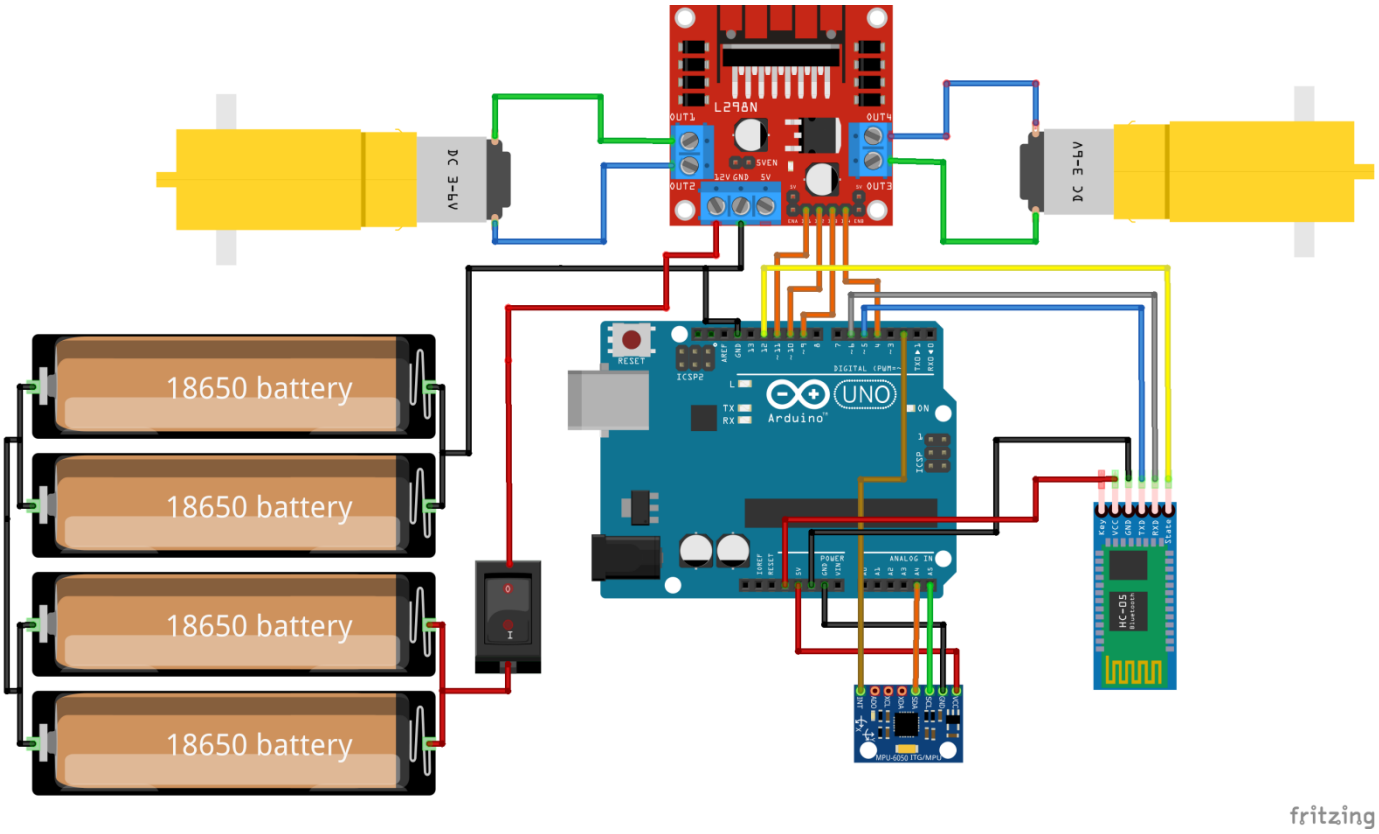
2.7. Napajanje

Za napajanje motora upotrijebljene su 4 ICR 18650 3.7V Li-ion baterije kapaciteta 2950 mAh. Dvije su spojene paralelno, a dvije serijski. Baterije su izvađene iz baterije za laptop A41-X-550E, reciklirane su i ponovno upotrijebljene. Na sljedećoj slici vide se upotrijebljene baterije.

Slika 9. Baterije 18650-26A

Izvor: https://secondlifestorage.com/custom/cell_images/icr1865026a.jpg (8. 6. 2023.)

3. Shema spajanja



Slika 10. Shema spajanja komponenta

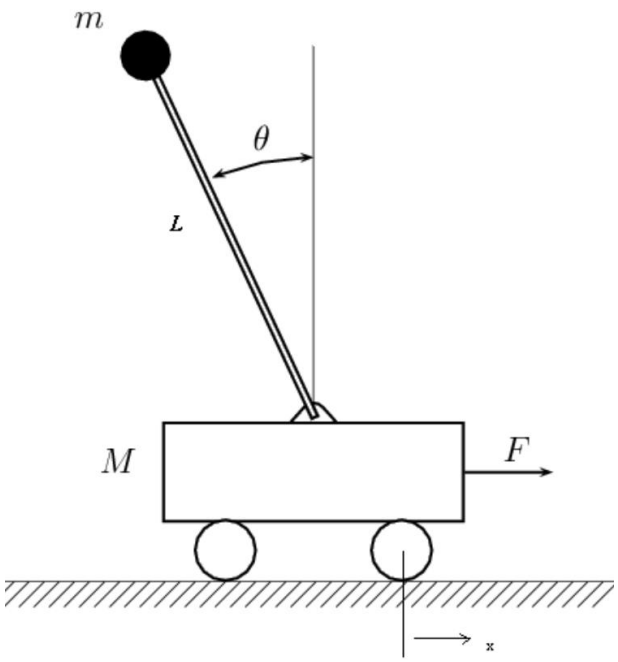
Izvor: Autor

Na Slici 10. prikazana je shema spajanja komponenti koja je izrađena u programskome alatu Fritzing. Alat se upotrebljava za izradu i dizajn elektroničkih sklopova i shema spajanja [3]. Neke od komponenti potrebno je skinuti u datoteke s .fzpz ekstenzijom i staviti u program ako ih nema. Skinuti su modeli baterija, MPU6050, L298N motor driver i motori.

4. Balansiranje

4.1. Obrnuto njihalo

Obrnuto njihalo (engl. *inverted pendulum*) jest sustav koji ima nestabilnu ravnotežu, što znači da ga čak i najmanja promjena uvjeta može dovesti do gubitka ravnoteže. Sustav se sastoji od šipke ili njihala koje je obješeno na točku oko koje se može slobodno njihati u različitim smjerovima [5]. Kada je njihalo uspravno, nalazi se u ravnotežnome stanju. Međutim, zbog djelovanja gravitacije, njihalo će se njihati. Kontrola je inverznoga njihala izazovna jer su za stabilizaciju sustava potrebne precizne i brze reakcije kako bi se održala ravnoteža. Stabilnost obrnutoga njihala ovisi o njegovoj raspodjeli mase i duljine, kao i o visini zakretne točke iznad tla. Ako je točka zakretanja niska, sustav će biti stabilniji, a obrnuto, ako je točka zakretanja visoka, sustav će biti manje stabilan.



Slika 11. Primjer obrnutoga njihala prikazanoga na kolicima

Izvor:https://www.researchgate.net/figure/nverted-Pendulum-Cart-system_fig2_228888952 (7. 6. 2023.)

Oznake su sljedeće:

- **M**- masa kolica
- **m**- masa njihala
- **F**- vanjska sila koja djeluje na kolica
- **x**- pomak kolica
- **θ** [theta] - kut njihala u odnosu na okomitu os
- **L** - duljina između središta osi i središta njihala.

Inercija ili tromost također ima važnu ulogu u određivanju ponašanja njihala. Inercija njihala važan je čimbenik u određivanju njegova ponašanja i odgovora na vanjske sile. Prema Hrvatskoj enciklopediji, inercija je svojstvo tijela kojim se ono odupire promjeni stanja gibanja. Masa njihala, m, predstavlja njegovu tromost i što je njihalo duže, veća je njegova tromost. Njihalo velike mase i velike duljine imat će veću inerciju i bit će ga teže pokretati ili kontrolirati u usporedbi s njihalom manje mase i kraće duljine.

Samobalansirajući robot na dvama kotačima temelji se na principu obrnutoga njihala. Robot se sastoji od tijela robota, odnosno baze s dvama kotačima koja pokreću dva istosmjerna motora. Elektromotori se kontroliraju električnim signalima koje generira Arduino Uno mikrokontroler (PMW) i upravljaju L298N modulom (H-mostovi). Kontrolni signali za upravljanje motorima generiraju se na temelju izmjerenih podataka prikupljenih inercijskom mjernom jedinicom (IMU), specifičnome modelu MPU6050. IMU pruža informacije o nagibu, kutnoj brzini i ubrzanju robota duž osi x, y i z [11]. Podatci za os x upotrebljavaju se za izračunavanje kuta prevrtanja robota. Za uspješno balansiranje robota ključno je imati precizne informacije koje pruža IMU.

4.2. PID regulacija

PID (engl. *Proportional-Integral-Derivative*) je kontrolni algoritam koji se upotrebljava za upravljanje ponašanjem obrnutoga njihala. Svrha je PID algoritma izračunati i primijeniti vrijednost koje omogućavaju njihalu da bude uspravno i spriječiti ga da se prevrne. [4]

Računaju se tri vrijednosti/komponente [5]:

- proporcionalna (P) kontrola: proporcionalna komponenta izračunava signal korekcije na temelju trenutačne pogreške između uspravnoga i stvarnoga stanja njihala. Ta komponenta pomaže da se njihalo približi željenome upravnom stanju. Formula za izračun proporcionalnoga signala jest:

$$u(t) = Kp\ e(t) \qquad (4.2.);$$

gdje je:
- Kp: proporcionalna konstanta (parametar koji se prilagođava)
- e: odstupanje između željene vrijednosti i trenutačne vrijednosti;

- integralna (I) kontrola: integralna komponenta izračunava signal korekcije na temelju akumulirane pogreške tijekom vremena. Ta komponenta pomaže ukloniti sve preostale pogreške koje mogu ostati nakon primjene proporcionalne kontrole. Formula za izračun integracijskoga signala jest:

$$u(t) = Ki \int_0^t e(t)\ dt \qquad (4.3.);$$

gdje je:
- Ki : integralna konstanta (parametar koji se prilagođava)
- e: odstupanje između željene vrijednosti i trenutačne vrijednosti
- dt : vremenski korak;

- kontrola derivacije (D): komponenta derivacije izračunava signal korekcije na temelju brzine promjene pogreške. Ta komponenta pomaže u predviđanju budućih promjena pogreške. Formula za izračun derivacijskoga signala jest:

$$u(t) = Kd\ \frac{de(t)}{dt} \qquad (4.4.);$$

gdje je:
- Kd: derivacijska konstanta (parametar koji se prilagođava)
- de - differential error: diferencijalna promjena odstupanja (promjena pogreške tijekom vremena)
- dt: vremenski korak.

Izlazi iz triju komponenti zbrajaju se kako bi se proizveo jedan signal korekcije, koji se primjenjuje na njihalo za kontrolu njegova gibanja. Vrijednosti PID algoritma potrebno je prilagođavati kako bi se dobili željeni rezultati. Krajnji izlazni signal PID kontrolera dobiva se zbrajanjem komponenti prema sljedećoj formuli:

$$u(t) = Kp\ e(t)\ + Ki\ \int_0^t e(t)\ dt\ +\ Kd\ \frac{de(t)}{dt} \qquad (4.5.);$$

gdje je:
- Kp: proporcionalna konstanta (parametar koji se prilagođava)
- e: odstupanje između željene vrijednosti i trenutačne vrijednosti
- Ki : integralna konstanta (parametar koji se prilagođava)
- dt : vremenski korak
- Kd: derivacijska konstanta (parametar koji se prilagođava)
- de - differential error: diferencijalna promjena odstupanja (promjena pogreške tijekom vremena).

5. Arduino kôd za balansiranje robota

5.1. Korištene biblioteke

Programski kôd preuzet je s internetske stranice [15], međutim dodavane su potrebne biblioteke (engl. *library*) za upravljanje komponentama te je kod prilagođen za Arduino Uno i mojoj shemi spajanja. Mijenjale su se vrijednosti za balansiranje robota poput PID vrijednost, željene vrijednosti nagiba i druge. Također, u kôdu je izmijenjena funkcija za Bluetooth upravljanje jer se upotrebljava druga mobilna aplikacija. U nastavku su objašnjene korištene biblioteke.

```
#include <SoftwareSerial.h>  // Za Bluetooth komunikaciju
#include "I2Cdev.h" // Komunikacija uređaja
#include <PID_v1.h>                    // Biblioteka za PID
#include "MPU6050_6Axis_MotionApps20.h"  // Biblioteka iz jrowberg
#include <Wire.h>       //Komunikacija s I2C/TWI uređajevima
```

Kôd 1. Prikaz korištenih biblioteka

Izvor: Autor

SoftwareSerial - uključivanjem te biblioteke može se upravljati serijskim priključcima na Arduino pločici. Omogućena je komunikacija s vanjskim uređajima. U programskome kôdu upotrijebljena je za uspostavljanje Bluetooth komunikacije.

I2Cdev [16] - omogućava komunikaciju s I2C uređajima, a to su uređaji koji služe za razmjenu podataka mikrokontrolera i drugih uređaja pomoću dvosmjernoga serijskog protokola. U tome programskom kôdu služi za komunikaciju s MPU6050.

PID_v1 [17]- služi za prilagodbu i primjenu PID kontrolera. Pomoću te biblioteke generira se izlazni signal kako bi se sustav mogao balansirati, odnosno doći na željenu vrijednost. U kôdu su prilagođeni parametri PID kontrolera i testirani nakon svake promjene kako bi se postigla željena ravnoteža.

MPU6050_6Axis_MotionApps20 [16] - biblioteka koja služi za upravljanje MPU6050 senzorima. Pomoću te biblioteke dobivaju se vrijednosti o položaju, orijentaciji i kretanju. Omogućava mjerenje akceleracije i brzine okretanja s obzirom na to da iz MPU6050 proizlaze 3 vrijednosti za žiroskop i 3 vrijednosti za akcelerometar.

Wire - omogućuava komunikaciju pomoću I2C sabirnice. Upotrebljava se za inicijalizaciju I2C komunikacije.

5.2. Postavljanje glavnih parametara

Željena vrijednost (engl. *Setpoint*) jest ciljna vrijednost koja se želi postići ili održavati u sustavu. U kontekstu ravnoteže robota, željena vrijednost predstavlja željeni kut nagiba robota koji želimo održavati kako bi robot bio u ravnotežnome stanju. Kada je robot u ravnoteži, njegov je kut nagiba jednak željenomu kutu. Međutim, ako se robot počne naginjati prema naprijed ili prema natrag, kut nagiba promijenit će se. Cilj je PID kontrolera generirati upravljački signal koji će dovesti kut nagiba robota natrag na željenu vrijednost. Željeni kut upotrebljava se kao ulazni parametar u PID kontroler. PID kontroler uspoređuje trenutačni kut nagiba sa željenim kutom te generira odgovarajući izlazni signal koji kontrolira brzinu i smjer motora kako bi se održao ili vratio na željenu vrijednost.

```
//PID vrijednosti podešavanje
double Kp = 31.2;  //proporcionalna engl. Proportional Gain
double Kd = 0.98;  //derivacijska engl. Derivative Gain
double Ki = 190;  //integralna engl. Integral Gain
//ulazne i izlazne vrijednosti
double input, output; //input-predstavlja trenutni kut nagiba
//output-je upravljački signal koji pokreće motore da održe
ravnotežu
// na temelju razlike između input i željeneVrijednosti
PID pid(&input, &output, &zeljenaVrijednost, Kp, Ki, Kd, DIRECT);
```

Kôd 2. Postavljanje PID vrijednosti

Izvor: Autor

Za primjenu toga koda upotrijebljena je vanjska biblioteka „<PID.h>“. Biblioteka podržava dva načina upravljanja: DIRECT i REVERSE. U načinu rada DIRECT, povećanje ulazne vrijednosti dovodi do povećanja izlazne vrijednosti. U REVERSE načinu jest suprotno. U načinu rada DIRECT, PID regulator odgovara na povećanje ulazne vrijednosti povećanjem izlazne vrijednosti. To znači da kada ulazna vrijednost odstupa od zadane vrijednosti, PID regulator prilagodit će izlaz u istome smjeru kako bi vratio sustav prema zadanoj vrijednosti.

U glavnoj funkciji void setup() poziva se funkcija initMPU() koja služi za inicijalizaciju I2C uređaja, odnosno senzora MPU6050. U okviru te funkcije, inicijalizira se komunikacija pomoću I2C protokola. Također se provjerava uspostavljanje veze s MPU6050 senzorom.

```
void initMPU() {
  Serial.println(F("Inicijalizacija I2C uređaja..."));
  Wire.begin();
  mpu.initialize();
  // provjera veze
  Serial.println(F("Testiranje povezanosti uređaja..."));
  Serial.println(mpu.testConnection() ? F("MPU6050 povezivanje
uspješno") : F("MPU6050 povezivanje neuspješno"));
```

Kôd 3. Funkcija za inicijalizaciju I2C uređaja i provjera povezanosti s MPU6050

Izvor: Autor

U glavnoj funkciji postavlja se MPU6050 za korištenje DMP(engl. *Digital Motion Processing*)

```
  devStatus = mpu.dmpInitialize();
```

Kôd 4. Inicijalizacija MPU6050 za korištenje DMP

Izvor: Autor

DMP mu omogućava obradu i izračunavanje orijentacije i kretanja iz senzorskih podataka, akcelerometra i žiroskopa. U slučaju da je inicijalizacija uspješna, varijabla će imati vrijednost '0'.

Unutar glavne petlje void loop() vrši se pokretanje motora. Poziva se funkcija Naprijed() koja omogućava pokretanje motora unaprijed. Funkcija Natrag() okreće motore unatrag, a Zaustavi() zaustavlja motore postavljanjem pinova na vrijednost LOW. Pinovi s kojima se upravlja jesu pinovi na komponenti L298N. Ti pinovi služe za kontrolu motora.

```
//slanje vrijednosti na modul L298N
#define IN1 11
#define IN2 10
#define IN3 9
#define IN4 3
```

Kôd 5. Definiranje pinova za povezivanje s L298N

Izvor: Autor

Jedna je od glavnih funkcija obradaPodataka(). U njoj se nalazi logika prema kojoj se robot balansira, kretanjem naprijed ili natrag. Provjerava se ulazna vrijednost, odnosno kut nagiba robota. Ako je kut između 45 i -45 stupnjeva, robot se kreće naprijed ili natrag. Ako je izlazna vrijednost veća od 0, kreće se naprijed, a ako je manja, kreće se unatrag.

```
void obradaPodataka() {

  if (!dmpSpreman) return;

  while (!mpuInterrupt && fifoCount < velicinaPaketa) {
    pid.Compute();

    //ako je na kutu većim od 45 prestaje raditi
    if (input > -45 && input < 45) {
      if (output > 0)  //output veći od 0
      {
        Naprijed();           //rotacija kotača unaprijed
      } else if (output < 0)  //output manji od 0
      {
        Natrag();  //rotacija kotača natrag
      }
    } else {
      Zaustavi();  //kotači se ne vrte
    }
  }
```

Kôd 6. Funkcija za dohvaćanje i obradu podataka

Izvor: Autor

Unutar petlje loop() također se vrši komunikacija Bluetooth uređajem. Provjerava se dostupnost podataka pomoću te veze. Ako su podatci dostupni, čita se naredba koju se šalje aplikacijom. Naredba koja je pročitana zatim se provjerava pomoću switch bloka. Ovisno o prvome primljenom karakteru, određuje se željeno skretanje motora. Ako je primljeni karakter 'R', postavlja se varijabla skretanje na 'R', što znači da želimo skrenuti udesno. Ako je primljeni karakter 'L', varijabla skretanje postavlja se na 'L', što označava željeno skretanje ulijevo. Ako nijedan od ta dva uvjeta nije zadovoljen, robot se nastavlja samo balansirati bez okretanja. Kako se robot ne bi vrtio u krug, provjerava se ako je već pritisnuta tipka za desno ili lijevo, a ako jest, skraćuje se vrijeme skretanja na 1,5 sekundi. Prikaz kôda koji objašnjava Bluetooth funkciju nalazi se u kôdu 7.

```
void BluetoothUpravljanje(){

  if (Bluetooth.available()) {
    naredba = Bluetooth.readString();
  switch (naredba.charAt(0))
    {
      case 'R': // Right, desno
        if (!isTurning) // Provjeri ako se ne rotira vec
        {
          skretanje = 'R';
          isTurning = true;
          turningStartTime = millis(); // Zabiljezi trenutno vrijeme
        }
        break;
      case 'L': // Left, lijevo
        if (!isTurning) //
        {
          skretanje = 'L';
          isTurning = true;
          turningStartTime = millis();
        }
        break;
      default:
        skretanje = 'P'; // Balansiranje
        break;
    }
  }
  if (isTurning && (millis() - turningStartTime >= turningDuration))
{
    isTurning = false; // Resetiraj
    skretanje = 'P'; // Nastavi balansiranje
    Naprijed();
  }
}
}
```

Kôd 7. Provjera veze s Bluetooth uređajem i čitanje primljenih naredbi

Izvor: Autor

Za prikaz vrijednosti na serijskome monitoru i grafu poziva se funkcija ispisVrijednost().

```
void ispisVrijednost(){
  Serial.print(input);
  Serial.print("\t");
  Serial.println(zeljenaVrijednost);
}
```

Kôd 8. Funkcija za ispis vrijednosti

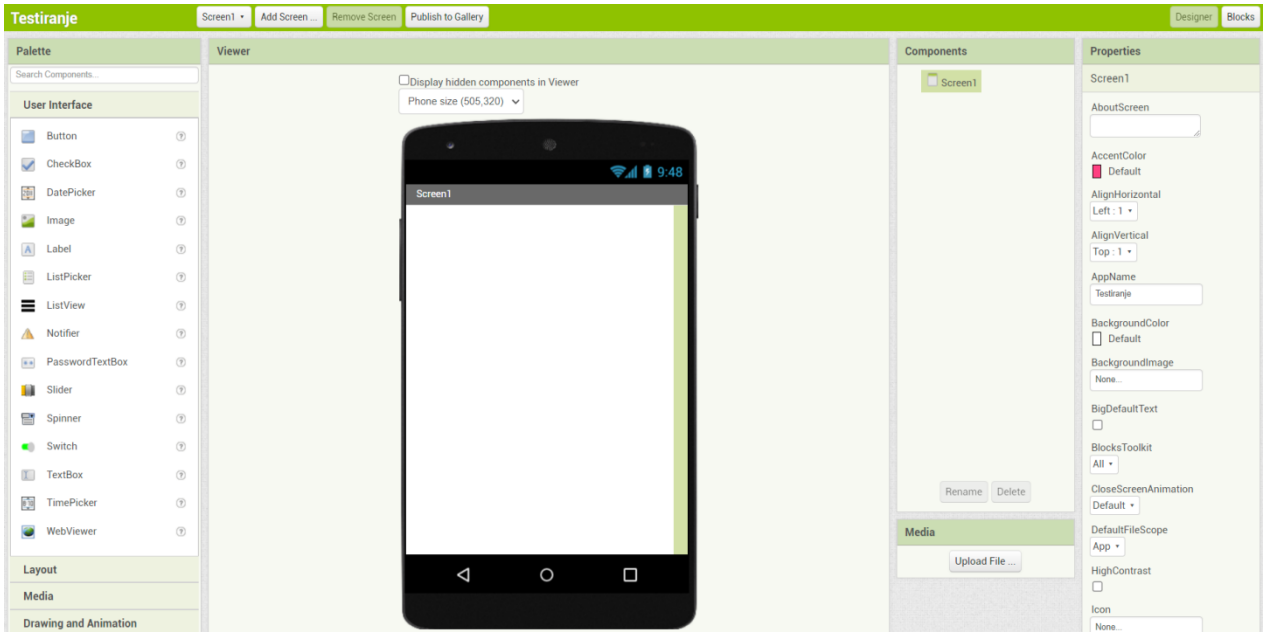
Izvor: Autor

Ostatak Arduino koda nalazi se u prilogu.

6. Izrada aplikacije

6.1. MIT App Inventor

Za izradu aplikacije upotrijebljen je MIT App Inventor, besplatna mrežna platforma koja omogućava brzo razvijanje mobilnih aplikacija. Za razvijanje aplikacija nije potrebno znanje o složenome programiranju. Umjesto programiranja, upotrebljavaju se unaprijed napisani programski kôdovi u blokovima. Linije programskoga kôda nije potrebno pisati. Ti se blokovi mogu jednostavno povlačiti unutar sučelja kako bi se stvorila funkcionalnost aplikacije. Platforma App Inventor nudi skup blokova koji omogućava interakciju s različitim značajkama mobilnih uređaja poput Bluetooth veze, koja je upotrijebljena za oblikovanje te aplikacije. Izgled korisničkoga sučelja vidi se na Slici 12.



Slika 12. Korisničko sučelje na MIT App Inventor

Izvor: Autor

6.2. Opis rada aplikacije

Prilikom ulaska u aplikaciju, kako bi korisnik mogao upravljati kretanjem robota, treba izabrati Bluetooth uređaj na koji se može povezati. Za pronalazak uparenih uređaja na mobitelu moraju biti upaljeni Bluetooth i lokacija. Aplikacija traži od korisnika dopuštenje za skeniranje i povezivanje s Bluetooth uređajem. Kako bi se povezala s robotom, odabire opciju prikaži uređaje i zatim HC-05. Pritiskom na povezivanje, korisnik se može povezati na željeni Bluetooth uređaj. Povezivanje je omogućeno dodavanjem Bluetooth klijenta (engl. *Client*) u aplikaciju. Aplikacija šalje karaktere pritiskom na strelice lijevo, desno, naprijed ili natrag. Primanje i obrada tih karaktera obrađeno je u Arduino kodu. Izgled aplikacije vidi se na Slici 11. Cjelokupni kod aplikacije u blokovima nalazi se u prilogu.

Slika 13. Prikaz aplikacije na Android uređaju

Izvor: Autor

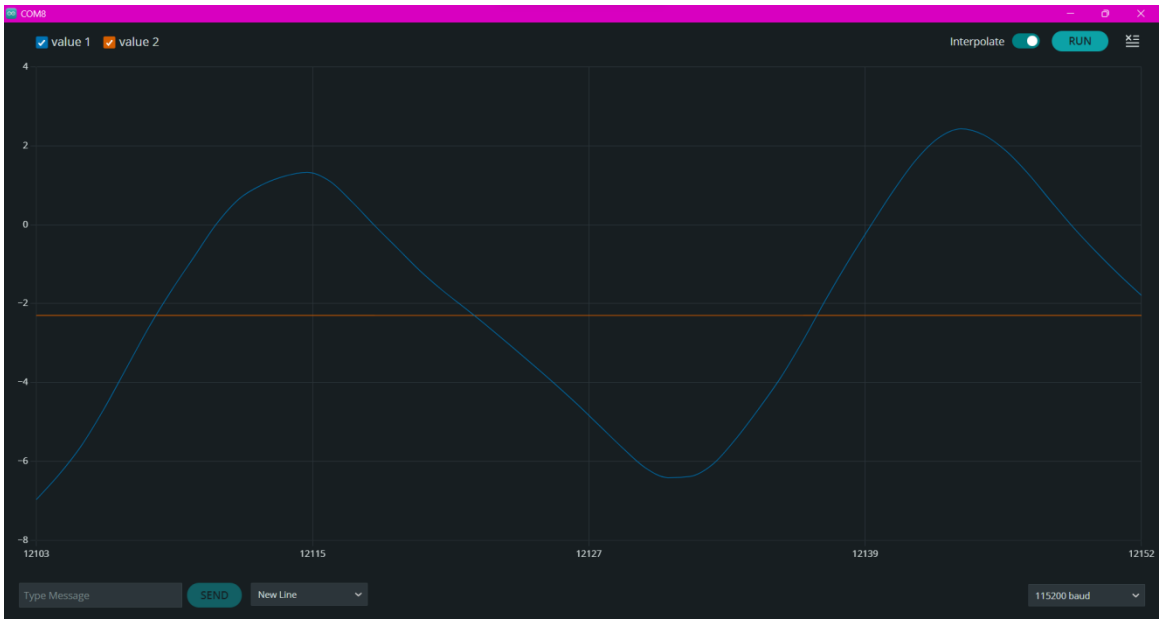
7. Grafički prikaz balansiranoga robota

```
Inicijalizacija I2C uređaja...
Testiranje povezanosti uređaja...
MPU6050 povezivanje uspješno
0.00    -2.30
-0.45   -2.30
-0.90   -2.30
-1.34   -2.30
-1.78   -2.30
-2.22   -2.30
-2.65   -2.30
-3.08   -2.30
-3.52   -2.30
-3.95   -2.30
-4.37   -2.30
-4.80   -2.30
-5.22   -2.30
-5.64   -2.30
-6.05   -2.30
```

Slika 14. Inicijalizacija uređaja

Izvor: Autor

Na Slici 14. može se vidjeti prikaz iz serijskoga monitora (engl. *Serial monitor*). On omogućava praćenje vrijednosti tijekom balansiranja i uvid koliko se vrijednosti razlikuju u milisekundama. Za početak se ispisuje inicijalizacija I2C uređaja i testiranje povezanosti. Ako je MPU6050 ispravno povezan, ispisat će se da je povezivanje uspješno. Prikazane su dvije vrijednosti; prva je ulaz (engl. *Input*), a druga je željena vrijednost (engl. *Setpoint*). Druga se vrijednost ne mijenja jer je to vrijednost koja je postavljena kao željeni kut nagiba. Ulaz se mijenja, a cilj je da ulaz bude što bliži željenoj vrijednosti. Na Slici 15. grafički je prikazano kako se odnosi ulazna vrijednost i željena vrijednost u određenome vremenu.



Slika 15. Grafički prikaz vrijednosti - 1

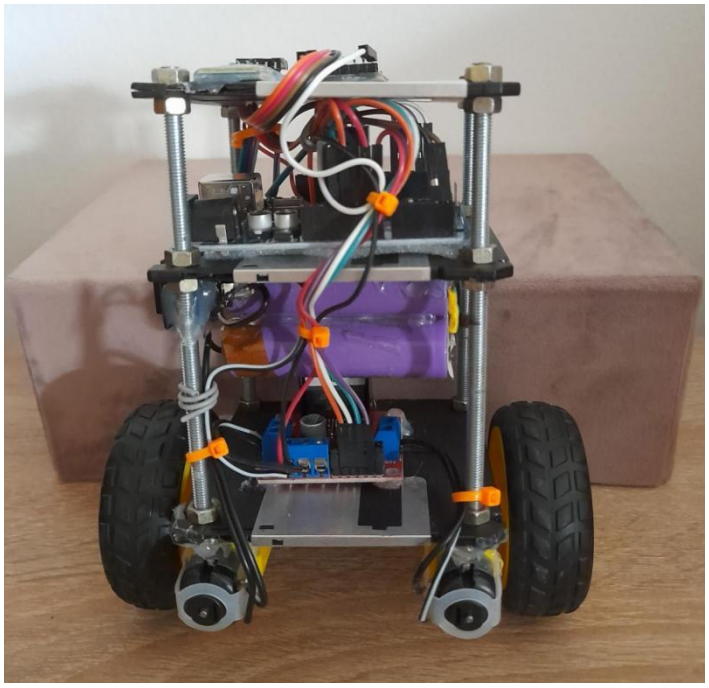
Izvor: Autor

Na Slici 15. prikazane su te dvije vrijednosti, željena vrijednost postavljena je na -2.30° i ulazna vrijednost u vremenu. Za prikaz vrijednosti upotrebljava se serijski crtač (engl. *Serial plotter*). Narančastom bojom označena je željena vrijednost, a plavom ulazna. Na grafu se može vidjeti kako ulazna vrijednost varira od oko -7° do 3°. Iako ulazna vrijednost nije potpuno jednaka željenoj vrijednosti, u tome se stanju robot uspijeva balansirati, ali pri tome lagano oscilira naprijed i natrag. Vrijednosti su dovoljno blizu željenomu kutu nagiba. Na Slici 16. može se vidjeti da je ulazna vrijednost došla na željeni kut nagiba.

Slika 16. Grafički prikaz vrijednosti - 2

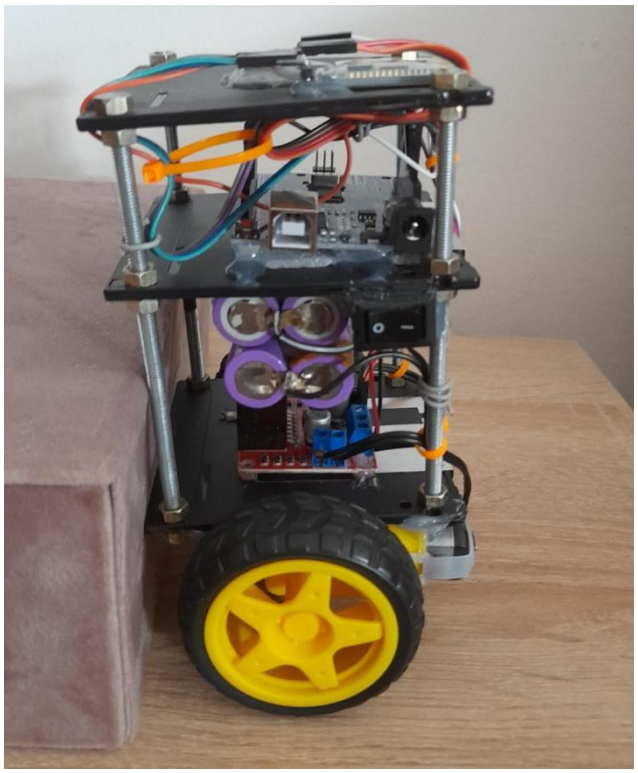
Izvor: Autor

8. Prikaz samobalasirajućega robota



Slika 17. Prikaz robota s prednje strane

Izvor: Autor



Slika 18. Prikaz robota s bočne strane

Izvor: Autor

9. Zaključak

Svrha je i glavni cilj ovoga završnog rada izrada i objašnjenje rada samobalansirajućega robota koji se kontrolira pomoću mobilnoga uređaja. Bluetooth komunikaciju između robota i mobitela omogućava mu komponenta HC-05. Korisnik šalje naredbe pritiskom na strelice koje se zatim pohranjuju i obrađuju pomoću koda napisanoga u Arduino IDE. Naredbe se šalju na Arduino Uno pločicu koja zatim te naredbe šalje na L298N Motor Driver. Pinovi na L298N služe za upravljanje smjerom i brzinom motora te omogućavaju robotu da se kretanje prilagodi primljenim naredbama pomoću Bluetootha. Za robot su se upotrijebila dva istosmjerna motora s reduktorom. Sve njegove komponente zajedno djeluju kako bi samobalansirajući robot održavao ravnotežu i reagirao na korisničke naredbe.

PID regulacija najvažnija mu je za balansiranje. Ona omogućava uspješno održavanje ravnoteže samobalansirajućega robota. PID se koristi izmjerenim vrijednostima nagiba i brzinom rotacije kako bi generirao odgovarajući izlazni signal za upravljanje motorima. Parametri PID kontrolera pravilno se prilagođavaju kako bi se postigla stabilnost i preciznost u održavanju ravnoteže.

Rad samobalansirajućega robota mogao bi biti poboljšan boljom prilagodbom PID parametara. Sve tri vrijednosti potrebno je prilagoditi sustavu kako bi se smanjile oscilacije i poboljšala ravnoteža sustava. Vrijednosti se prilagođavaju kako bi sustav na vrijeme reagirao na promjene. Na bazu robota mogli bi se dodati i novi senzori poput senzora udaljenosti. Pomoću toga senzora robot bi mogao prepoznati prepreke i reagirati prije nego što se sudari. Osim tehničkih poboljšanja, može se razmotriti i poboljšanje cjelokupnoga izgleda robota kao što je dodavanje kućišta, drukčija raspodjela komponenti te smanjene visine ili težine robota za lakše balansiranje.

Popis kratica

PID	*proportional - integral - derivative*	proporcionalan - integralni - derivacijski
IDE	*integrated development environment*	integrirano razvojno okruženje
DC	*direct current*	istosmjerna struja
MPU	*Motion Processing Unit*	jedinica za obradu kretanja
IMU	*inertial measurement unit*	inercijalna mjerna jedinica
MEMS	*Micro-electromechanical systems*	mikro-elektromehanički sustavi
MCU	*Microcontroller*	mikrokontroler
DMP	*Digital Motion Processor*	digitalni procesor kretanja
I2C	*Inter-Integrated Circuit*	inter-integrirani krug
ROM	*Read only memory*	memorija samo za čitanje
I/O	*Input/ Output*	ulaz/ izlaz
PMW	*Pulse-Width-Modulation*	pulsko širinska modulacija
SRAM	*static random access memory*	statička memorija s izravnim pristupom
EEPROM	*Electrically Erasable Programmable Read-Only Memory*	
električno izbrisiva programabilna memorija samo za čitanje		
LED	*light-emitting diode*	svjetlosna dioda
USB	*universal serial bus*	univerzalna serijska sabirnica
Li-ion	*Lithium-ion*	litij-ionska

Popis slika

Popis tablica

Popis kôdova

Popis literature

Korišteni programski alati:

[1] Arduino IDE [Online] 2023. Dostupno na: https://www.arduino.cc/ (28. 4. 2023.)

[2] MIT App Inventor [Online] 2023. Dostupno na: https://appinventor.mit.edu/ (28. 4. 2023.)

[3] Fritzing [Online] 2023. Dostupno na: https://fritzing.org/ (30.5.2023.)

Korištena literatura:

[4] Yon Yaw Lim Choon Lih Hoo , Yen Myan Felicia Wong: Stabilising an Inverted Pendulum with PID Controller. MATEC Web Conf [Elektronički časopis]. 2018. Broj sveska 152. Dostupno na: https://doi.org/10.1051/matecconf/201815202009 (7. 6. 2023.)

[5] Maldini M. - Medium. Objašnjenje PID algoritma. [Online]. 2018. Dostupno na: https://maldus512.medium.com/pid-control-explained-45b671f10bc7 (30. 5. 2023.)

[6] With Segway Tours. Što je Segway. 2022. [Online]. Dostupno na: https://www.whitsundaysegwaytours.com.au/whatisasegway (30. 5. 2023.)

[7] Tutorialspoint. Razlika između koračnog i istosmjernog motora 2022. [Online]. Dostupno na: https://www.tutorialspoint.com/difference-between-stepper-motor-and-dc-motor (30. 5. 2023.)

[8] ISL. Što je istosmjerni motor. 2023. [Online]. Dostupno na: https://islproducts.com/design-note/dc-motor-dc-gear-motor-basics/ (3. 6. 2023.)

[9] Sogears. Karakteristike i primjena istosmjernog motora. 2019. [Online]. Dostupno na: https://hr.sogears.com/blog/karakteristike-i-primjena-istosmjernog-motora (3. 6. 2023.)

[10] Lastminuteengineers. Sučelje L298N DC Motor Driver Module s Arduinom. 2023.[Online]. Dostupno na: https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/ (3. 6. 2023.)

[11] Electronicwings. Mpu6050. 2018. [Online]. Dostupno na: https://www.electronicwings.com/sensors-modules/mpu6050-gyroscope-accelerometer-temperature-sensor-module (3. 6. 2023.)

[12] Arduino. Arduino UNO R3. 2023. [Online]. Dostupno na: https://docs.arduino.cc/hardware/uno-rev3 (3. 6. 2023.)

[13] Components101. HC-05 - Bluetooth Modul. 2021. [Online]. Dostupno na: https://components101.com/wireless/hc-05-bluetooth-module (4. 6. 2023.)

[14] Circuitbread. Što je PWM signal. 2022. [Online]. Dostupno na: https://www.circuitbread.com/ee-faq/what-is-a-pwm-signal (7. 6. 2023.)

[15] Github. Samobalansirajući robot. [Online]. Dostupno na: https://github.com/topics/self-balancing-robot (28. 4. 2023.)

Korištene biblioteke za programski kôd:

[16] Github. i2cdevlib. [Online]. Dostupno na: https://github.com/jrowberg/i2cdevlib/blob/master/Arduino/I2Cdev/I2Cdev.h (28. 4. 2023.)

[17] Github. Arduino-PID-Library. [Online]. Dostupno na: https://github.com/br3ttb/Arduino-PID-Library/blob/master/PID_v1.h (28. 4. 2023.)

Prilog 1. Arduino programski kôd

```
#include <SoftwareSerial.h>  // Za Bluetooth komunikaciju
#include "I2Cdev.h" // Komunikacija uređaja
#include <PID_v1.h>  // Biblioteka za PID
#include "MPU6050_6Axis_MotionApps20.h"  // Biblioteka iz jrowberg
#include <Wire.h>  //Komunikacija s I2C/TWI uređajevima

//slanje vrijednosti na modul L298N
#define IN1 11
#define IN2 10
#define IN3 9
#define IN4 3

MPU6050 mpu;

String naredba;
char skretanje = 'P';  //defaultna vrijednost kretanja

SoftwareSerial Bluetooth(5, 6);  //RX pin, TX pin

boolean BTpovezan = false;

bool isTurning = false;
unsigned long turningStartTime;   //trajanje okretanja
const unsigned long turningDuration = 1200;

const byte BTpin = 12;     // Pin za provjeru stanja Bluetooth veze
bool dmpSpreman = false;  //Digital Motion Processor (DMP), koristi
se za dohvaćanje podataka povezanih s kretanjem, postavi true ako je
success

uint8_t mpuIntStatus;     // sadrži bajt statusa prekida iz MPU-a
uint8_t devStatus;        //status uređaja
uint16_t velicinaPaketa;  // očekivana veličina DMP paketa (zadano
je 42 bajta)
uint16_t fifoCount;       //brojac za bajtove u FIFO
uint8_t fifoBuffer[64];

Quaternion q;         //predstavlja orijentaciju uređaja u
kvaternionskom formatu. Kvaternioni su matematički prikaz rotacija u
trodimenzionalnom prostoru.
VectorFloat gravity;  //predstavlja vektor gravitacije. Sadrži
vrijednosti gravitacijske sile duž X, Y i Z osi.
float ypr[3];         //je niz koji pohranjuje izračunate kutove
skretanja, nagiba i prevrtanja uređaja. Ovi kutovi predstavljaju
rotaciju uređaja oko tri osi. YAW PITCH, ROLL ANGLE


double zeljenaVrijednost = -2.30;  //koristena kao setpoint
//Predstavlja centar stabilnosti koji robot pokušava održati.
//Postavljanjem na -2,30, specificira se željeni kut nagiba (u
stupnjevima) pri kojem se robot treba balansirati.
```

```
//PID vrijednosti podešavanje
double Kp = 31.2;  //proporcionalna engl. Proportional Gain
double Kd = 0.98;  //derivacijska engl. Derivative Gain
double Ki = 190;   //integralna engl. Integral Gain
//ulazne i izlazne vrijednosti
double input, output;  //input-predstavlja trenutni kut nagiba
//output-je upravljački signal koji pokreće motore da održe
ravnotežu
//na temelju razlike između input i željeneVrijednosti
PID pid(&input, &output, &zeljenaVrijednost, Kp, Ki, Kd, DIRECT);

volatile bool mpuInterrupt = false;  // Indikator kada se MPU
Interrupt PIN postavlja u stanje HIGH

void dmpPodaciSpremni() {
  mpuInterrupt = true;
}

void setup() {

  Serial.begin(115200);   //USB baudrate fixirana na 115200 bps
  Bluetooth.begin(9600);  //bluetooth radi na tome baud rate-u

  while (!BTpovezan) {
    if (digitalRead(BTpin) == HIGH) {
      BTpovezan = true;
    }
  }

  initMPU();
  initMotors();
}

void loop() {

  obradaPodataka();
  BluetoothUpravljanje();
  ispisVrijednost();
}


void initMPU() {

  Serial.println(F("Inicijalizacija I2C uređaja..."));
  Wire.begin();
  mpu.initialize();
  // provjera veze
  Serial.println(F("Testiranje povezanosti uređaja..."));
  Serial.println(mpu.testConnection() ? F("MPU6050 povezivanje
uspješno")
                                      : F("MPU6050 povezivanje
neuspješno"));

  //Pokreće DMP (Digital Motion Processor) za dobivanje podataka
senzora
  devStatus = mpu.dmpInitialize();
```

```
  //Default gyrooffset za mpu6050 modul
  mpu.setXGyroOffset(220);
  mpu.setYGyroOffset(76);
  mpu.setZGyroOffset(-85);
  mpu.setZAccelOffset(1788);

  if (devStatus == 0) {
    mpu.setDMPEnabled(true);
    attachInterrupt(0, dmpPodaciSpremni, RISING);
    mpuIntStatus = mpu.getIntStatus();
    dmpSpreman = true;
    velicinaPaketa = mpu.dmpGetFIFOPacketSize();

    pid.SetMode(AUTOMATIC);
    pid.SetSampleTime(10);
    pid.SetOutputLimits(-255, 255);
  }
}

void BluetoothUpravljanje(){

  if (Bluetooth.available()) {
    naredba = Bluetooth.readString();
  switch (naredba.charAt(0))
    {
      case 'R': // Right, desno
        if (!isTurning) // Provjeri ako se ne rotira vec
        {
          skretanje = 'R';
          isTurning = true;
          turningStartTime = millis(); // Zabiljezi trenutno vrijeme
        }
        break;
      case 'L': // Left, lijevo
        if (!isTurning) //
        {
          skretanje = 'L';
          isTurning = true;
          turningStartTime = millis();
        }
        break;
      default:
        skretanje = 'P'; // Balansiranje
        break;
    }
  }

  if (isTurning && (millis() - turningStartTime >= turningDuration))
{

    isTurning = false; // Resetiraj
    skretanje = 'P'; // Nastavi balansiranje
    Naprijed();
  }
}
```

```
void initMotors() {

  pinMode(11, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(3, OUTPUT);

  analogWrite(11, LOW);
  analogWrite(10, LOW);
  analogWrite(9, LOW);
  analogWrite(3, LOW);
}

void obradaPodataka() {

  if (!dmpSpreman) return;

  while (!mpuInterrupt && fifoCount < velicinaPaketa) {
    pid.Compute();

    //ako je na kutu većim od 45 prestaje raditi
    if (input > -45 && input < 45) {
      if (output > 0)  //output veći od 0
      {
        Naprijed();           //rotacija kotača unaprijed
      } else if (output < 0)  //output manji od 0
      {
        Natrag();  //rotacija kotača natrag
      }
    } else {
      Zaustavi();  //kotači se ne vrte
    }
  }

  mpuInterrupt = false;
  mpuIntStatus = mpu.getIntStatus();
  fifoCount = mpu.getFIFOCount();

  if ((mpuIntStatus & 0x10) || fifoCount == 1024) {
    mpu.resetFIFO();
  } else if (mpuIntStatus & 0x02) {
    while (fifoCount < velicinaPaketa) {
      fifoCount = mpu.getFIFOCount();
    }

    mpu.getFIFOBytes(fifoBuffer, velicinaPaketa);
    fifoCount -= velicinaPaketa;
    mpu.dmpGetQuaternion(&q, fifoBuffer);
    mpu.dmpGetGravity(&gravity, &q);
    mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);

    input = ypr[1] * 180 / M_PI;  //vrši se pretvorba u stupnjeve
  }
}
```

```
void Naprijed() {
  if (skretanje == 'R') {
    analogWrite(3, map(output, 0, 70, 0, 60));  // Postepeno povećaj
brzinu motora
    analogWrite(10, 0);
  } else if (skretanje == 'L') {
    analogWrite(3, 0);
    analogWrite(10, map(output, 0, 70, 0, 60));  // Postepeno
povećaj brzinu motora
  } else                                          // Naprijed
  {
    analogWrite(3, output);
    analogWrite(10, output);
  }

  analogWrite(9, 0);
  analogWrite(11, 0);
}

void Natrag() {
  if (skretanje == 'R') {
    analogWrite(11, map(output * -1, 0, 70, 0, 60));  // Postepeno
povećaj brzinu motora
    analogWrite(9, 0);
  } else if (skretanje == 'L') {
    analogWrite(11, 0);
    analogWrite(9, map(output * -1, 0, 70, 0, 60));  // Postepeno
povećaj brzinu motora
  } else {
    analogWrite(9, output * -1);
    analogWrite(11, output * -1);
  }

  analogWrite(3, 0);
  analogWrite(10, 0);
}


void Zaustavi() {
  analogWrite(IN1, 0);
  analogWrite(IN2, 0);
  analogWrite(IN3, 0);
  analogWrite(IN4, 0);
}

void ispisVrijednost() {

  Serial.print(input);
  Serial.print("\t");
  Serial.println(zeljenaVrijednost);
}
```

Prilog 2. Kôd aplikacije izrađene u MIT App Inventor