

# Ultrazvučni senzori u mobilnoj robotici

---

**Bogdan, Luka**

**Undergraduate thesis / Završni rad**

**2016**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Rijeka, Faculty of Humanities and Social Sciences / Sveučilište u Rijeci, Filozofski fakultet u Rijeci**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:186:372134>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-03-12**



*Repository / Repozitorij:*

[Repository of the University of Rijeka, Faculty of Humanities and Social Sciences - FHSSRI Repository](#)



**SVEUČILIŠTE U RIJECI**  
**FILOZOFSKI FAKULTET U RIJECI**

**Odsjek za politehniku**

**Luka Bogdan**

**Ultrazvučni senzori u mobilnoj robotici**  
**(završni rad)**

**Rijeka, 2016. godine**

**SVEUČILIŠTE U RIJECI**

**FILOZOFSKI FAKULTET U RIJECI**

Studijski program: sveučilišni preddiplomski studij politehnike

**Luka Bogdan**

Mat. Broj: 0009060927

## **Ultrazvučni senzori u mobilnoj robotici**

**- završni rad -**

**Mentor: mr.sc. Gordan Đurović**

**Rijeka, 2016. godine**

# FILOZOFSKI FAKULTET U RIJECI

## Odsjek za Politehniku

U Rijeci, 15. travanj 2015. godine

# ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Luka Bogdan**

Studij: **Sveučilišni preddiplomski studij politehnike**

Naslov završnog rada: **Ultrazvučni senzori u mobilnoj robotici**

Znanstveno područje: **2. Tehničke znanosti;**

Znanstveno polje: **2.03. Elektrotehnika;**

Znanstvena grana: **2.03.06 automatizacija i robotika**

Kratak opis zadatka: Objasniti povijesni razvoj i princip rada ultrazvučnih senzora te područja njihov primjene. Prikazati i opisati sve elemente od kojih se sastoji mobilni robot tvrtke RidgeSoft, njihovu međusobnu povezanost i načine korištenja. Detaljno opisati ultrazvučne senzore koji se koriste na mobilnom robotu i njihov način rada. Izraditi program za izbjegavanje prepreka koji će koristiti ultrazvučne senzore mobilnog robota u programskom jeziku JAVA, detaljno opisati sve elemente programa te algoritam kojim se postiže izbjegavanje prepreka.

Izraditi i demonstrirati program za izbjegavanje prepreka korištenjem ultrazvučnih senzora na mobilnom robotu tvrtke RidgeSoft.

Zadatak uručen pristupniku: **15. travnja 2015. godine**

Ovjera prihvatanja završnog rada od strane mentora: \_\_\_\_\_

Završni rad predan: \_\_\_\_\_

Datum obrane završnog rada: \_\_\_\_\_

Članovi ispitnog povjerenstva: 1. predsjednik - \_\_\_\_\_

2. mentor - \_\_\_\_\_

3. član - \_\_\_\_\_

Konačna ocjena: \_\_\_\_\_

Mentor

\_\_\_\_\_  
mr. sc. Gordan Đurović

## **IZJAVA**

Izjavljujem da sam završni rad izradio samostalno, isključivo znanjem stečenim na Odsjeku za politehniku Filozofskoga fakulteta u Rijeci, služeći se navedenim izvorima podataka i uz stručno vodstvo mentora mr.sc. Gordana Đurović.

U Rijeci, 22. svibnja 2016.

## SAŽETAK

Ovim radom se prikazuje princip rada i primjena ultrazvučnih senzora u mobilnoj robotici. Uz to, govori se o drugim područjima njihove primjene te njihov razvoj kroz povijest. Prikazuju se i opisuju svi elementi od kojih se sastoji mobilni robot tvrtke RidgeSoft te njihovu međusobnu povezanost i načine korištenja. Detaljno se opisuju ultrazvučni senzori koji se koriste na mobilnom robotu i njihov način rada te program u programskom jeziku JAVA koji omogućava robotu izbjeći prepreke pomoću ultrazvučnih senzora.

Ključne riječi: Ultrazvučni senzori, ultrazvuk, mobilna robotika, izbjegavanje prepreka, RidgeSoft robot, IntelliBrain, programski jezik JAVA.

U Rijeci, 22. svibnja 2016.

## **SUMMARY**

With this project we show the principles of work and use of ultrasound sensors in mobile robotics. With that it's talked about other areas of their use and their development through history. It is shown and are described all the elements from which the mobile robot from company RidgeSoft is made and their connection with each other and their ways to use. In details are described ultrasound sensors which are used on mobile robot and their ways of working and program in programming language JAVA which allows the robot to avoid obstacles using ultrasound sensors.

**Keywords:** Ultrasound sensors, ultrasound, Mobile robotics, avoiding obstacles, RidgeSoft robot, IntelliBrain, programming language JAVA.

# SADRŽAJ

<b>1. Uvod</b> .....	1
<b>2. Povijesni razvoj i princip rada ultrazvučnih senzora</b> .....	2
2.1 Područja primjene ultrazvučnih senzora.....	3
2.2 Princip rada ultrazvučnih senzora.....	4
2.3 Dobivanje ultrazvuka.....	6
<b>3. IntelliBrain mobilni robot</b> .....	7
3.1 Elementi mobilnog robota .....	7
3.1.1 IntelliBrain 2 kontroler.....	8
3.1.2 Priključci za senzore i motore .....	9
3.1.3 Elementi za komunikaciju .....	10
3.1.4 Servo motori i kotači .....	11
3.1.5 Infracrveni senzori.....	12
3.1.6 Ultrazvučni senzor.....	15
<b>4. Programiranje robota</b> .....	17
4.1 Programska podrška .....	17
4.2 Java programski jezik .....	19
4.3 Program za izbjegavanje prepreka.....	19
<b>5. Zaključak</b> .....	25
<b>6. Popis literature</b> .....	26



## POPIS SLIKA

Slika 2.1. Princip rada ultrazvučnih senzora .....	4
Slika 3.1. IntelliBrain robot.....	7
Slika 3.2. IntelliBrain 2 kontroler .....	8
Slika 3.5. Smjer okretanja servo motora .....	11
Slika 3.6. Napon na izlazu infracrvenih senzora .....	12
Slika 3.7. Izlazni napon infracrvenih senzora s obzirom na položaj kotača .....	13
Slika 3.9. Ultrazvučni senzor .....	15
Slika 4.10. Slučajevi nepreciznog mjerenja .....	16
Slika 4.1. RoboJDE programska podrška .....	17
Slika 4.2. Alati programske podrške .....	18
Slika 4.3. Algoritam za izbjegavanje prepreka .....	24

## POPIS TABLICA

Tablica 2.1. Brzine zvuka u medijima .....	5
Tablica 3.3. Priklučci IntelliBrain 2 kontrolera .....	9
Tablica 3.4. Elementi za komunikaciju .....	10
Tablica 3.8. Stanja linijskih senzora .....	14

## 1. Uvod

Ultrazvučna tehnologija se u posljednjih 50 godina razvila u mnogo različitih tehnoloških grana dok danas najveću primjenu imaju kao parking senzori, uređaji za otkrivanje jata riba te kao alarm sustavi. Veliku primjenu imaju i u industriji te su pogodni za beskontaktno ispitivanje pukotina u cijevima i zavarima. U robotici su pronašli svoju primjenu zbog jednostavne ugradnje i niske cijene, a u mobilnoj robotici su preuzeli ulogu „očiju“ kod robota.

Ovim radom se prikazuje princip rada i način korištenja ultrazvučnih senzora na primjeru edukacijskog robota te međusobna povezanost elemenata robota s programskim kodom.

## 2. Povijesni razvoj i princip rada ultrazvučnih senzora

Kako ultrazvuk slijedi principe opisane u akustici, tako je njegov razvoj utjecao razvojem akustike. Smatra se da je proučavanje akustike započeo grčki filozof Pitagora, čiji su pokusi na svojstva vibrirajućih žica bili toliko popularni da su doveli do ugađanja sustava koji nosi njegovo ime (sonometar), dok je Aristotel pretpostavljao da zvučni val odjekuje u zraku kroz gibanje zraka.

Prvo otkrivanje ultrazvuka bilo je 1790. godine, kada je Lazzaro Spallanzani prvi otkrio da se šišmiši mogu orijentirati i utvrditi položaj lovine u letu pomoću sluha, a ne samo očima.

Jean-Daniel Colladen, švicarski fizičar te inženjer i matematičar Charles-Francois Sturm, 1826. godine koriste podvodno zvono kako bi odredili brzinu zvuka u vodi te pokusom utvrđuju da je brzina zvuka u vodi 1435 metara po sekundi.

Nakon toga, 1880. godine Pierre i Jacques Curie otkrivaju moderne ultrazvučne pretvarače. Otkrivaju odnos između električnog napona i djelovanje vanjske sile na kristal, takozvani piezoelektrični efekt, što je uvelike povećalo razvoj ultrazvučnih uređaja i njegovih primjena u znanosti, medicini, industriji i drugim područjima. Nadalje, Potonuće broda Titanic 1912., potaknulo je interes fizičara u ovom području te kao rezultat istraživanja, Paul Langevin je 1917. izumio hidrofon, mikrofon koji je dizajniran za korištenje pod vodom, za snimanje ili slušanje podvodnog zvuka te tako može detektirati ledenjake pa čak i podmornice.

Iste godine Lewis Fry Richardson podnosi prvi patent uređaja za određivanje udaljenosti pomoći odjeka zvuka u zraku.

Krajem 1930.-ih godina Dr. Karl Dussik koristi tehniku zvanu hiperfonografija, koja bilježi odjeke ultrazvučnih valova na posebno osjetljivom papiru. Ova tehnika se koristi za proizvodnju ultrazvučnih slika mozga za otkrivanje tumora.

Nakon toga, mnogi znanstvenici poput Ian Donald, Douglas Howrya, Joseph Holmes, John Wilda i John Reida, poboljšavaju različite aspekte ultrazvučnih senzora u području medicine koji omogućuju dijagnozu raka želuca, ciste jajnika, otkrivanje trudnoće, tumora.

## 2.1 Područja primjene ultrazvučnih senzora

Najpoznatija i najčešća upotreba ultrazvučnih senzora je u medicini za dijagnostiku unutarnjih organa te pregled i promatranje razvoja fetusa u maternici. Korištenje ultrazvuka je pogodno jer prodire duboko u tijelo, a nema nikakve posljedice po čovjekovo zdravlje. Osim toga, ultrazvuk se može koristiti i u terapijske svrhe takozvane ultrazvučne masaže koje postižu bolje i brže zacjeljivanje ozljeda koštanog tkiva i ligamenata. Također se koristi za otkrivanje tumora.

Ultrazvučni senzori u industriji najviše se koriste za kontrolu kvalitete materijala tako što se ultrazvučni valovi šire kroz homogene materijale i zatim odbijaju na granici materijala različitih akustičnih osobina (otpornosti), odnosno nehomogenosti (grešaka) u materijalu. Prednosti ove metode su: debljina ispitivanog predmeta je neograničena, dovoljan je pristup predmetu kontrole samo s jedne strane, provođenje kontrole je bezopasno i ne zahtijeva zaštitna sredstva, osjetljivost metode je visoka, a pronalaženje pogrešaka jednostavno, metoda je relativno neosjetljiva na uvjete okoline.

Ultrazvučna kontrola se u metalurškoj praksi najčešće koristi kod ispitivanja odljevaka, ispitivanja traka i profila, ispitivanja cijevi, mjerenja debljina stijenki, ispitivanja zavarenih spojeva.

Primjenjuje se i ultrazvučna kontrola električnih uređaja koji omogućavaju ispitivanje spojeva unutar elektro ormara bez otvaranja što bitno utječe na sigurnost ispitivača jer se time sprječava nehotično dodirivanje dijelova pod naponom. Dakle nije potrebno zaustavljati rad postrojenja ili isključivati napajanje električnih uređaja što je obavezno prilikom otvaranja uslijed ispitivanja električnih uređaja u zonama opasnosti. Primjenom infra-crvene termografije mogu se otkriti iskrenja i proboji, ali tek u kasnijoj fazi kada je već došlo do zagrijavanja. Ultrazvučnom metodom možemo uočiti kvar u samoj (inicijalnoj) početnoj fazi.

Ultrazvučni senzori se još koriste: u pomorstvu kao dubinomjer, u ribolovu za otkrivanje jata riba, u vojnu svrhu za otkrivanje podmornica, mina, torpeda.

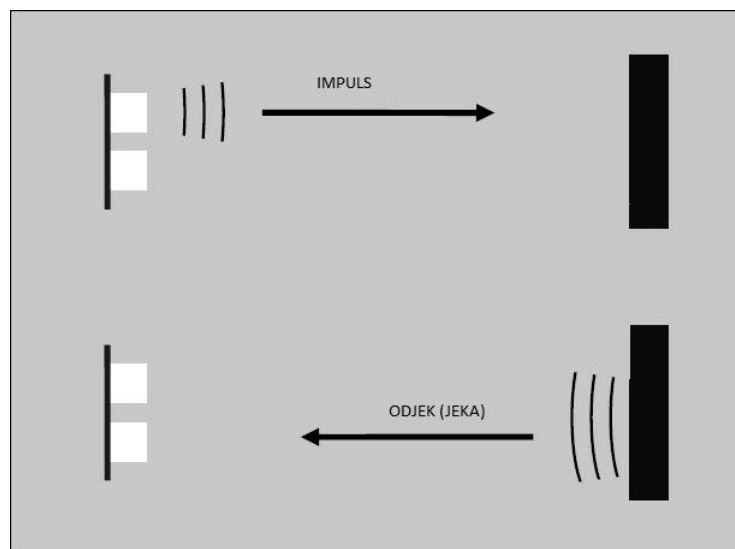
## 2.2 Princip rada ultrazvučnih senzora

Ultrazvuk je zvuk s frekvencijama višim od gornje zvučne granice ljudskog sluha te se ne razlikuje u fizikalnim svojstvima od zvuka kojeg možemo čuti. Gornja zvučna granica ljudskog sluha se razlikuje od osobe do osobe ali uglavnom je to oko 20.000 Hz (Hertza) kod zdravih mladih osoba.

Senzori su uređaji koji mjere i prepoznaju neku fizikalnu veličinu kao što su npr. udaljenost, temperatura, tlak i druge fizikalne veličine.

Dakle, ultrazvučni senzori su senzori koji mjere vrijeme tj. računaju udaljenost od uređaja (senzora) do objekta pomoću ultrazvuka. Takav ultrazvučni senzor se danas još naziva i sonar (što dolazi od akronima riječi eng. SOund Navigation and Ranging).

Ultrazvučni senzori rade na principu generiranja kratkog zvučnog vala (impulsa), a zatim primanja istog koji se odbio od najbližeg objekta, kao što nam prikazuje slika 2.1.



Slika 2.1. Princip rada ultrazvučnih senzora

Ako precizno izračunamo vrijeme od početka generiranja impulsa i njegovog povratka u senzor nakon odbijanja od objekta, udaljenost se može lako izračunati.

Brzina zvuka u zraku je 343 metara po sekundi što je jednako 29.155  $\mu\text{s}$  (mikrosekundi) po centimetru. Kako bi izračunali udaljenost kojom zvuk putuje koristimo sljedeću formulu:

$$Udaljenost [m] = \left( \text{vrijeme [s]} \times \text{brzina zvuka} \left[ \frac{m}{s} \right] \right) \div 2$$

Umnožak vremena i brzine zvuka moramo podijeliti sa dva jer zvuk putuje do objekta i natrag. Ako želimo izračunati udaljenost u centimetrima jednostavnije je koristiti sljedeću formulu:

$$Udaljenost [cm] = (\text{vrijeme} [\mu\text{s}] \div 2) \div 29$$

Naprimjer, ako je potrebno 100  $\mu\text{s}$  da bi se impuls vratio u ultrazvučni senzor, dobijemo da je udaljenost približno 1.7 centimetara.

$$Udaljenost = (100 \div 2) \div 29$$

$$Udaljenost = 1.7 \text{ cm}$$

Brzina zvuka je brzina kojom se širi zvučni val u nekom mediju. Kod krutih medija ovisi o elastičnosti dok kod plinova ovisi o izentropskom (adijabatskom) koeficijentu plina te o njegovoj temperaturi, dok ne ovisi o gustoći i tlaku plina.

Medij	Brzina zvuka na 10°C i atmosferskom tlaku od 10 <sup>5</sup> Pa
Kisik	317 m/s (na 20°C)
Zrak	319 m/s (na -20°C)
Zrak	343 m/s (na 20°C)
Vodik	1270 m/s
Voda	1440 m/s
Željezo	5000 m/s

Tablica 2. Brzine zvuka u medijima

## 2.3 Dobivanje ultrazvuka

Najčešći proizvođač odnosno generator ultrazvuka je kvarcni generator, čiji je rad zasnovan na piezoelektričnom efektu. To je pojava da se na nekim kristalima (kvarc, turmalin i dr.) koji su pogodno odrezani javlja električna polarizacija kada se silom izvrši elastična deformacija. Može se postići i obrnuti efekt kada je primjenjen električni naboj na kristal dolazi do mehaničkih deformacija. Nedostatak kvarcnih izvora je da ne mogu titrati velikim amplitudama, jer se kristal može razoriti. Uz to, frekvencija ultrazvuka ovisi o dimenzijama kristala.

Magnetostrikcija je svojstvo feromagnetskog materijala da mijenja oblik pod utjecajem magnetskog polja (kobalt, nikel, itd.). Magnetostrikcija je odgovorna za zvuk kojega proizvode transformatori i drugi uređaji sa željeznim jezgrama koji rade na izmjeničnu struju. To je slična pojava piezoelektričnom efektu. Fizikalne pojave promjene mehaničkih svojstava ili geometrijskog oblika pod djelovanjem magnetskog polja nazivaju se izravnim magneto-mehaničkim učincima, a pojave kad se mehaničkim naprezanjem mijenja magnetizacija uzorka nazivaju se inverznim magnetomehaničkim učincima.

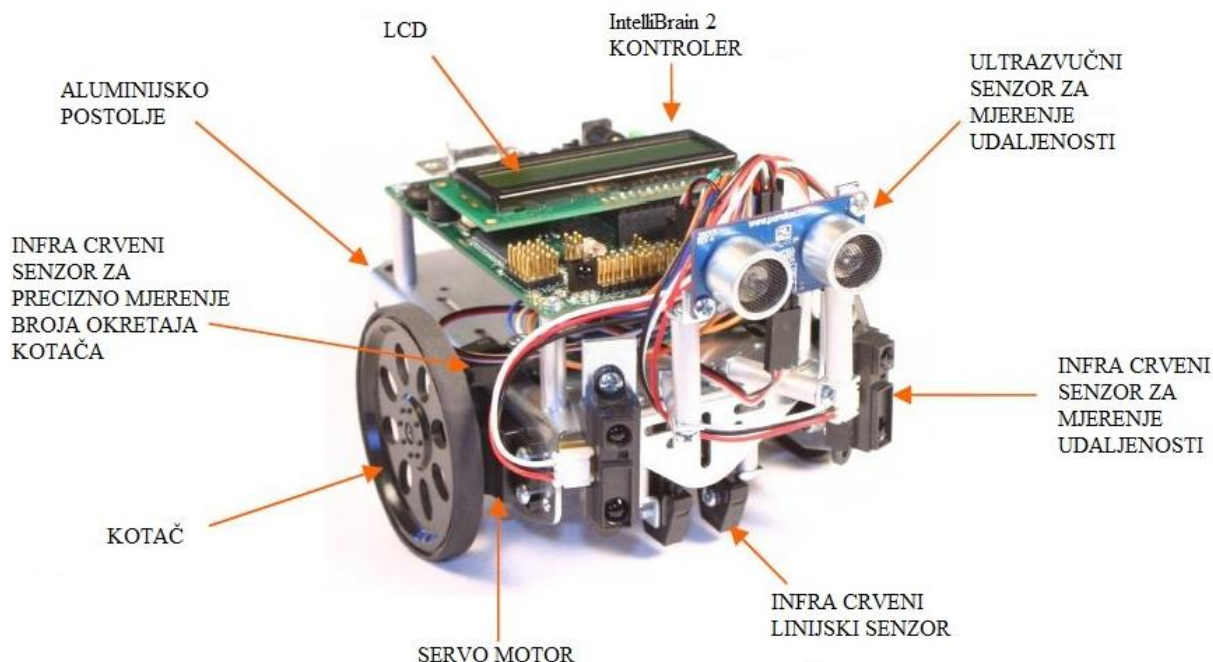


### 3. IntelliBrain mobilni robot

IntelliBrain je edukacijski robot koji je dizajniran kako bi studenti upoznali međusobnu povezanost robotike i programiranja te princip rada njegovih elemenata. Robot koristi nekoliko senzora koji služe za prepoznavanje i izbjegavanje prepreka te navigaciju u prostoru. Robot u potpunosti može biti programiran pomoću JAVA programskog jezika koji se unosi preko računalnog programa RoboJDE.

#### 3.1 Elementi mobilnog robota

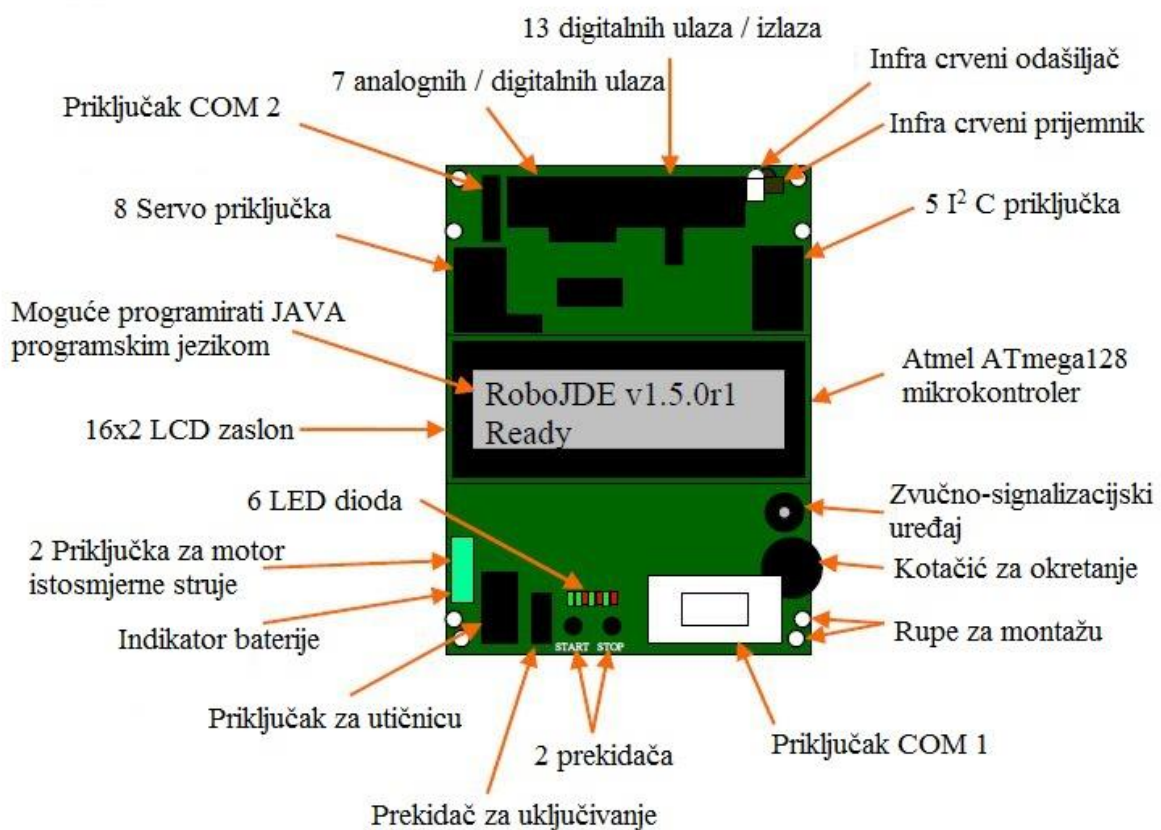
Glavni elementi od kojih se sastoji IntelliBrain robot su: IntelliBrain 2 kontroler s LCD zaslonom, aluminijsko postolje, servo motori, kotači i senzori koje možemo vidjeti na slici 3.1, te baterije i kućište za baterije koje se nalaze ispod aluminijskog postolja. Na stražnjoj strani robota se još nalazi takozvani kotač-loptica koji podupire stražnji kraj.



Slika 3.1. IntelliBrain robot

### 3.1.1 IntelliBrain 2 kontroler

IntelliBrain kontroler povezuje sve elemente robota te preko mikrokontrolera izvršuje naredbe JAVA programa. Program može biti jednostavan koji će samo ispisati naše ime na LCD zaslonu ili kompleksniji kao što je naprimjer program za izbjegavanje prepreka i navigaciju u prostoru. Na slici 3.2 možemo vidjeti gdje su smješteni elementi i priključci na kontroleru.



Slika 3.2. IntelliBrain 2 kontroler

Senzori, servo motori i baterija su spojeni na kontroler preko ulaznih i izlaznih priključaka koji imaju tri ili četiri pinova: uzemljenje, napajanje i pin za signal. Svaki od priključaka ima oznaku koja pokazuje koja je vrsta priključka te njegov broj.

### 3.1.2 Priključci za senzore i motore

U sljedećoj tablici (tablica 3.3) možemo vidjeti čemu služe navedeni priključci kontrolera.

Vrsta priključka	Oznaka	Opis
Analogni	A1-A7	Analogni priključak koristi analogno-digitalni pretvarač koji raspoznaje napone 0 i 5 volti te ih pretvara u cijeli broj od 0 do 1023, gdje 0 predstavlja 0 volti dok 1023 predstavlja 5 volti.
Digitalni	IO1-IO13	Digitalni priključak ovisi o logičkom signalu (uključeno/isključeno, točno ili netočno). Kako ovaj priključak može biti ulaz i izlaz, u oba slučaja 0 volti na priključku znači logički netočno (isključeno), a 5 volti znači uključeno odnosno logički točno, vrijedi i obratno.
Servo	S1-S8	Servo priključci se izravno spajaju na servo motore.
Motor	M1, M2	Motor priključci se izravno spajaju na konvencionalne istosmjerne motore koji koriste modulaciju širine impulsa kako bi mogli upravljati pojedinim motorom promjenom napona.
Serijski	COM 1, COM 2	Serijski COM 1 priključak služi za spajanje robota s računalom kada unosimo programski kod u robota. Serijski priključci također služe za spajanje kompleksnijih senzora kao što su kamere i navigacijski (GPS) uređaji. Spajanjem serijskog Bluetooth adaptera na ovaj priključak, robot može bežično komunicirati sa računalom ili s drugim robotima.
I <sup>2</sup> C	I <sup>2</sup> C	I <sup>2</sup> C priključak služi za spajanje naprednijih senzora poput kompasa ili produkcije ljudskog govora

Tablica 3.3. Priključci IntelliBrain 2 kontrolera

### 3.1.3 Elementi za komunikaciju

Kako bi mogli pokrenuti i izvršiti program mora postojati neki način komunikacije između robota i čovjeka. IntelliBrain 2 kontroler sadrži nekoliko značajki koje omogućuju komunikaciju te upravljanje robota. (Tablica 3.4)

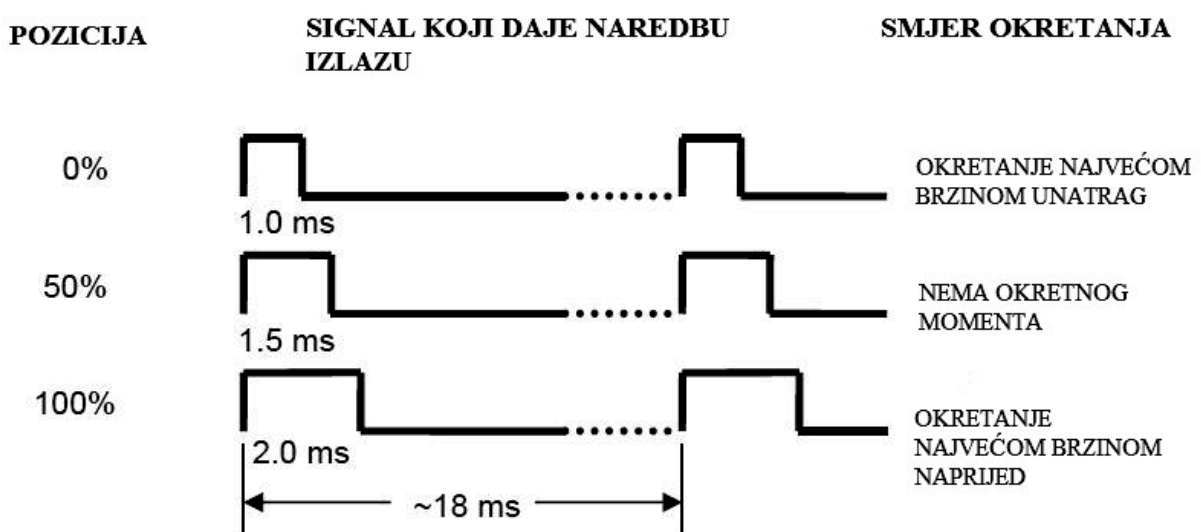
Uređaj	Opis
LCD zaslon	LCD zaslon koji koristimo na robotu ima dva reda s šesnaest stupaca prostora za prikaz znakova i slova. Lcd nam direktno pokazuje očitavanje senzora te nam time olakšava dijagnostiku te greške senzora.
Taster prekidač	Dva taster prekidača, označeni sa START i STOP su prekidači kojima pokrećemo i prekidamo Java program. Mogu se programirati da služe nekoj drugoj svrsi.
Kotačić za okretanje	Kotačić za okretanje radi na istom principu kao kotačić za zvuk kod radija. Može se koristiti za kontrolu brzine servo motora ili kao upravljač za Lcd zaslon.
Zvučno-signalizacijski uređaj	Zvučno-signalizacijski uređaj se može programirati tako da proizvodi kratki signal ili melodiju.
Univerzalni daljinski upravljač	Univerzalni daljinski upravljač omogućuje upravljanje robotom na daljinu.
LED diode	Sedam LED dioda služe kao indikatori. Šest ih se može programirati tako da budu uključene, isključene ili trepere tijekom izvođenja programa. Na robotu imamo tri zelene i tri crvene diode. Prva dioda s lijeve strane je indikator baterije koja svijetli kad je robot uključen te se ne može programirati. Ostalih šest dioda brojimo od 1 do 6, nakon diode indikatora baterije.

Tablica 3.4. Elementi za komunikaciju

### 3.1.4 Servo motori i kotači

Mnogi roboti uključujući i IntelliBrain edukacijski robot, koriste dva nezavisno upravljana servo motora (eng. Hobby servo motors) spojena na kotače koji omogućuju usmjeravanje robota. Servo motori su elektro-motori koji se upravljaju preko njihovog ugrađenog kontrolera pomoću kojeg možemo upravljati brzinom i ubrzanjem motora. Servo motori se najviše razlikuju od ostalih elektro-motora po tome što imaju mogućnost precizne kontrole kutnog položaja i zadržavanja pozicije te povratnu vezu koja javlja računalu informacije o radu te greškama servo motora. IntelliBrain robot koristi Parallax servo motore konstantnog okretaja, ali moraju biti upravljani putem kontrolera servo motora. Kod servo motora konstantnog okretaja, kontroler ima mogućnost upravljati napajanjem i smjerom okretanja.

Servo motori se upravljaju na principu periodičnog pulsiranja signala (slika 3.5). Taj signal je obično bijela ili žuta žica spojena na servo motor. Princip možemo objasniti kao da uključujemo i isključujemo napajanje u kratkom vremenu. Kod ovog motora, jedna milisekunda daje naredbu servo motoru poziciju 0%, što znači okretanje osovine najvećom brzinom u jednom smjeru (npr. u smjeru kazaljke na satu), dok dvije milisekunde daju naredbu motoru poziciju 100%, što znači okretanje osovine motora u drugom smjeru (u suprotnom smjeru od kazaljke na satu). Ako dajemo servo motoru naredbu pozicije 50%, na izlazu servo motora neće biti okretnog momenta.

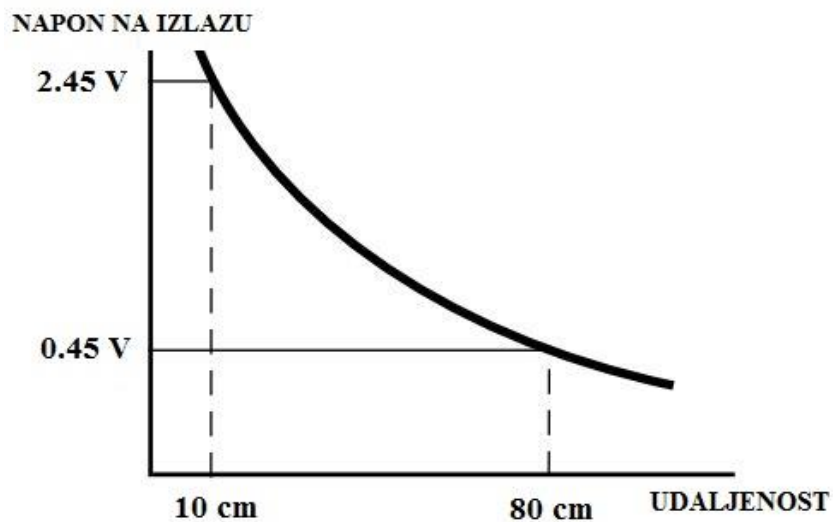


Slika 3.5. Smjer okretanja servo motora

### 3.1.5 Infracrveni senzori

Na IntelliBrain robotu možemo primjetiti dvije vrste senzora koji rade na principu infracrvene svjetlosti, a to su: dva senzora za mjerenje udaljenosti, dva linijska senzora i dva senzora za brojanje okretaja kotača koji su iste vrste kao i linijski.

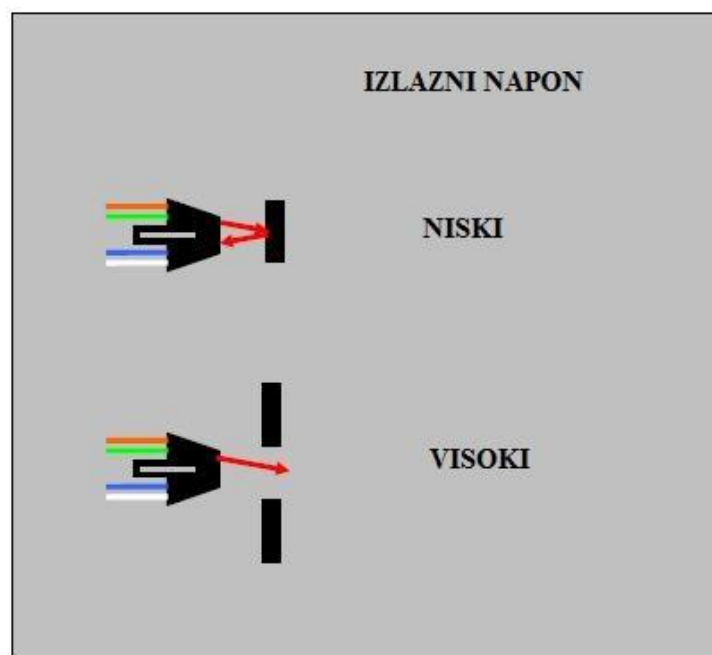
Infracrveni senzori Sharp GP2D12 za mjerenje udaljenosti imaju istu ulogu kao i ultrazvučni senzori te imaju raspon od 10 do 80 cm i nisu toliko precizni kao ultrazvučni senzori. Rade na principu emitiranja infracrvene svjetlosti koja se odbija od objekta te vraća u senzor koja onda na izlazu daje napon ovisno o udaljenosti tj. Jakosti svjetlosti (Slika 3.6). Na IntelliBrain robotu koristimo dva takva senzora kako bi robot prepoznao prepreke pod većim kutem.



Slika 3.6. Napon na izlazu infracrvenih senzora

Kako bi ovi senzori mogli prepoznati broj okretaja kotača uvjet je da robot ima razmaknute kružne otvore na samom kotaču što omogućava infracrvenom-fotoreflektivnom senzoru da prepozna kada je ispred njega otvor ili „kostur“ kotača te njihovim brojanjem možemo dobiti koliki je pomak kotača. Kod IntelliBrain robota oba kotača imaju po osam kružnih otvora te robot ima dva nezavisna Fairchild QRB1134 infracrvena-fotoreflektivna senzora od kojih je jedan pored lijevog i drugi pored desnog kotača.

Princip rada ovog sustava je sličan principu na koji rade slavine i javnim toaletima koje se automatski uključe i isključe. Infracrveni senzor sadrži emiter i detektor, gdje emiter emitira infracrvenu svjetlost, a detektor prepoznaje infracrvenu svjetlost. Detektor na izlazu može dati signal tj. napon koji ovisi o intezitetu svjetlosti koju detektor prima. Dakle, kada senzor emitira infracrvenu svjetlost dok se ispred njega nalazi „kostur“ kotača, senzor će se od njega odbiti u detektor te će u tom slučaju na izlazu biti niski napon. Kada senzor emitira svjetlost dok je kotač u tom položaju da se ispred senzora nalazi otvor, tada se infracrvena svjetlost neće odbiti i u tom slučaju napon na izlazu detektora bit će visok (slika 3.7).



Slika 3.7. Izlazni napon infracrvenih senzora s obzirom na položaj kotača

Iste infracrvene-fotoreflektirajuće senzore koje smo upotrijebili za brojanje okretaja kotača možemo također primjeniti kao linijske senzore. Linijski senzori su senzori koji koriste infracrveno svjetlo za detektiranje reflektivne površine ispod robota što mu omogućuje praćenje ne reflektivne crne linije preko veoma reflektivne bijele površine. Sa dva takva senzora program može detektirati kada se robot nalazi lijevo, desno ili u centru crne linije. Ako bi koristili samo jedan linijski senzor, program bi mogao prepoznati samo ako senzor detektira liniju ili ne te nebi znao da li se robot nalazi lijevo ili desno od linije. Sa dva senzora program može prepoznati na kojem rubu linije se robot nalazi te tako ispraviti smjer.

S obzirom na položaj senzora prema liniji, u bilo koje vrijeme, robot će biti u jednom od šest stanja navedenim u Tablici 3.8. Iz tablice možemo uočiti koju naredbu treba robot izvršiti kada se nađe u jednom od stanja. Stanje “Izgubljen“ mora postojati ako program ne može prepoznati je li robot s desne ili lijeve strane linije tj. kad niti jedan senzor ne detektira liniju. To je moguće iz nekoliko razloga naprimjer ako pokrenemo program, a senzori u tom trenutku ne detektiraju liniju. Isto tako, ako robot dođe do završetka linije, program će biti u stanju “Izgubljen“ i program će se zaustaviti.

Naziv stanja	Opis stanja	Naredba
Izgubljen	Nepoznata pozicija	Stop
Oba lijevo	Oba senzora su lijevo od linije	Skreni čvrsto desno
Jedan lijevo	Jedan senzor je lijevo od linije	Skreni desno
Centar	Oba senzora su preko linije	Vози ravno
Jedan desno	Jedan senzor je desno od linije	Skreni lijevo
Oba desno	Oba senzora su desno od linije	Skreni čvrsto lijevo

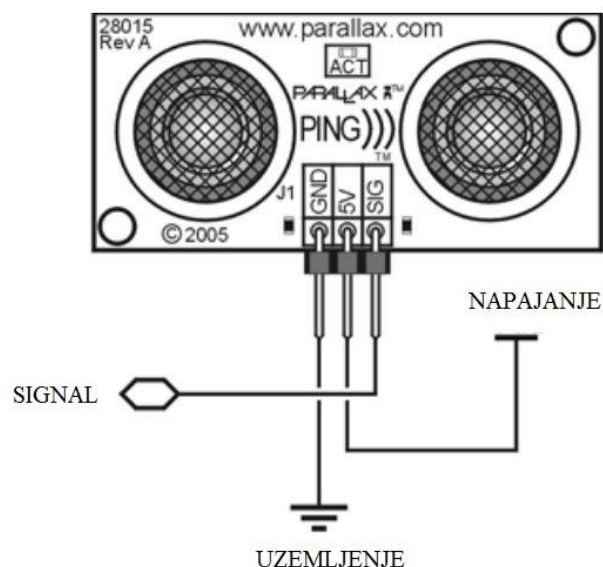
Tablica 3.8. Stanja linijskih senzora



### 3.1.6 Ultrazvučni senzor

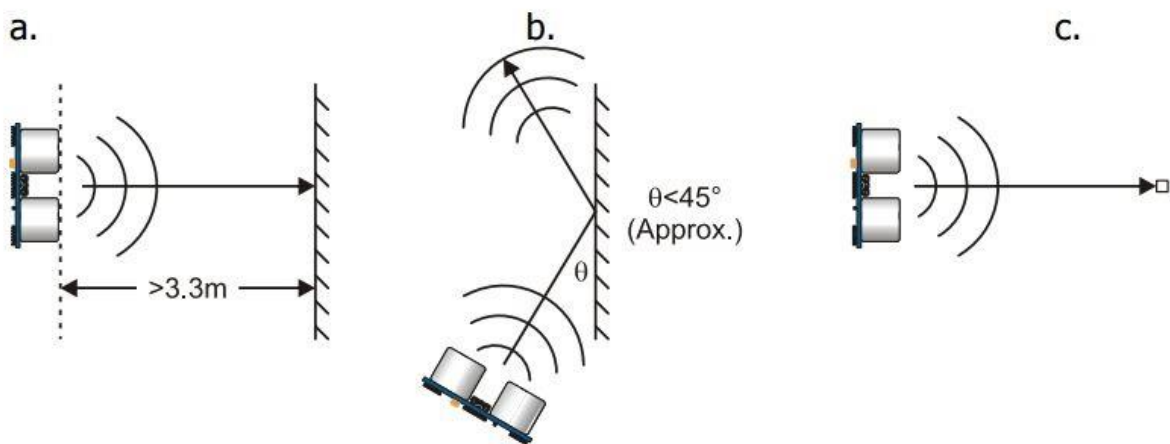
Ultrazvučni senzor Parallax Ping))) IntelliBrain mobilnog robota, mjeri vrlo preciznu udaljenost od početka senzora do objekta u rasponu od 2 centimetra do 3 metra. Karakterističan je po „očima“ od kojih jedno predstavlja generator ultrazvuka koji generira i širi impuls ultrazvuka frekvencije 40 kHz, dok drugo „oko“ predstavlja prijemnik koji prima taj impuls nakon što se odbije od objekta. Nakon primanja signala, senzor mjeri udaljenost na temelju brzine zvuka u zraku i vremena koje potrebno da se signal odbije od objekta i vrati u senzor. Jedan od uvjeta je ovog senzora je razmak između svakog mjerenja koji mora iznositi minimalno 200 mikrosekundi, a ako senzor ne primi signal odnosno impuls, generator će poslati novi impuls nakon 18.5 milisekundi. Isto tako, impuls koji šaljemo mora biti najmanje 5 mikrosekundi dugačak koji je već tvornički postavljen te je potrebno samo upisati naredbu koja šalje impuls.

Kod Parallax Ping))) senzora, prijemnik za primanje zvuka odnosno impulsa se uključuje 750 mikrosekundi nakon generiranja ultrazvučnog impulsa kako senzor ne bi čuo zvuk na samom početku generiranja. Protoklo vrijeme od generiranja impulsa do primanja istog u prijemnik, možemo dobiti tako da vremenu od uključivanja primjenika i primanja impulsa, dodamo 750 mikrosekundi. Na slici 3.9 vidimo senzore i kako priključujemo pinove na senzor.



Slika 3.9. Ultrazvučni senzor

Parallax Ping))) ultrazvučni senzori ne mogu precizno izmjeriti udaljenost do objekta u tri slučaja: a) ako je udaljenost do objekta veća od 3 metra, b) ako je kut smjera generiranih valova prema površini manji od  $45^\circ$ , c) ako je objekt toliko malih dimenzija da se zvuk ne može odbiti natrag u senzor. Također, ako smo senzor postavili tako da je blizu površine po kojoj se kreće robot, moguće je da senzor detektira odbijeni zvuk od te površine. Kada bi na robota ugradili dva ili više ultrazvučnih senzora izbjegli bi slučaj b), dok bi programski kod tada bio nešto složeniji.



Slika 3.10. Slučajevi nepreciznog mjerenja

Specifikacije Parallax PING))) senzora:

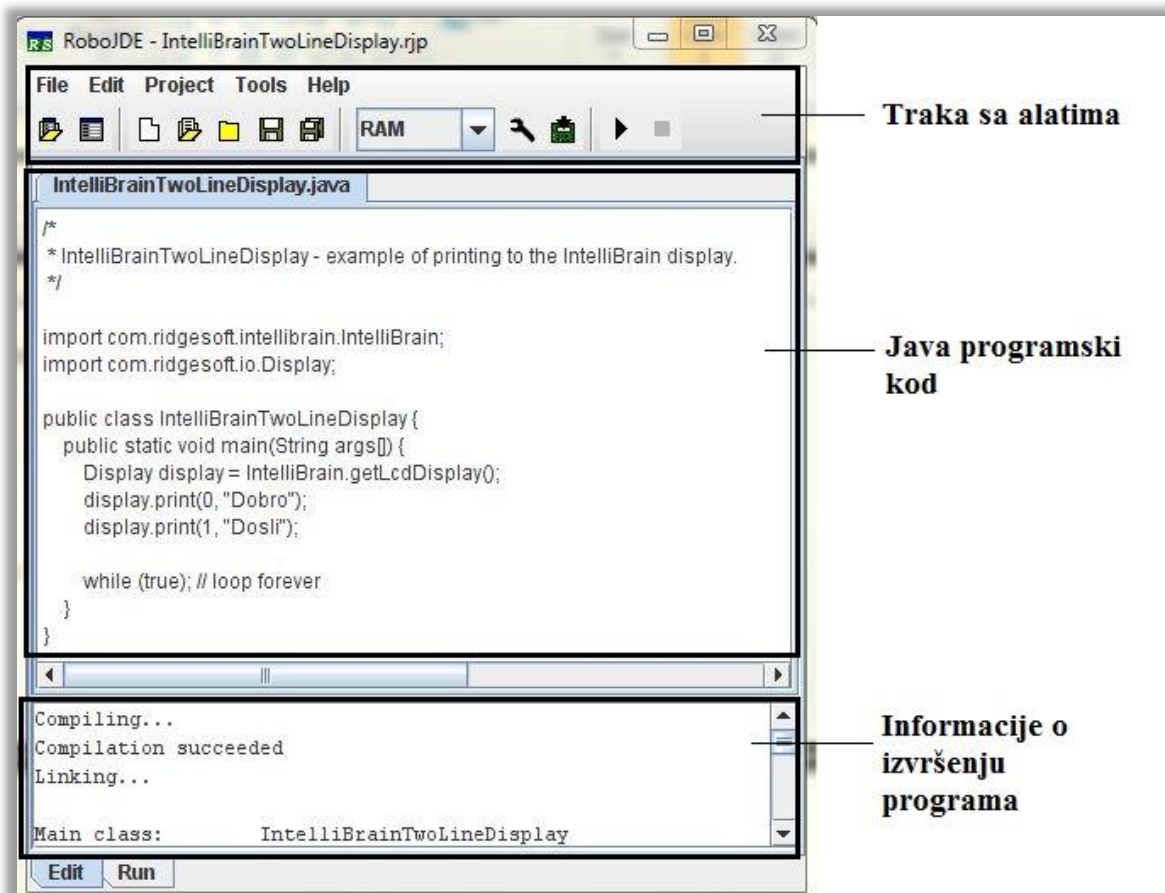
- Napajanje: 5 V
- Jakost električne struje: 30 mA, maksimalno 35 mA
- Frekvencija: 40 kHz
- Najmanja mjerljiva udaljenost: 2 cm
- Najveća mjerljiva udaljenost: 3 m
- Dimenzije: 22 mm x 46 mm x 16 mm
- Radna temperatura: 0 – 70° C
- 3 pina za spajanje: uzemljenje, napajanje i signal
- Težina: 9 g

## 4. Programiranje robota

Računalni program je skup uputa koje obavljaju određeni zadatak kada računalo to zatraži. Dakle možemo reći da je programiranje kreativni proces pisanja instrukcija koje računalo mora izvršiti. Glavni dio računala koji obrađuje programski kod je središnja jedinica za obradu ili mikroprocesor.

### 4.1 Programska podrška

Kako bi mogli izraditi programski kod za izbjegavanje prepreka te istoga učitati na robota, moramo imati računalo s računalnim programom (eng. Softwer) RoboJDE s kojim možemo napisani programski kod prevesti u kod razumljiv robotu te ga poslati u memoriju robota. Na slici 4.1 vidimo opisano sučelje RoboJDE programa.

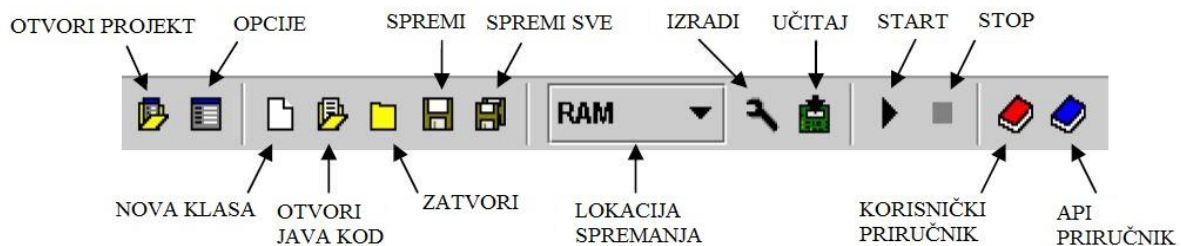


Slika 4.1. RoboJDE programska podrška

Također, ako u Java programskom kodu imamo grešku, u prozoru s informacijama o izvršenju programa, program će ispisati vrstu greške te dio koda koji program ne razumije i broj retka u kojem se greška nalazi. Time sami program možemo koristiti za pisanje koda kako bi za vrijeme njegove izrade ispravljali greške koje nam je puno jednostavnije i brže ispravljati korak po korak nego ih ispravljati nakon gotovog koda, pogotovo ako je kod većeg sadržaja.

Na slici 4.2 možemo vidjeti alate RoboJDE programa od kojih su nam najznačajniji: Lokacija spremanja, Izradi i Učitaj.

Ako pritisnemo na alat “Lokacija spremanja“, pojavit će nam se izbornik u kojem možemo odabrati memoriju u koju ćemo spremiti programski kod, a to može biti RAM ili FLASH memorija. Ako spremimo kod u RAM memoriju, program će se izbrisati onog trenutka kada isključimo napajanje robota dok kod spremanja koda na FLASH memoriju program se neće izbrisati nakon isključivanja napajanja. Unatoč tome, FLASH memorija se može potrošiti nakon ograničenog broja pisanja tj. spremanja. Zato pri testiranju programa uvijek koristimo RAM memoriju.



Slika 4.2. Alati programske podrške

Ako pritisnemo na alat “Izradi“, program će obaviti cijeli proces koji je potreban kako bi robot razumio kod te ispravno izvršavao naredbe. Taj proces se sastoji od prevođenja (eng. Compile) i povezivanja (eng. Link). Prevođenje je postupak kojim se nama razumljiv programski kod prevodi u strojni kod, kojeg razumije mikroprocesor, dok postupak povezivanja povezuje programski kod i klase koje smo koristili u kodu. Klase možemo definirati kao skup već napisanih naredbi koje koristimo često ili naredbe povezane sa nekim objektom, kao npr. servo motori.

Nakon završetka tog procesa, u prozoru s informacijama o izvršenju programa možemo vidjeti ako je prevođenje uspješno završeno ili je prevoditelj prepoznao greške u kodu.

Alat "Učitaj" izvršuje isti proces kao i alat Izradi te po završetku sprema programski kod u memoriju robota ovisno koju smo odabrali za lokaciju spremanja.

## **4.2 Java programski jezik**

Jedan od najpopularnijih programskih jezika je Java programski jezik koji omogućuje pisanje računalnih uputa pomoću naredbi temeljenih na engleskom jeziku. Java je objektno-orijentiran jezik koji ima bogat skup klasa kojima štedimo memoriju i pojednostavljujemo pisanje programa. Značajan je po tome što se nakon prevođenja može koristiti na mnoštvo računalnih arhitektura i operativnih sustava. Najviše se koristi tamo gdje je važna brzina razvoja programskog sustava dok je nedostatak Jave sporiji rad samog programa.

## **4.3 Program za izbjegavanje prepreka**

Na sljedećim stranicama opisani su elementi i algoritam programa za izbjegavanje prepreka koji koristi ultrazvučne senzore. Prije samog početka programa navest ćemo neka od pravila Java programskog jezika kojih se strogo moramo držati:

- Svaka naredba mora završavati sa točka-zarezom " ; "
- Moramo paziti na pisanje velikih i malih slova
- Svaka varijabla mora biti deklarirana (prirodni broj, cijeli broj, znakovi, slova i sl.)
- Sve varijable moraju početi sa slovom i ne smiju sadržavati razmake, navodnike i slične simbole

Na početku pisanja svakog programa moramo navesti klase koje ćemo koristiti u pisanju koda. U sljedećim redovima naveli smo klase za svaki element robota koji smo koristili u programu kao naprimjer klase za: IntelliBrain 2 kontroler, LCD zaslon, servo motor, ultrazvučni senzor.

```
import com.ridgesoft.intellibrain.IntelliBrain;
import com.ridgesoft.io.Display;
import com.ridgesoft.robotics.Motor;
import com.ridgesoft.robotics.ContinuousRotationServo;
import com.ridgesoft.robotics.RangeFinder;
import com.ridgesoft.robotics.sensors.ParallaxPing;
import com.ridgesoft.robotics.PushButton;
import com.ridgesoft.robotics.*;
```

Naš program smo također definirali kao klasu te ga tako možemo koristiti u drugim programima i nije potrebno pisati cijeli kod za izbjegavanje prepreka već ga samo pozovemo kao što smo uveli klase u samom početku programa. Kako bi program prepoznao koji dio čini klasu, taj dio programa moramo staviti u vitičaste zagrade.

```
public class Avoid {
```

U sljedećim redovima deklarirali smo program po pravilima programskog jezika gdje `public` označava vidljivost programa koji je u našem slučaju `java`. Naredba `static` označuje da program nije vezan za objekt te klasu možemo pozvati i bez objekta za koju je klasa napisana. Naredba `void` označava da program nema povratnu vrijednost te iznosi onda iznosi 0, dok naredbom `main(String[] args)` označavamo početak programa i složenost varijabli.

```
public static void main(String[] args) {
```

Zatim, naredba `try` koja obuhvaća dio koda koji će se izvršiti iako navedemo izuzetak naredbom `catch`, koja se piše nakon bloka označenog vitičastim zagradama.

```
try {
```

U sljedećim redovima pozivamo LCD zaslon i deklariramo ga naredbom `display` kojom zatim ispisujemo tekst u oba reda zaslona gdje broj 0 označava prvi red, a broj 1 definira drugi red.

```
Display display = IntelliBrain.getLcdDisplay();  
display.print(0, "Izbjegavanje");  
display.print(1, "prepreka");
```

Sljedećim naredbama pozivamo servo motore te ih deklariramo pod objekt `Motor` na lijevi i desni. Kako bi robot pomoću servo motora vozio ravno, smjer okretanja kod jednog motora mora biti u suprotnom smjeru od drugog motora. Stoga smo lijevom motoru zadali logičku naredbu `false` kako bi izbjegli negativni broj pri deklariranju brzine motora.

```
Motor leftMotor = new  
ContinuousRotationServo(IntelliBrain.getServo(1),  
false, 10);  
Motor rightMotor = new  
ContinuousRotationServo(IntelliBrain.getServo(2),  
true, 10);
```

U sljedećim naredbama pozivamo ultrazvučne senzore prema pravilima Java programskog jezika.

```
SonarRangeFinder sonarSensor = new  
ParallaxPing(IntelliBrain.getDigitalIO(3));
```

Sljedećom naredbom pozivamo tipku “start“ te nakon što pritisnemo tipku program će se nastaviti odvijati.

```
PushButton startButton = IntelliBrain.getStartButton();  
  
startButton.waitReleased();
```

Ako ne navedemo članove for petlje, blok naredbi koje obuhvaća petlja će se neprestano ponavljati, time će robot izbjegavati prepreke sve dok ne pritisnemo tipku stop na kontroleru ili isključimo napajanje robota.

```
for ( ; ; ) {
```

Sljedećim naredbama ultrazvučni senzor će generirati impuls i nakon generiranja program će pričekati 100 milisekundi za primanje odbijenog impulsa. U tom vremenu senzor za prijem signala čeka vraćanje odbijenog impulsa te u isto vrijeme pozivamo senzor za prijem signala i prvu očitane udaljenost pohranjujemo u varijablu `range`. Ako u tom vremenu senzor ne učita udaljenost odnosno ako je udaljenost veća od najveće mjerljive udaljenosti senzora, varijabla `range` će iznositi -1.0 inča. Sa naredbom `float` deklariramo varijablu kao cijeli broj.

```
sonarSensor.ping();  
Thread.sleep(100);  
float range = sonarSensor.getDistanceInches();
```

U sljedećim redovima koristimo if-else naredbu koja izvršava blok naredbi ako je uvjet zadovoljen te ako nije, izvršava blok naredbi koji slijede nakon nardebe `else`. U našem slučaju, u klasi ultrazvučnih senzora udaljenost se ispisuje u mjernoj jedinici inč. (eng. Inch). Ako je varijabla `range` veća od 0 inča i ujedno manja od 9 inča (logički “i“ se u java programskom jeziku piše simbolima “&&“) izvršiti će se blok naredbi u kojima smo lijevom motoru zadržali smjer okretaja dok smo desnom motoru promjenili smjer okretaja kako bi se robot okrenuo u svojoj osi prema desnoj strani i tako izbjegao prepreku.



Vrijeme okretaja robota smo definirali naredbom `Thread.sleep(100)` koja pauzira program na 100 milisekundi što omogućava robotu da promjeni smjer kretanja.

```
if (range > 0.0f && range < 9.0f) {  
    leftMotor.setPower(12);  
    rightMotor.setPower(-12);  
    Thread.sleep(100);  
}
```

Naredba `else` će se izvršiti samo kada uvjet `if` naredbe nije zadovoljen. To znači da ispred robota nema prepreke koju robot treba izbjeći i tada robot može voziti ravno što smo definirali kodom u bloku `else` naredbe. Svaki blok naredbi moramo zatvoriti vitičastim zagradama. U zadnjim redovima vidimo tri vitičaste zagrade od kojih prva zatvara `else` blok, druga `for` petlju koju smo započeli u prijašnjim redovima, a treća vitičasta zagrada zatvara `try` naredbu.

```
else {  
    leftMotor.setPower(10);  
    rightMotor.setPower(10);  
}  
}  
}
```

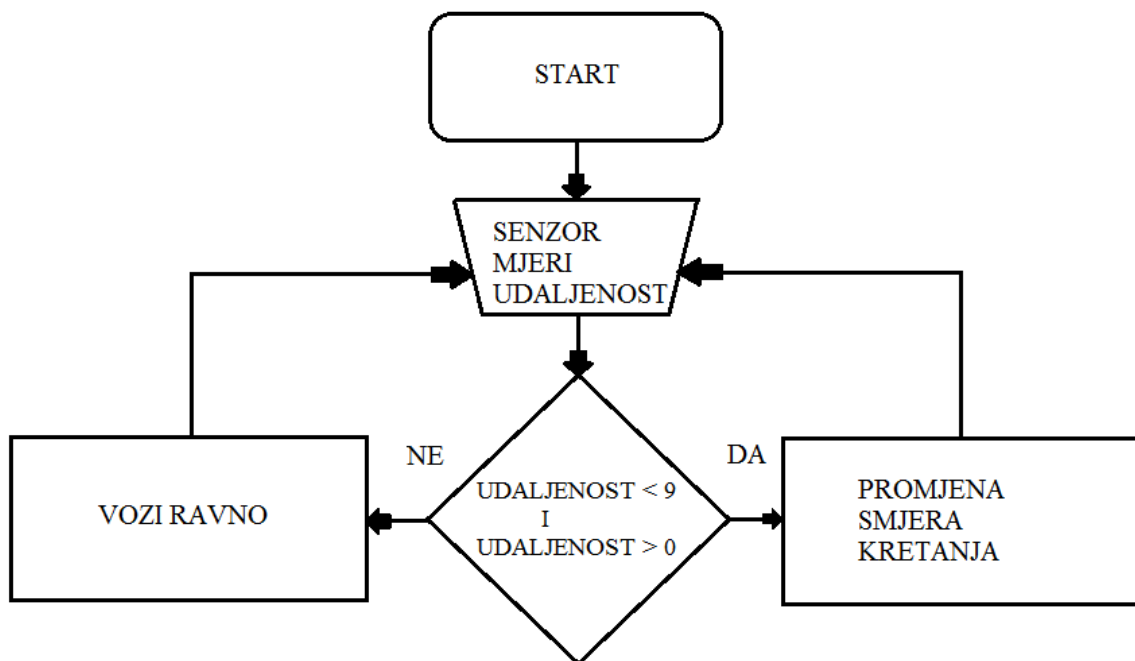
U sljedećim redovima nastavljamo započetu naredbu `try` poslije koje moramo definirati izuzetak. Ovim naredbama smo definirali program da unatoč mogućim odstupanjima program i dalje nastavi s naredbama te da samo zabilježi moguće greške u novi podatak izvan programa.

Zadnje dvije vitičaste zagrade čine kraj bloka klase Avoid i naredbe main.

```
catch (Throwable t) {  
    t.printStackTrace();  
}  
}  
}
```

Zatvaranjem vitičaste zagrade klase Avoid završili smo naš program odnosno klasu. Nakon toga program možemo kompajlirati (prevoditi) te tako testirati program od sintaktičkih pogrešaka. Ako program uspješno kompajlira programski kod, program možemo učitati na memoriju robota te nakon toga pokrenuti robot koji će koristiti učitani kod.

Na slici 4.3 vidimo algoritam kojim se postiže izbjegavanje prepreka te kojeg prikazujemo dijagramom tijeka. Algoritam je postupak ili formula za rješavanje nekog problema i obično se prije pisanja programa prvo izradi algoritam kako bi nam olakšao pisanje programa.



Slika 4.3. Algoritam za izbjegavanje prepreka

## 5. Zaključak

Ultrazvučni senzori su senzori koji šalju impuls i nakon što se odbije od objekta senzori ga primaju. Na temelju brzine zvuka i vremena koje je potrebno impulsu od senzora do objekta i nazad možemo lako izračunati udaljenost od objekta.

Koriste u više tehnoloških grana zbog niza prednosti kao što su: jednostavna ugradnja, cijena, preciznosti mjerenja, mogućnosti beskontaktnog ispitivanja, povezivanje sa računalom. Unatoč tome, ima i nedostatke: imaju minimalnu mjerljivu udaljenost, ovise o vremenskim utjecajima, mogućnost refleksije vala.

Danas se ultrazvučni senzori najviše koriste u medicini za vizualni pregled fetusa tijekom trudnoće, u pomorstvu kao dubinomjer i u uređaju za otkrivanje jata riba, u industriji za beskontaktna ispitivanja te kao alarm sustavi i parking senzori.

Zbog mogućnosti povezivanja i upravljanja preko računala, porasla im je upotreba u mobilnoj robotici. Korištenjem IntelliBrain-bot robot tvrtke Ridgesoft upoznajemo međusobnu povezanost ostalih elemenata robota i ultrazvučnih senzora sa Java programom odnosno programskim kodom. S mogućnosti pisanjem programskog koda dobivamo beskonačno mnogo mogućnosti za izradi programa koji će obavljati neki zadatak.

## 6. Popis literature

1. <http://www.ridgesoft.com/intellibrainbot/intellibrainbot.htm> 1.09.2016.
2. <https://www.parallax.com/sites/default/files/downloads/28015-PING-Documentation-v1.6.pdf> 1.09.2016.
1. Pavić T., Krklec Ž., Rumbak S., Ex-Bilten 2012., Primjena ultrazvuka u industriji s eksplozivnom atmosferom.