

# mBOT ROBOT PRATILAC ZVUKA

---

**Maračić, Borna**

**Undergraduate thesis / Završni rad**

**2018**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **The Polytechnic of Rijeka / Veleučilište u Rijeci**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:125:747405>

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

*Download date / Datum preuzimanja:* **2024-11-27**



*Repository / Repozitorij:*

[Polytechnic of Rijeka Digital Repository - DR PolyRi](#)



# **VELEUČILIŠTE U RIJECI**

Borna Maračić

## **mBOT ROBOT PRATILAC ZVUKA** (Završni rad)

Rijeka, 2018.



# **VELEUČILIŠTE U RIJECI**

Stručni studij Telematika

## **mBOT ROBOT PRATILAC ZVUKA**

(Završni rad)

MENTOR

Damir Malnar

STUDENT

Borna Maračić

MBS: 2427000004/14

Rijeka, lipanj 2018.

VELEUČILIŠTE U RIJECI

Studij Telematike

Rijeka, 17.04.2018.

**ZADATAK  
za završni rad**

Pristupniku Borni Maračić MBS: 2427000004/14

Studentu stručnoga studija telematike izdaje se zadatak za završni rad – tema završnog rada pod nazivom:

**mBOT ROBOT PRATILAC ZVUKA**


Sadržaj zadatka: Uspostaviti osnovni terminološki sustav koji uključuje Arduino IDE te mikrokontroler, senzore zvuka i aktuatore na mCore razvojnoj pločici iz Makeblock mBot robotske edukacijske platforme. Provesti analizu postojećih rješenja. Na osnovu provedene analize dizajnirati i realizirati robotski sustav za praćenje zvuka koristeći senzore zvuka i mBot kao osnovnu robotsku platformu. Prikazati glavne programske algoritme. Opisati uporabu sustava.

Rad obraditi sukladno odredbama Pravilnika o završnom radu Veleučilišta u Rijeci.

Zadano: 17.4.2018.

Predati do: 15.9.2018.

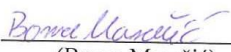
Mentor:

  
(Damir Malnar, mag. ing. el.)

Voditelj studija:

  
(izv.prof.dr.sc. Alen Jakupović)

Zadatak primio dana: 17.4.2018.

  
(Borna Maračić)

Prilog 5.

## IZJAVA

Izjavljujem da sam završni rad pod naslovom „mBot robot pratilac zvuka“ izradio samostalno pod nadzorom i uz stručnu pomoć mentora Damira Malnara.

Ime i prezime

Borisa Mercečić  
(potpis studenta)

## SAŽETAK

Robot pratilac zvuka je projekt izgradnje i programiranja robota sposobnog za detekciju zvuka u svojoj okolini i kretanje prema izvoru, namijenjen testiranju i istraživanju automatiziranih sustava u detekciji i reakciji na audio podražaje.

Analizom problema i prijašnjih rješenja zaključilo se da roboti podatke o svojoj okolini prikupljaju sensorima, a samu obradu informacija i upravljanje obavlja računalo. Kako bi se odredio smjer izvora zvuka potrebno je koristiti dva ili više senzora i uspoređivati intenzitete i/ili kašnjenja zvuka među sensorima. U određivanju smjera zvuka koristi se i koncept trilateracije.

Dizajniralo se i implementiralo hardversko i softversko rješenje. Robot pratioc zvuka koristi tri mikrofona kao senzore a određivanje smjera vrši se korištenjem trilateracije i mjerenjem razlike u razini zvuka između mikrofona.

Hardverske i softverske komponente korištene u izradi su Makeblockov mBot robot i Me Sound Sensor mikrofoni te Arduino IDE razvojno okruženje. Mikrofonu su povezani sa mBotom i postavljeni u trokutastu formaciju, koja je potrebna za trilateraciju. Arduino razvojno okruženje poslužilo je za realizaciju programskog rješenja: robot reagira na razinu zvuka u svojoj okolini, i ukoliko je okolina tiha, robot miruje, a ukoliko je intenzitet zvuka iznad određene razine, robot izračuna odakle zvuk dolazi i kreće se u smjeru izvora.

Ključne riječi: mBot, Zvuk, Senzor, Trilateracija, Praćenje

## SADRŽAJ

1. UVOD .....	1
2. Analiza zadatka .....	2
2.1. Analiza prijašnjih rješenja.....	5
2.1.1. Sound Following Rover .....	5
2.1.2. Blindbot .....	6
3. Dizajn rješenja.....	8
3.1. Specifikacije i opis komponenti.....	8
3.1.1. mBot .....	8
3.1.2. Me Sound Senzori - mBot interactive Light & Sound mikrofoni .....	13
3.1.3. mBlock i Arduino IDE razvojna okruženja.....	15
3.2. Koncept robotskog rješenja .....	18
3.3. Dizajn programskog rješenja .....	21
4. Implementacija rješenja .....	26
4.2. Konstrukcija mBot robota s dodatcima .....	26
4.3. Programiranje robota .....	32
5. Uporaba sustava .....	39
6. Zaključak.....	40
POPIS KRATICA I AKRONIMA.....	41
POPIS LITERATURE .....	42
POPIS TABLICA.....	43
POPIS SLIKA .....	43
POPIS KODOVA.....	44
POPIS PRILOGA.....	44
Prilog 1 - Arduino kod .....	45



## 1. UVOD

Robot pratilac zvuka je projekt sastavljanja i programiranja robota sposobnog za detekciju zvuka u neposrednoj okolini te kretanje prema izvoru.

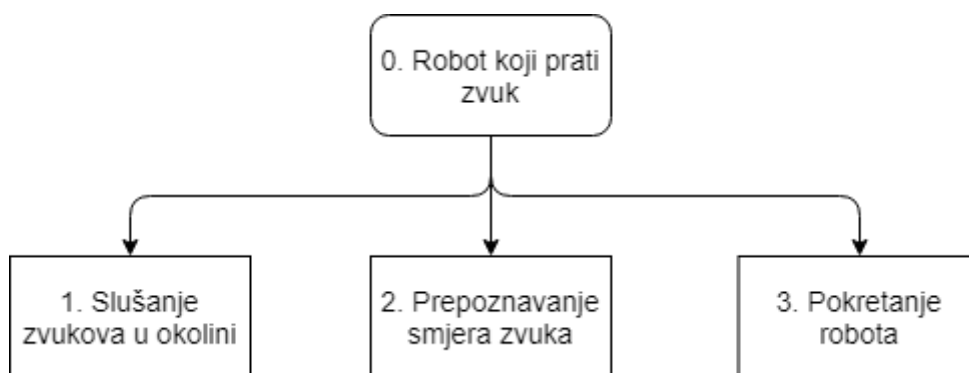
Sustav je namijenjen testiranju i istraživanju sposobnosti robota u rješavanju navedenog problema. Korisnici sustava mogu biti svi inženjeri, programeri i znanstveni istraživači kojima je potrebna baza ili prototip za testiranje, istraživanje ili izgradnju projekata koji uključuju sposobnosti robota da percipira i reagira na zvuk u svojoj okolini.

Projektna tema je izabrana zbog autorovog osobnog interesa u ostvarivanju ideje robota koji kretanjem prati korisnika samo na osnovi zvuka koji korisnik proizvodi, te zbog sadržajne povezanosti sa Stručnim studijem Telematike.

## 2. Analiza zadatka

Kao konačan rezultat treba izraditi robota koji je sposoban detektirati zvukove iz neposredne okoline, odrediti iz kojeg smjera zvuk dolazi te kretati se u smjeru izvora. Osnovne funkcionalnosti prikazane su dijagramom funkcija na slici 1.

Slika 1 - osnovni dijagram funkcija



Izvor: autor

Kako bi se traženi rezultat ostvario potrebno je znati što je to robot, kako funkcionira, na koji način može percipirati okolinu, kako se njime upravlja te na koji način je sposoban kretati se.

Robot je računalno programibilan, automatski upravljani stroj, koji može unaprijediti ili zamijeniti ljudski rad automatskim izvršavanjem serija kompleksnih zadataka. Disciplina koja se bavi dizajniranjem, izradom i upravljanjem robotima naziva se Robotika.

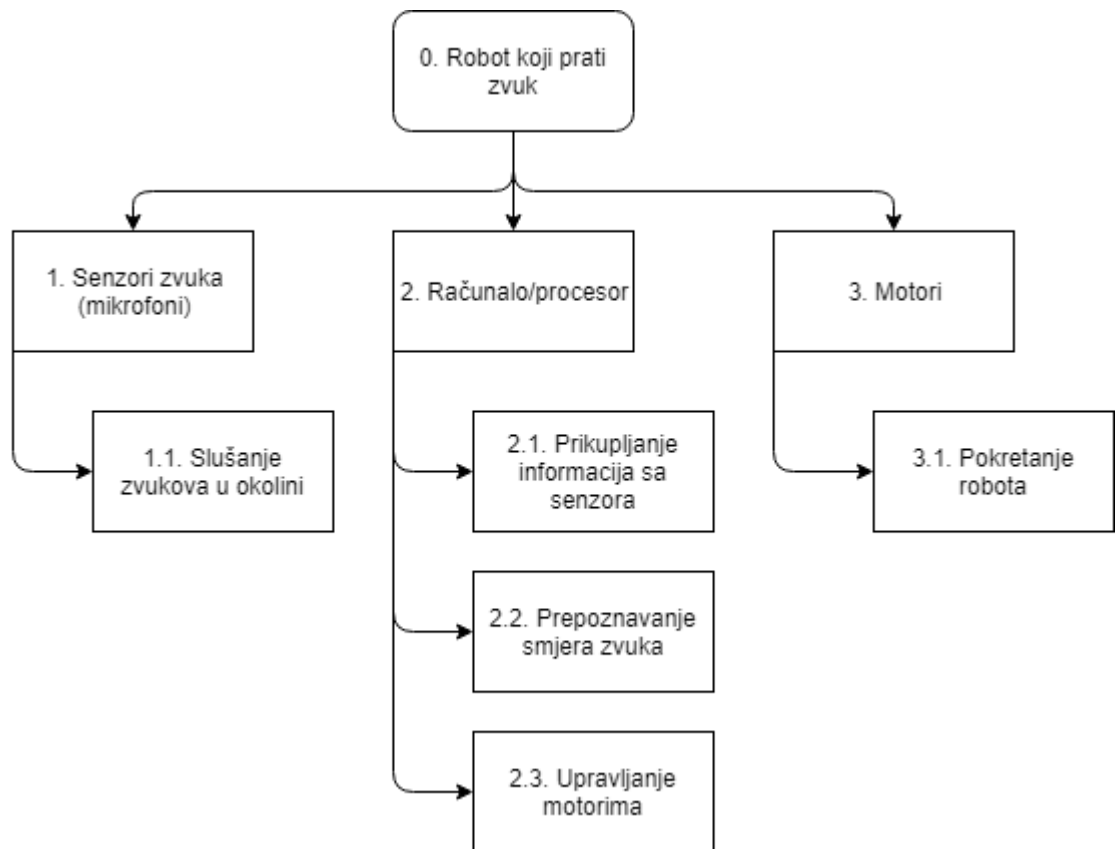
Robotika se razvila iz automatike, specijalizirane grane automatizacije u kojoj strojevi imaju određene antropomorfne karakteristike, npr., u industriji se koriste mehaničke robotske ruke koje se mogu programirati u seriju korisnih pokreta sa svrhom zamjene čovjeka kod zamornih repetitivnih radnji. Automatika obuhvaća skup tehnologija koje se odnose na izvođenje procesa putem programiranih naredbi uz povretnu vezu kako bi se osiguralo pravilno izvršavanje procesa. Rezultirani sustav sposoban je za rad bez ljudske interakcije. Automatizacija općenito podrazumijeva integraciju strojeva u samoupravne sustave, odnosno skupinu sustava u kojima se ljudska kontrola i inteligencija zamjenjuje računalom.

Upravljanje robotima obavlja računalo. Računalo je uređaj čija je svrha spremanje, procesiranje i prikaz informacija. Računalni program, ili softver, je detaljni plan ili procedura sa svrhom rješavanja nekog problema, odnosno jednoznačan niz uputa za postizanje rješenja pomoću računala, dok je hardver fizički dio računala. Programi mogu biti pohranjeni u memoriji računala i omogućavaju računalu da izvodi spremljeni niz zadataka po potrebi. Programi se pripremaju formuliranjem zadatka, te nakon toga izražavanjem programa u pripadajućem računalnom jeziku, koji je najprilagođeniji aplikaciji. Programski jezik je jezik za izražavanje skupa detaljnih uputa za digitalno računalo. Prije izvršavanja upute se kroz nekoliko stadija prevode u strojni jezik, kodirani program u numeričkom obliku koji je prepoznatljiv i direktno izvršiv od strane računala, te se u tom obliku mogu izravno izvršiti. Prevođenje iz programskih jezika više razine („*high-level*“), takozvanih problemsko-orijentiranih jezika, u assemblerske i strojne jezike niže razine („*low-level*“) izvodi compiler. Programski jezici više razine omogućavaju programerima lakši način čitanja i pisanja računalnih programskih uputa i ne zahtijevaju od programera da moraju brinuti o svim podatkovnim operacijama, dok programski jezici niže razine zahtijevaju od programera da eksplicitno upravlja svim računalnim operacijama vezanima za spremanje i obrađivanje podataka.

Roboti nalaze široku primjenu u industriji. Industrijski roboti su strojevi fleksibilnog ponašanja, generalno nehumanoidne forme, sa malo ljudskolikih fizičkih atributa (npr. robotska ruka) koji su razvijeni i korišteni u industrijske svrhe. 1960.-e u robotici se počinju koristiti mikroručunala, računala razvijena oko mikroprocesora, čipa s procesorskom snagom jednakoj računalnoj centralnoj procesorskoj jedinici (CPU, engl. Central Processing Unit). Mikrokontroleri su omogućili realizaciju kompleksnijih radnji u odnosu na radnje upravljane dotadašnjim sklopovljem. Osamdesetih godina dvadesetog stoljeća na Institutu Tehnologije u Massachusettsu (MIT, engl. Massachusetts Institute of Technology) pri laboratoriju za umjetnu inteligenciju (AI, engl. Artificial Intelligence) počinju se koristiti senzori u robotici. Senzori su fizikalno-elektronički uređaji čija je svrha pretvorba različitih fizikalnih veličina u električne te slanje preoblikovane informacije dalje, drugim elektroničkim uređajima, najčešće računalnom procesoru. Senzori reagiraju na fizičku stimulaciju: toplinu, svjetlost, zvuk, pritisak, magnetsko polje, i sl., te odašilju pripadajući rezultirajući impuls i na taj način omogućili su robotima da prikupljaju informacije o svojoj okolini (Encyclopaedia Britannica, lipanj 2018.).

U konačnici, rješenje zadatka treba biti robot koji će pomoću senzora za zvuk, odnosno mikrofona, percipirati zvukove u svojoj okolini te rezultirajuće informacije slati računalo. Nakon toga, ovisno o primljenim informacijama, računalo, ili u sučaju robota, mikroračunalo, će putem računalnog programa odrediti iz kojeg smjera dolazi registrirani zvuk te poslati signal motorima robota koji ga pokreću ovisno o smjeru izvora zvuka. Detaljne funkcionalnosti sustava koji treba realizirati prikazane su na slici 2.

Slika 2 - konačni dijagram funkcija



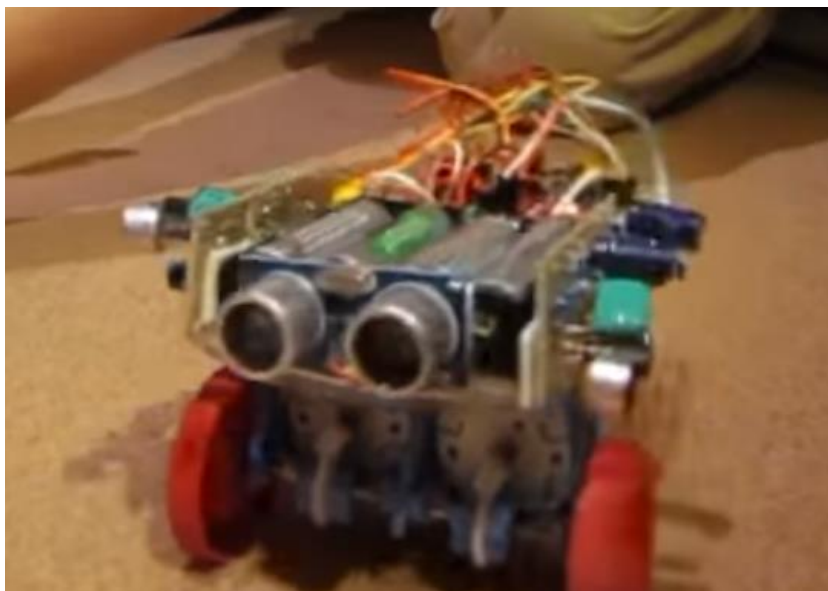
Izvor: autor

## 2.1. Analiza prijašnjih rješenja

### 2.1.1. Sound Following Rover

Sound Following Rover je robot koji prepoznaje smjer zvuka uspoređujući razine zvuka sa dva senzora postavljena na lijevu i desnu stranu robota. Robot se sastoji od Arduino Nano mikrokontrolera, dva istosmjerna motora za pogon, tri kotača, od kojih su dva pogonska, a treći služi kao balans, dva mikrofona koji služe kao senzori zvuka te 4 AA baterije kao izvor energije. Mikrokontroler i mikrofoni napajani su sa 4,8 V, a isti izvor napaja i pojačalo koje signal sa senzora zvuka pojačava 300 puta. Usporedbom razine signala u oba mikrofona robot dobija informaciju iz kojeg smjera je zvuk glasniji i prema toj se informaciji okreće dok se ne okrene prema izvoru nakog čega se kreće naprijed. Nedostaci opisanog rješenja su nepotpuna pokrivenost robota; ukoliko je izvor zvuka direktno iza robota, robot će se kretati prema naprijed jer smatra da je izvor zvuka ispred njega, te moguća interferencija zvuka s mikrofonom pri odbijanju zvuka od zidova i sličnih prepreka. Sound Following rover prikazan je na slici 3 (Black, lipanj 2018.).

Slika 3 - Sound Following Rover

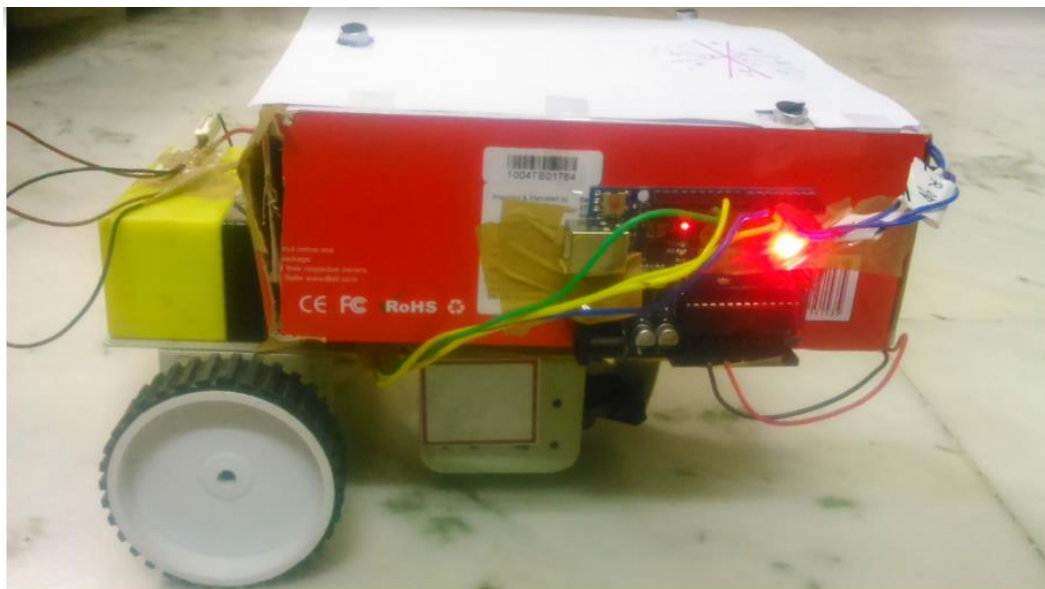


Izvor: autor

### 2.1.2. Blindbot

Blindbot je projekt Abhisheka i Dhruvila Shah sa Instituta Tehnologije Sveučilišta Nirma u Ahmedabadu. Prvobitno rješenje također je koristilo razliku u razini zvuka između dva mikrofona slično rješenju Peter Black robota, što je rezultiralo i sličnim nedostacima. Zaključak je bio da mjerenje razlike u razinama zvuka nije dovoljno precizno na većim udaljenostima. Sukladno, Abhishek i Dhruvil su napravili novo rješenje sa tri mikrofona, koje riješava nedostatak diferenciranja prednjeg i stražnjeg smjera. Rješenje također uključuje i princip trilateracije te mjerenje kašnjenja signala umjesto nepreciznog mjerenja razlike intenziteta. Blindbot se sastoji od 3 mikrofona kao senzora zvuka, baterije od 7.2V, Arduino Uno mikrokontrolera, IC 1234D pojačala snage za motore i 2 motora. Blindbot je prikazan na slici 4. (Shah A., Shah D., lipanj 2018.)

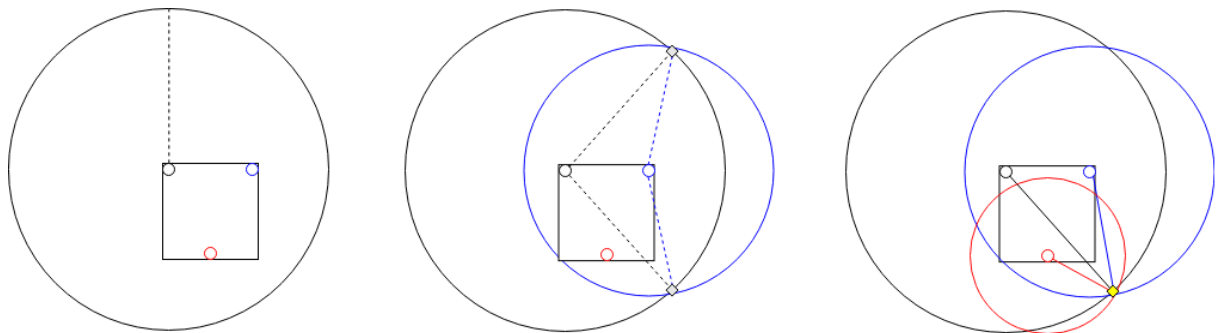
Slika 4 - Blindbot



Izvor: Shah A., D., lipanj 2018.

Blindbot mjeri kašnjenje između senzora koji primaju zvuk kako bi odredio iz kojeg smjera zvuk dolazi. Ta metoda bazirana je na principu trilateracije; procesu određivanja apsolutne ili relativne lokacije točaka mjerenjem udaljenosti, koristeći kružnice, sfere, ili trokute, a široku primjenu nalazi u nadziranju i navigaciji, specifično u GPS sustavima (GISGeography, 2018.). Postupak određivanja smjera principom trilateracije je: koriste se tri senzora, koji zasebno mjere svoju udaljenost od objekta. Mjerenja prva dva senzora određuju kut pod kojim se objekt nalazi, a treće mjerenje određuje stranu. Primjer trilateracije prikazan je na slici 5. gdje kvadrat predstavlja robota, crvena, bijela i plava točka su senzori, iscrtkane linije i kružnice predstavljaju izmjerenu udaljenost objekta od senzora, bijeli rombi predstavljaju moguće položaje objekta nakon mjerenja prva dva senzora te žuti romb predstavlja konačni izmjereni položaj detektiranog objekta.

Slika 5 - primjer trilateracije



(izvor: autor)

Prostor oko Blindbota je zatim podijeljen u 8 jednakih zona, odnosno sektora, širokih  $45^\circ$ , označenih brojevima od 1 do 8 počinjući od zone koja se nalazi direktno naprijed. Ovisno o rezultatima mjerenja trilateracije, lokacija zvuka pada u jednu od tih zona. Blindbot se nakon toga okreće prema zoni u kojoj se zvuk nalazi i kreće se prema izvoru zvuka. (Shah A., D., lipanj 2018.)

### 3. Dizajn rješenja

Kao baza rješenja, odabran je Sound follower robot prema Peteru Blacku. Sound follower robot prepoznaje smjer iz kojeg dolazi zvuk uspoređivanjem razine zvuka između dva mikrofona postavljena na njegovoj lijevoj i desnoj strani. Glavni problem Sound follower robota je nemogućnost prepoznavanja dolazi li zvuk s prednje ili stražnje strane. Ovaj nedostatak nadoknadiće se dodavanjem trećeg mikrofona koji će biti postavljen na stražnjem dijelu robota i korištenjem osnovnih principa trilateracije, prema Blindbot rješenju (Shah A., D., lipanj 2018.). Unatoč mnogim prednostima korištenja vremenske razlike u detekciji zvuka koja se koristi u Blindbotu, konačno rješenje koristit će detekciju na osnovi razlike razine zvuka korištenu u Sound Follower Robotu zbog manje kompleksnosti cjelokupnog sustava. Od hardverskih komponenti koristit će se Makeblock mBot i Me Sound Senzor mikrofoni, a od softvera koristit će se Arduino IDE.

#### 3.1. Specifikacije i opis komponenti

##### 3.1.1. mBot

mBot je modularni i programibilni robot baziran na Arduino Uno mikroprocesorskoj upravljačkoj ploči izrađen od kompanije Makeblock. Makeblock je privatna kineska tehnološka kompanija iz Shenzena osnovana 2013., i proizvodi hardver i softver za robotiku baziran na Arduino mikroprocesorskim pločama i Scratch programskom jeziku. Arduino je open-source platforma za izradu elektroničkih projekata. Sastoji se od mikrokontrolera, i integriranog razvojnog okruženja (IDE, *engl.* **I**ntegrated **D**evelopment **E**nvironment) koji se pokreće na osobnom računalu i koristi za pisanje i učitavanje računalnog koda na mikrokontroler. Arduino ne zahtijeva zaseban hardver kako bi se kod učitao na ploču, nego koristi USB sučelje. Scratch je besplatni edukativni programski jezik koji je razvio Lifelong Kindergarten Group na MIT-u. Namjena mu je edukacija početnika u svijetu programiranja, primarno osnovno i srednjoškolske djece. Makeblock također pruža i mnoge druge obrazovne alate za edukaciju početnika u svijetu programiranja, inženjerstva i robotike.



mBot je opisan kao sve-u-jednom praktično rješenje za edukaciju robotike, programiranja i elektronike. Jednostavan je za sastavljanje i pruža mnoge mogućnosti učenja vještina iz područja STEM-a (STEM, *engl.* Science, Technology, Engineering, Mathematics), kroz mBlock, programersko okruženje inspirirano Scratch-om 2.0, i povezivanjem s pametnim uređajima i računalima kabelom, Bluetoothom ili 2.4GHz WiFi vezom (ovisno o verziji mBot-a). mBlock također uključuje i Arduino i Makeblock biblioteke za olakšano programiranje. Jezgra mBota je mCore, glavna kontrolna ploča posebno dizajnirana za mBot robota. Bazirana je na Arduino Uno mikrokontroleru. Arduino Uno je mikrokontrolerska ploča bazirana na ATmega328 mikrokontroleru. Uno sadrži 14 digitalnih I/O pinova; od kojih se šest može koristiti kao izlazi za impulsno-širinsku modulaciju (PWM, *engl.* Pulse Width Modulation), a preostali se mogu konfigurirati kao analogni ulazi. Posjeduje i 16 MHz-ni kristalni oscilator, USB konekciju, utičnicu za napajanje, ICSP sučelje za programator (ICSP, *engl.* In-Circuit Serial Programming) te gumb za reset. ICSP je mogućnost programiranja nakon ugrađivanja u sustav, umjesto potrebe za programiranjem prije ugrađivanja. mCore također integrira senzore i dodatke: zujalice za stvaranje zvučnih signala, senzore svjetla i RGB LED diode. Specifikacije mBota i mCorea prikazane su tablicama 1 i 2.

Tablica 1 - specifikacije mBota

mBot Specifikacije	
Glavna kontrolna ploča/čip	mCore main control board
Senzori	Light Sensor, Me Ultrasonic Sensor, Me Line Follower Sensor
Konektivnost	Bluetooth/ 2.4GHz (version) i USB type B
Napajanje	3.7V litij baterija / 4x 1.5V AA baterije
Dimenzije	170 x 130 x 90 mm
Težina	1034g

Izvor: Makeblock

Tablica 2 - specifikacije mCore-a

mCore specifikacije	
Radni napon	3.7 – 6 V DC
Mikrokontroler	Atmega328
Dimenzije	90 x 79 x 18 mm
Težina	43g

Izvor: Makeblock

Četiri RJ25 porta spojena su na digitalne i analogne ulaze i izlaze (pinove) mCorea, port 1 spojen je na pin 12, port 2 na pin 10, port 3 na pin A3 i port 4 na pin A1, i oni služe kao glavni konektori za senzore i druge dodatke mBota. Pinovi RJ25 konektora spajaju se redom, s lijeva na desno, sa serijskom linijom takta (SCL *engl. Serial Clock Line*, serijskom podatkovnom linijom (SDA, *engl. Serial Data Line*), napajanjem te s analognim i digitalnim ulazima/izlazima sa zadnja dva pina, kako je prikazano u tablici 3.

Tablica 3 - konekcije mCore RJ25 pinova

mCore RJ25 pinovi	
Pin 1	SCL
Pin 2	SDA
Pin 3	GND
Pin 4	5V
Pin 5	I/O
Pin 6	I/O

Izvor: Makeblock

mBot sadrži 2 konektora za analogne ulaze i izlaze i 2 konektora za digitalne. Iako su pinovi 9, 11, A0 i A2 slobodni, oni sami po sebi nisu dostupni na mBotu. Osim standardnih Arduino pinova, mCore koristi i dodatna 2 pina za motore označena M1 i M2, koji se spajaju na lijevi i desni motor. Ostale, neoznačene portove, mBot koristi za spajanje s drugim sensorima i dodacima na mCore ploči: port 6 je povezan sa LDR (*engl.* **L**ight **D**ependent **R**esistor) senzorom svjetla, port 7 sa dvije RGB LED diode te port 8 sa zujalicom. Spojevi mBot portova prikazani su u tablici 4.

Tablica 4 - povezivanja mBot portova

mBot pinovi, portovi i konektori	
Port 1	Digitalni I/O 1
Port 2	Digitalni I/O 2
Port 3	Analogni I/O 1
Port 4	Analogni I/O 2
Port 6	Senzor svjetla (LDR)
Port 7	2 RGB LED diode
Port 8	Zujalica
M1	Lijevi motor
M2	Desni motor

Izvor: Jensa, Flashgamer, 2. studeni 2016.

Glavna procesorska jedinica mCorea je Atmega328 mikrokontroler. Atmega328 je 8-bitni mikrokontroler baziran na RISC (*engl. Reduced Instruction Set Computer*) arhitekturi. Može postići do 20 MIPS<sup>1</sup>-a pri taktu od 20 MHz-a, što je ujedno i maksimalna frakvencija. Bitne značajke ATmega328 mikrokontrolera su dane u tablici 5.

Tablica 5 - specifikacije ATmega328 mikrokontrolera

ATmega328/P specifikacije	
Broj pinova	28/32
Flash memorija (byte)	32k
SRAM <sup>2</sup> memorija (byte)	2k
EEPROM <sup>3</sup> memorija (byte)	1k
I/O linije općenite uporabe	23
SPI <sup>4</sup>	2
TWI <sup>5</sup> (I <sup>2</sup> C) <sup>6</sup>	1
USART <sup>7</sup>	1
ADC <sup>8</sup>	10-bitni 15kSPS <sup>9</sup>
Broj ADC kanala	8
8-bit-na brojila	2
16-bit-na brojila	1

Izvor: Atmel, studeni 2016.

<sup>1</sup> MIPS, *engl. Million Instructions Per Second*

<sup>2</sup> SRAM, *engl. Static Random-Access Memory*

<sup>3</sup> EEPROM, *engl. Electrically Erasable Programmable Read-Only Memory*

<sup>4</sup> SPI, *engl. Serial Peripheral Interface*

<sup>5</sup> TWI, *engl. Two Wire Interface*

<sup>6</sup> I<sup>2</sup>C, *engl. Inter-Integrated Circuit*

<sup>7</sup> USART, *engl. Universal Synchronous and Asynchronous Receiver-Transmitter*

<sup>8</sup> ADC, *engl. Analog-to-Digital Converter*

<sup>9</sup> kSPS, *engl. kiloSamples Per Second*

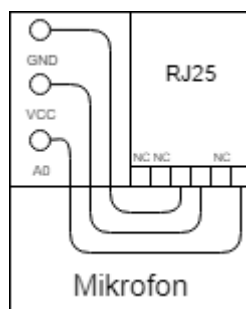
### 3.1.2. Me Sound Senzori - mBot interactive Light & Sound mikrofoni

Baziran na principu mikrofona, Me zvučni senzor može se koristiti za detektiranje intenziteta zvuka u neposrednoj okolini. Modul se može koristiti za izgradnju interaktivnih projekata, kao što je glasom upravljana sklopka ili robot koji pleše. Glavna komponenta je LM2904 pojačalo.

LM2904 se sastoji od dva nezavisna operacijska pojačala za rad na jednom napajanju. Niska potrošnja struje je neovisna o naponu napajanja. LM2904 pojačalo ima pojačanje od 100 dB, širinu frekvencijskog pojasa od 1 MHz, a može biti spojeno na jedno napajanje od 3 do 32V ili simetrično napajanje od +/-1.5 do +/-16V. (Texas Instruments, studeni 2014.)

Me Sound Sensori imaju visoku osjetljivost koja je vizualno predstavljena intenzitetom svijetla integrirane LED diode. U svrhu programiranja mikrofoni podržavaju mBlock grafičko sučelje (GUI, *engl.* Graphic User Interface) za programiranje. Za spajanje sa mikrokontrolerskim pločama MeSoundSensor koristi RJ25 priključak za jednostavno povezivanje, ali sadrži i priključke pin tipa kako bi podržao većinu razvojnih kontrolnih ploča, uključujući Arduino seriju. Sadrži tri pin priključka: GND, VCC i A0. GND i VCC se odnose na napajanje, a A0 je analogni izlaz. Shema spajanja prikazana je na slici 6, a popis priključnica sa značenjem dan je u tablici 6.

Slika 6 - Me Sound Sensor shema



izvor: autor

Tablica 6 - pinovi Me Sound Sensor mikroфона

Mikrofon RJ25 pinovi	
Pin 1	NC <sup>10</sup>
Pin 2	NC
Pin 3	GND
Pin 4	VCC
Pin 5	NC
Pin 6	A0

Izvor: Makeblock

MeSound senzori predviđeni su za 5V DC uz maksimalnu struju 0,5mA pri impedanciji od 2,2kOhm. Osjetljivost mikroфона je od 50 do 54 dB-a na frekvenciji od 1 kHz, a radna frakvencija je od 16 do 20 kHz. Dimenzije Me Sound Sensora su 52x24x18 mm dužine, širine i visine. Specifikacije senzora prikazane su u tablici 7.

Tablica 7 - tehničke specifikacije Me Sound Sensor mikroфона

Tehničke specifikacije	
Radni napon	5V DC
Osjetljivost mikroфона (1kHz)	50-54 dB
Impedancija mikroфона	2.2 KΩ
Frekvencija mikroфона	16-20 kHz
Maksimalna struja	0.5 mA
Dimenzije	52x24x18 mm
Pojačalo	LM2904 nisko-naponsko pojačalo
Način kontrole	Jedan analogni izlazni port

Izvor: Makeblock, 15. studeni 2017.

---

<sup>10</sup> NC, engl. Not Connected

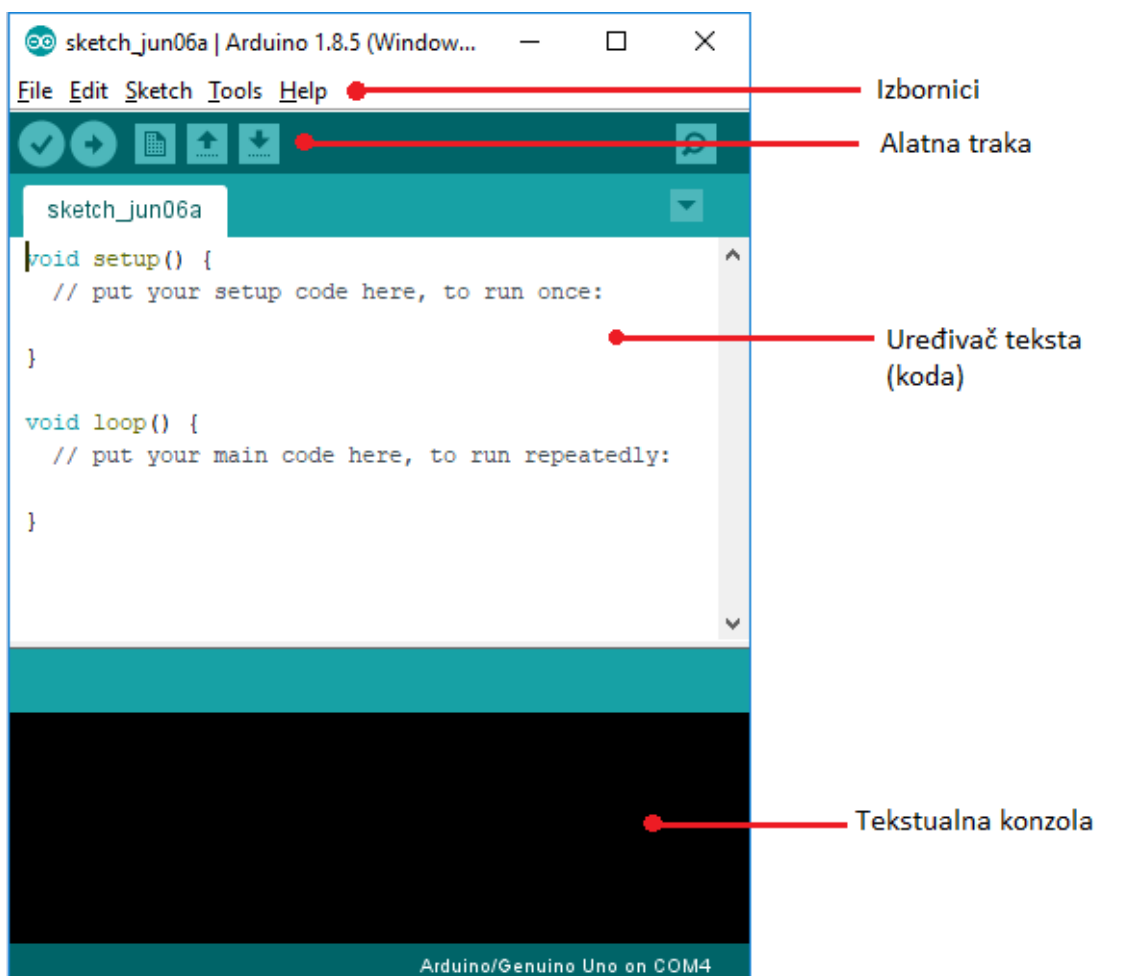
### 3.1.3. mBlock i Arduino IDE razvojna okruženja

mBlock je verzija Scratcha 2.0 koja može učitati programe na kontrolne ploče bazirane na Arduinu. mBlock sadrži: Scratch grafičko programiranje, Arduino programski jezik baziran na setu C i C++ funkcija za naprednije programere i Arduino-Scratch funkcije za sve Arduino mikrokontrolere i Makeblock Scratch funkcije za specifične Makeblock senzore, motore i prikladnu elektroniku. Podržava Bluetooth, 2.4GHz Wifi, USB i druga serijska sučelja. Inspirirano Scratchem, programiranje u mBlocku sastoji se od nizanja predprogramiranih blokova funkcija koje se pri učitavanju konvertiraju u Arduino kod.

Arduino IDE je cross-platformska Java aplikacija koja služi kao uređivač i prevodilac koda te je sposobna za serijski prijenos firmware-a na mikrokontrolersku ploču. Firmware je specifična vrsta računalnog softvera koji služi za upravljanje hardverom. Firmware može ili pružiti standardizirano operativno okruženje za složeniji softver uređaja (omogućavajući veću neovisnost hardvera) ili funkcionirati kao potpuni operativni sustav manje složenijih uređaja, obavljajući sve funkcije upravljanja, nadzora i manipulacije podacima. Gotovo svi elektronički uređaji osim onih najjednostavnijih sadrže firmware. Tipični primjeri uređaja sa firmwareom su potrošački uređaji, računala, računalne periferije te ugrađeni sustavi.

Arduino IDE je integrirano programsko okruženje koje se može koristiti pri radu s Arduino mikrokontrolerima. Programski jezik je set C i C++ funkcija koje mogu biti pozvane u programskom kodu. Temelji se na Processingu, IDE-u dizajniranom za uvod u programiranje, a Arduino programski jezik izveden je iz Wiringa, programskog jezika sličnog C-u također temeljenog na Processingu. Arduino IDE sadrži: uređivač teksta za pisanje koda, područje poruka, tekstualnu konzolu, alatnu traku sa gumbima za uobičajene funkcije i niz izbornika. Arduino IDE prozor prikazan je na slici 7.

Slika 7 - Arduino IDE prozor



Izvor: autor



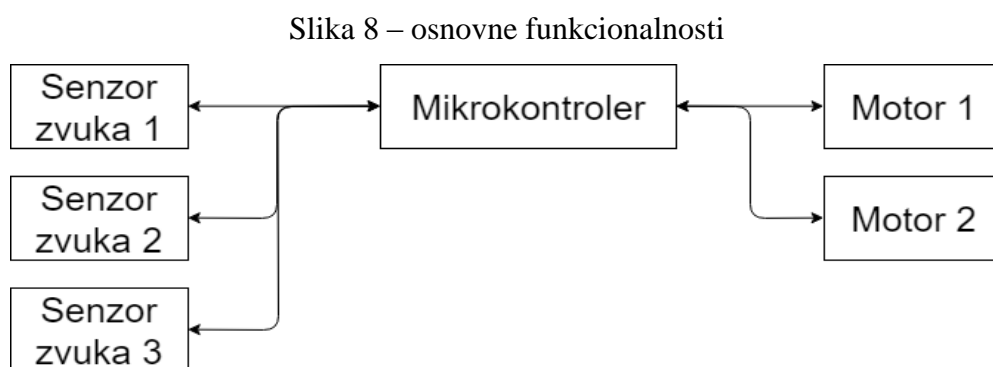
Arduino IDE se povezuje sa Arduino i Genuino hardverom za učitavanje programa i komunikaciju s uređajima. Programi napisani koristeći Arduino softver nazivaju se Sketch-evi. Sketch-evi se pišu u Arduinovom uređivaču teksta i spremaju se s ekstenzijom .ino. Područje poruka daje povratne informacije prilikom spremanja i exportanja te ispisuje poruke grešaka (error messages). Tekstualna konzola prikazuje izlazni tekst Arduino softvera, uključujući potpune poruke o greškama i druge informacije. Alatna traka sadrži gumb za provjeru i učitavanje programa, izradu, otvaranje i spremanje sketcheva i otvaranje serijskog monitora. Serijski monitor prikazuje i šalje podatke između Arduino softvera i Arduino ploče preko USB ili serijskog konektora.

Biblioteke koda (Libraries) pružaju dodatne funkcionalnosti za korištenje u sketчевima, npr. za rad s hardverom ili manipulaciju podacima. Biblioteke se dodaju u Sketch odabirom Sketch > import Library opcije u izborniku, ili korištenjem "#include" direktive unutar koda, pri čemu se odabrana biblioteka kompajlira zajedno sa sketchem. Neke biblioteke su automatski uključene u Arduino softver, a druge se mogu preuzeti s raznih vanjskih izvora ili pomoću Library Managera.

Prije učitavanja programskog koda potrebno je odabrati na koji tip mikrokontrolera se kod učitava te na kojem serijskom portu je mikrokontroler spojen s računalom. To se može napraviti odabirom opcija iz Tools > Boards, i Tools > Port izbornika. Tokom učitavanja koda, Arduino se automatski resetira i njegove RX i TX LED diode signaliziraju da je učitavanje u tijeku. Kada je učitavanje dovršeno Arduino softver će prikazati potvrdnu poruku ili poruku greške, ovisno o uspješnosti učitavanja. Za učitavanje koda na mikrokontroler koristi se Arduino bootloader, program koji je učitao na mikrokontroler i omogućava prenošenje koda bez dodatnog hardvera. Bootloader je aktivan nekoliko sekundi kada se ploča resetira i onda pokrene najnoviji kod koji je učitao na mikrokontroler. Bootloader će signalizirati LED diodom na pinu 13 kada se pokrene, odnosno kada se mikrokontroler resetira.

### 3.2. Koncept robotskog rješenja

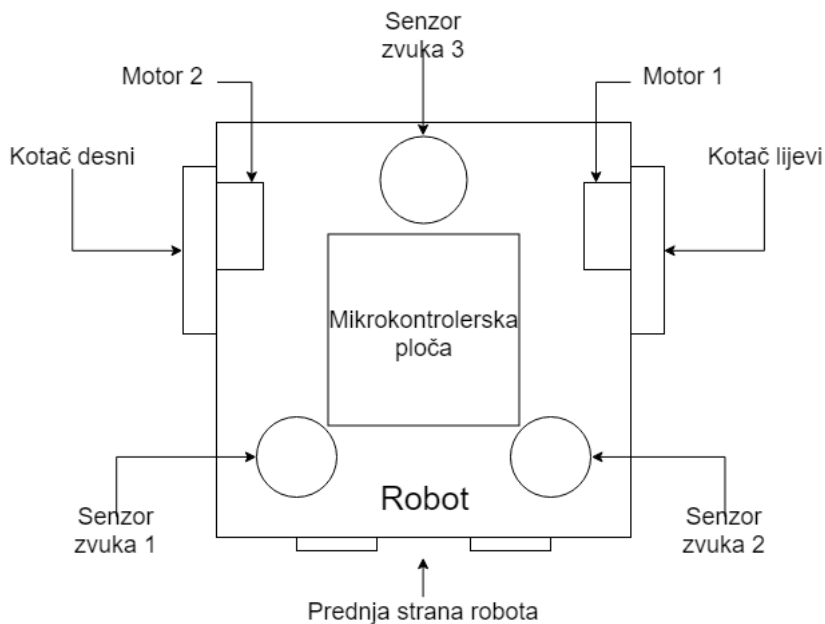
Robot koji se izrađuje mora biti sposoban detektirati zvuk iz neposredne okoline neovisno o smjeru, izračunati smjer zvuka i pokretati se. Prema tome, Robot mora imati senzore zvuka, raspoređene tako da pokriju sve strane robota te mikrokontroler koji će izvoditi funkcije računanja smjera i upravljanja ostalim funkcijama i motore koji će ga pokretati. Shematski prikaz osnovnih funkcionalnosti koje trebaju biti prisutne prikazane su na slici 8.



Izvor: autor

Sastavljeni Makeblockov mBot u sebi već sadrži mikrokontrolersku ploču, mCore i dva motora koji upravljaju kotačima, iz čega proizlazi kako je potrebno dodati još samo senzore zvuka. Na osnovu primjera Sound Following Robota Petera Blacka i Blindbota Abhisheka i Dhruvila Shaha, dva senzora postavljaju se na prednju stranu robota, a treći na stražnju u trokutastoj formaciji, prema slici 9.

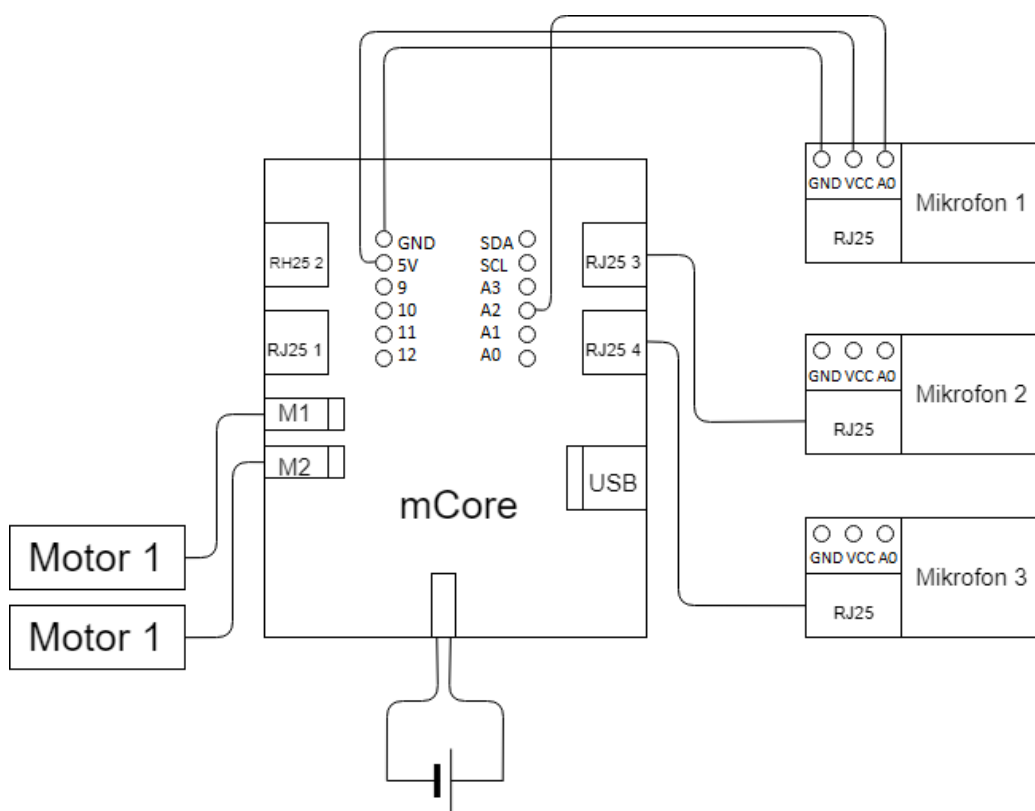
Slika 9 - shema robotskog rješenja



Izvor: autor

Motori mBota spojeni su na M1 i M2 konektore na mCore ploči. Potrebno je spojiti tri Me Sound Sensor mikrofona na mCore dok ploča ima samo dva RJ25 analogna konektora. Prema tome, dva mikrofona spajaju se na analogne RJ25 portove mCorea, označene brojevima 3 i 4, a jedan mikrofon spaja se žicama direktno na ploču. Analogni izlaz mikrofona 1, A0, spaja se s pinom analognog ulaza mCorea A2, a, preko RJ25 konektora, analogni izlaz mikrofona 2 spaja se na pin A3 i analogni izlaz mikrofona 3 spaja se na pin A1, prema slici 10 i tablici 8.

Slika 10 - shema povezivanja hardwarea



Izvor: autor

Tablica 8 - povezivanje Arduino i mikrofon pinova

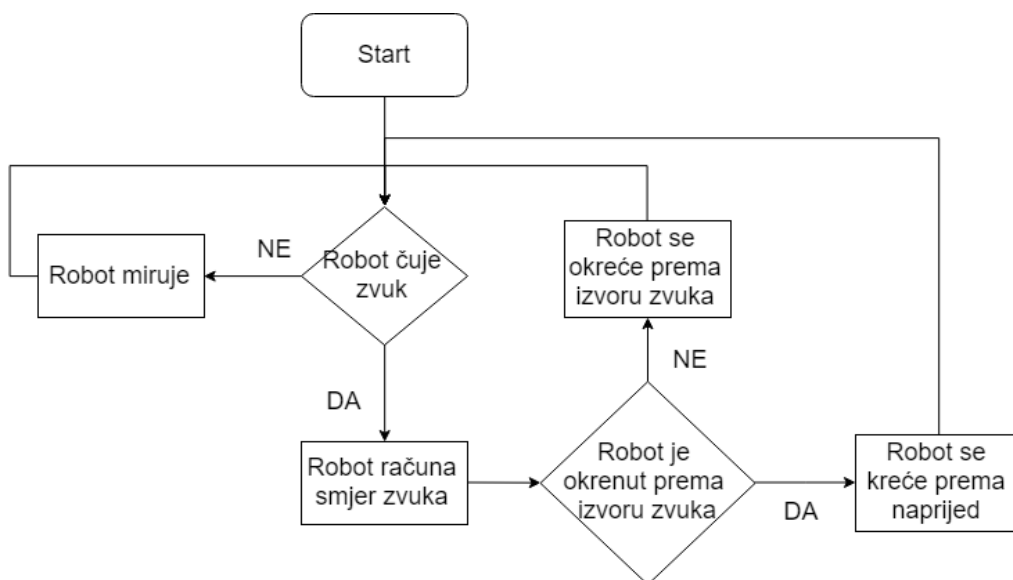
A0	Arduino pin mCorea
Mikrofon 1	A2
Mikrofon 2	A3
Mikrofon 3	A1

Izvor: autor

### 3.3. Dizajn programskog rješenja

Robot treba biti programiran za detekciju zvukova u svojoj okolini, i ukoliko je okolina tiha, robot miruje, a ukoliko čuje zvuk, izračuna odakle zvuk dolazi i kreće se u tom smjeru. Funkcionalna blok-shema ponašanja robota obzirom na senzorske ulaze dana je na slici 11, a realizacija pseudokodom unutar tekstualnog okvira Kod 1.

Slika 11 - dijagram toka



Izvor: autor

Kod 1 – pseudokod 1

```
-Početak
-Ponavljaj
  -Slušaj zvuk u okolini:
    Dok se zvuk čuje,
      -Izračunaj smjer zvuka
      -Kreći se prema zvuku
    -Inače
      Stani
```

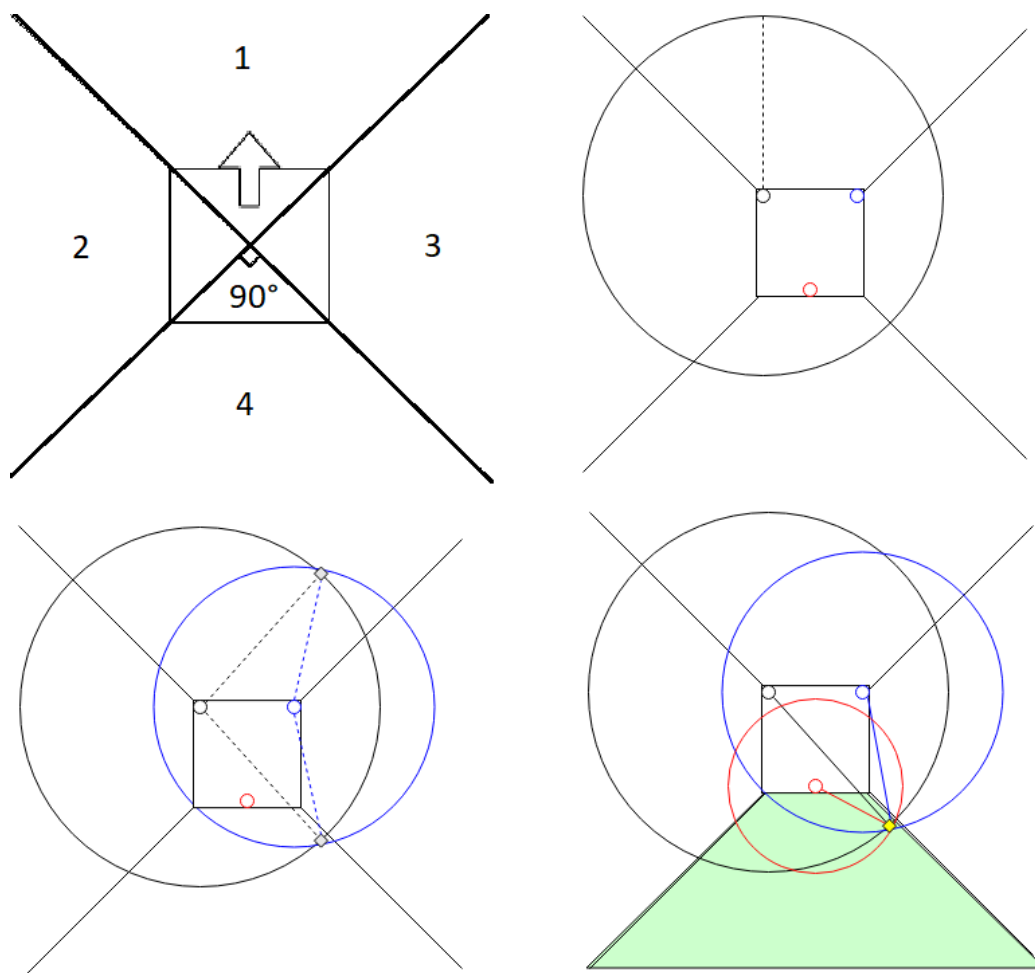
Detekciju zvukova u svojoj okolini robot će obavljati stalnim prikupljanjem podataka sa Me Sound Sensor mikrofona. Me Sound Sensor mikrokontroleru šalje cjelobrojnu vrijednost od 0 do približno 980 ovisno o intenzitetu zvuka u okolini; do 483 u tihim uvjetima, te od 483 do približno 980 u bučnim. Primanje većeg broja od neke određene vrijednosti registrirati će se kao primjećeni zvuk. 375 će se postaviti kao minimalna vrijednost koja se registrira kao primjećeni zvuk. Ta vrijednost je odabrana iz razloga što rad mBot motora proizvodi zvuk koji stvara vrijednost od prosječno 350, pa bi manja vrijednost od toga uzrokovala da se robot kreće nasumično. Realizacija detekcije zvukova pseudokodom dana je unutar tekstualnog okvira Kod 2.

#### Kod 2 – pseudokod 2

-Početak
-Ponavljaj
-Dohvati vrijednost sa Me Sound Sensora:
Dok je vrijednost 375 ili veća,
-Izračunaj smjer zvuka
-Kreći se prema zvuku
-Inače
Stani

Za izračunavanje smjera iz kojeg dolazi zvuk koristit će se osnovni principi trilateracije zajedno sa podjelom područja oko robota na zone prema primjeru Blindbota. Prema principima trilateracije smjer zvuka izračunati će se uspoređivanjem razine zvuka na prvom i drugom senzoru i zatim na trećem sa vrijednosti na prvom ili na drugom senzoru ovisno o rezultatu prve usporedbe, odnosno, čija je vrijednost bila veća. Prva usporedba pokazati će dolazi li zvuk s lijeve ili s desne strane robota, a druga dolazi li zvuk sprijeda ili straga. Područje oko robota biti će podijeljeno na četiri zone slučajja približne širine  $90^\circ$ . Rezultat računanja smjera zvuka postaviti će izvor zvuka u jednu od te četiri zone. Podjela područja na zone i primjena trilateracije prikazana je na slici 12, a realizacija određivanja smjera dana je pseudokodom unutar tekstualnog okvira Kod 3.

Slika 12 - podjela na zone i primjena trilateracije



Izvor: autor

### Kod 3 – pseudokod 3

-Početak

-Ponavljaj

-Dohvati vrijednost sa Me Sound Sensora:

Dok je vrijednost 375 ili veća,

-Ako je vrijednost lijevog senzora veća od desnog, zvuk pada u 2. zonu;

Inače, ako je vrijednost desnog senzora veća od lijevog, zvuk pada u 3. zonu;

Inače, ako su vrijednosti podjednake

-ako je vrijednost stražnjeg senzora veća od veće vrijednosti lijevog i desnog senzora zvuk pada u 4. zonu;

Inače zvuk pada u 1. zonu

-Kreći se prema zvuku

-Inače

Stani



Kako bi se kretao u smjeru zvuka, robot će se kretati ovisno o zoni u koju zvuk padne. Ukoliko zvuk padne u prvu zonu, odnosno ispred mBota, mBot će se kretati naprijed, a ako padne u druge zone mBot će se okretati dok zvuk ne bude u prvoj. Ako zvuk padne u drugu zonu okretat će se ulijevo, ako padne u treću okretat će se udesno, a ako padne u četvrtu okretati će se ulijevo ili udesno ovisno o tome na kojem senzoru je vrijednost jača. Kako bi se mBot kretao prema naprijed aktivirat će se oba motora uz jednaku snagu, a kako bi se okretao u stranu motorima će se postaviti suprotne snage ovisno u kojem smjeru je potrebno izvršiti okret. Snaga se mBot motorima postavlja 8-bit-im vrijednostima, odnosno u vrijednostima od 0 do 255, pri čemu je 255 maksimalna snaga motora. Prema Makeblock uputama snaga motora postaviti će se na vrijednost +/- 100. Motori se u suprotnom smjeru okreću postavljanjem negativnih vrijednosti. Kada robot stoji motorima se postavlja vrijednost 0. Realizacija kretanja robota dana je pseudokodom unutar tekstualnog okvira Kod 4.

#### Kod 4 – pseudokod 4

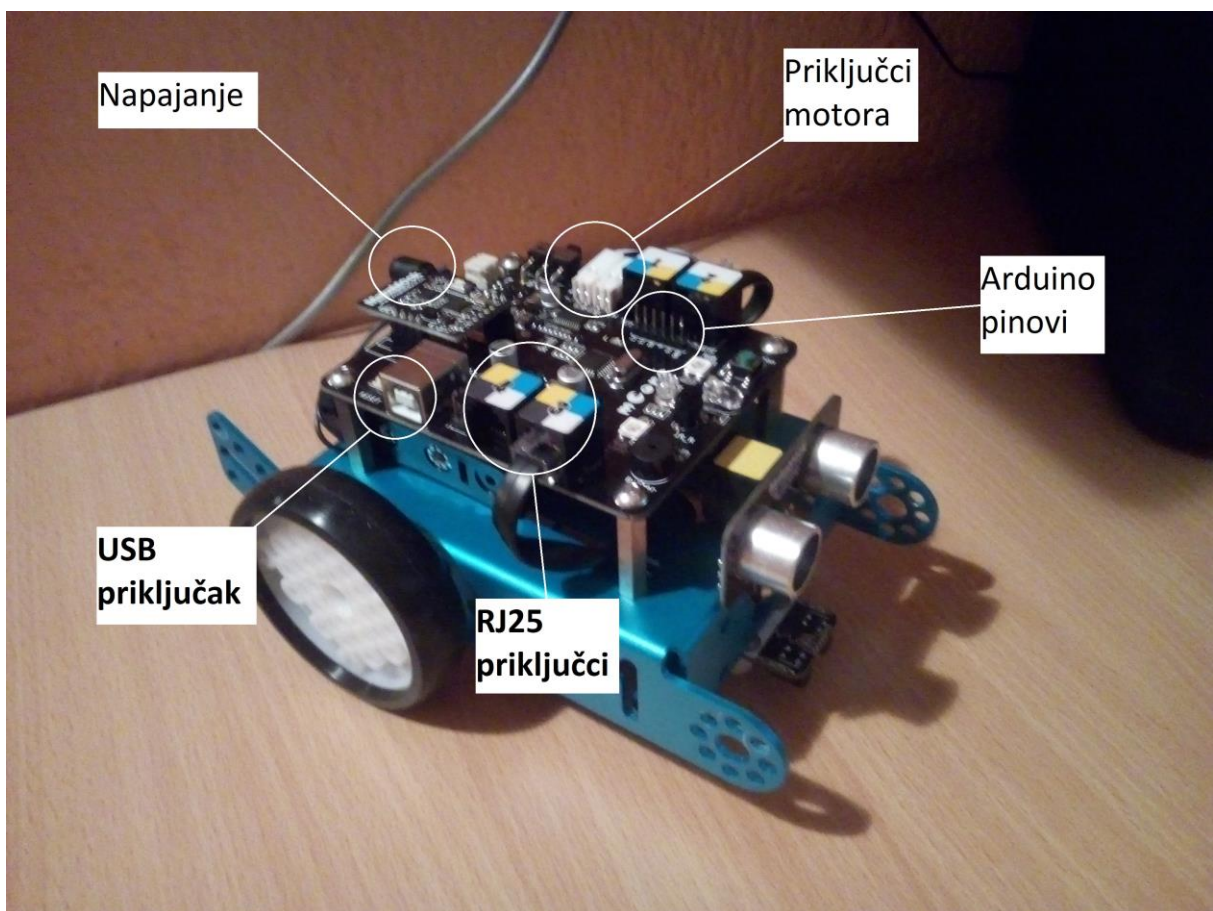
```
-Početak
-Ponavljaj
  -Dohvati vrijednost sa Me Sound Sensora:
  Dok je vrijednost 375 ili veća,
    -Ako je vrijednost lijevog senzora veća od desnog, zvuk pada u 2. zonu;
    Inače, ako je vrijednost desnog senzora veća od lijevog, zvuk pada u 3. zonu;
    Inače, ako su vrijednosti podjednake
      -ako je vrijednost stražnjeg senzora veća od veće vrijednosti lijevog i desnog
      senzora zvuk pada u 4. zonu;
      Inače zvuk pada u 1. zonu
    -Ako je zvuk u 1. zoni: snaga motora 1 = 100, snaga motora 2 =100;
    Inače, ako je zvuk u 2. zoni: snaga motora 1 = -100, snaga motora 2 =100;
    Inače, ako je zvuk u 3. zoni: snaga motora 1 = 100, snaga motora 2 = -100;
    Inače, ako je zvuk u 4. zoni:
      Ako je vrijednost lijevog senzora veća od desnog:
        snaga motora 1 = -100, snaga motora 2 = 100;
      Inače: snaga motora 1 = 100, snaga motora 2 = -100;
  -Inače
    Snaga motora 1 = 0, snaga motora 2 = 0;
```

## 4. Implementacija rješenja

### 4.2. Konstrukcija mBot robota s dodatcima

U implementaciji robotskog rješenja korišten je već sastavljeni mBot (slika 12). Sastavljeni mBot već sadrži mCore mikrokontrolersku ploču i motore, potrebno je jedino instalirati mikrofone koji će služiti kao senzori zvuka. Mbot je vidljiv na slici 13.

Slika 13 - mBot



Izvor: autor

Koristeći dodatne nosače iz mBot seta sastavljena je platforma koja će držati senzore zvuka. U skladu s dizajnom robotskog rješenja, nosači su instalirani kako bi omogućili postavljanje mikrofona u trokutastu formaciju oko mBota. Prednja platforma mora biti šira kako bi se na nju mogla postaviti oba prednja mikrofona. Napravljena je od jedne široke grede i dvije tanke grede spojene parom vijaka i matica na široku gredu. Obje platforme instaliraju se na mBota spajanjem na predviđena mjesta na prednjoj i stražnjoj strani mBota. Popis korištenih dijelova prikazan je u tablici 9, a platforme i njihova instalacija na mBot prikazana je na slici 14..

Tablica 9 - dijelovi 1

Alati i dijelovi	Količina
MBot set debela greda	2
MBot set tanka greda	2
MBot set vijci	8
MBot set matice	8

(Izvor: autor)

Slika 14 - instalacija 1



Izvor: autor

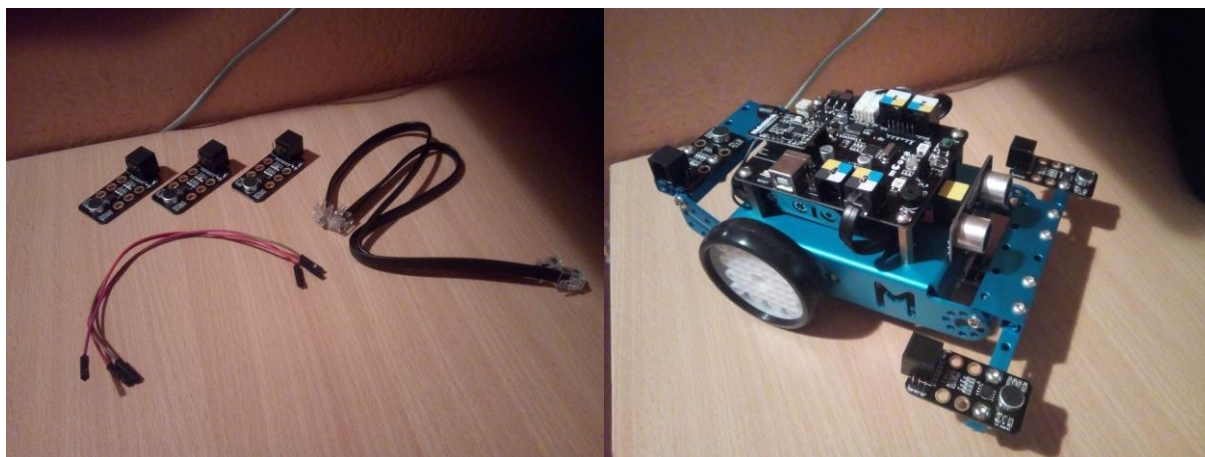
Kao sljedeći korak Me Sound Sensor mikrofoni se instaliraju na platforme i spajaju na mCore mikroprocesorsku ploču. U skladu s dizajnom robotskog rješenja senzori se instaliraju u trokutastoj formaciji; prednji senzori postavljaju se na krajeve široke prednje platforme, a stražnji se postavlja na stražnju, orijentiran tako da se mikrofona senzora nalazi u centru platforme. RJ25 konektor stražnjeg mikrofona orijentiran je prema desnoj strani robota kako bi se olakšalo spajanje s prikladnim RJ25 portom mCorea. Desni i stražnji senzori se spajaju s mCoreom RJ25 kabelima na prikladne RJ25 konektore, a lijevi sensor se žicama spaja direktno na Arduino pinove mCorea. Popis korištenih dijelova prikazan je u tablici 10, a instalacija i spajanje Me Sound Sensora prikazani su na slici 15.

Tablica 10 - dijelovi 2

Alati i dijelovi	Količina
Makeblock Me Sound Sensor	3
RJ25 kabel	2
Žice	3

Izvor: autor

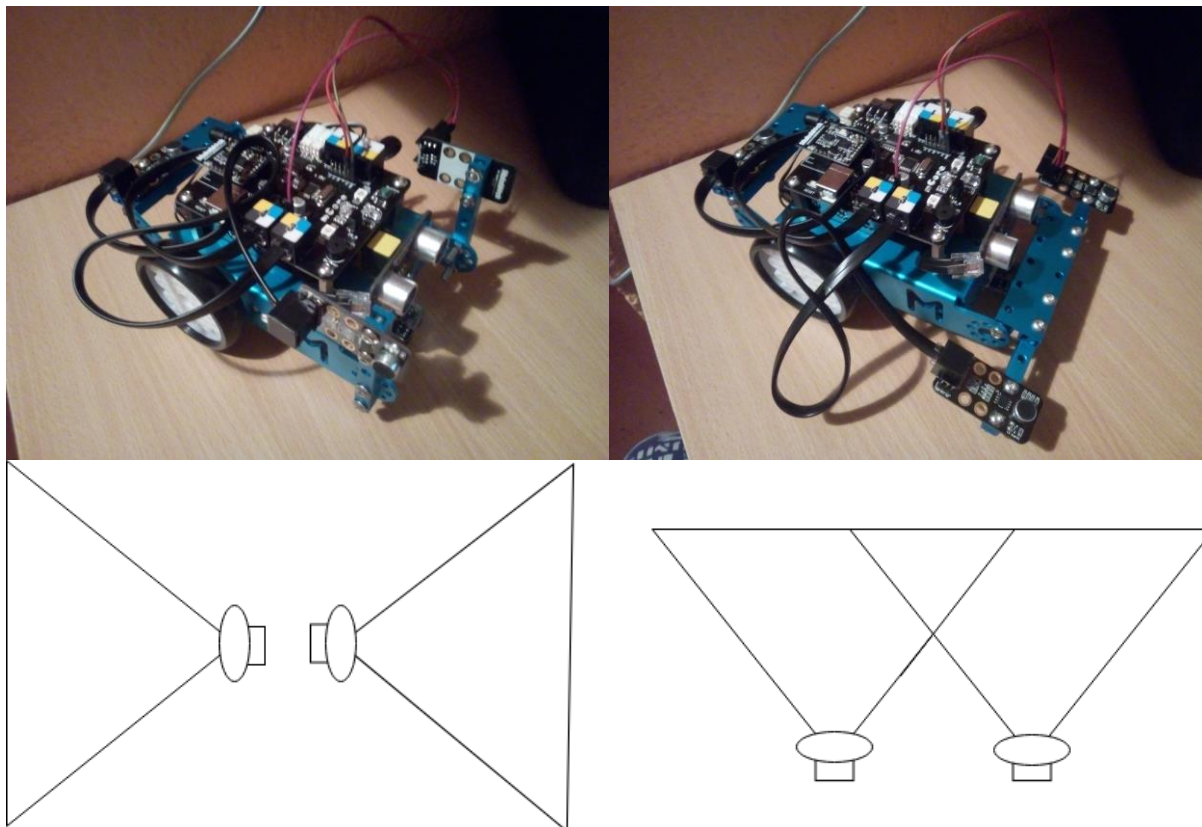
Slika 15 - instalacija 2



Izvor: autor

Tokom implementacije testirala su se dva različita položaja mikrofona: horizontalni okrenuti prema gore i vertikalni okrenuti prema van, prikazani na slici 16.

Slika 16 - usporedba područja osjetljivosti vertikalno (slika lijevo) i horizontalno (slika desno) postavljenih senzora zvuka.

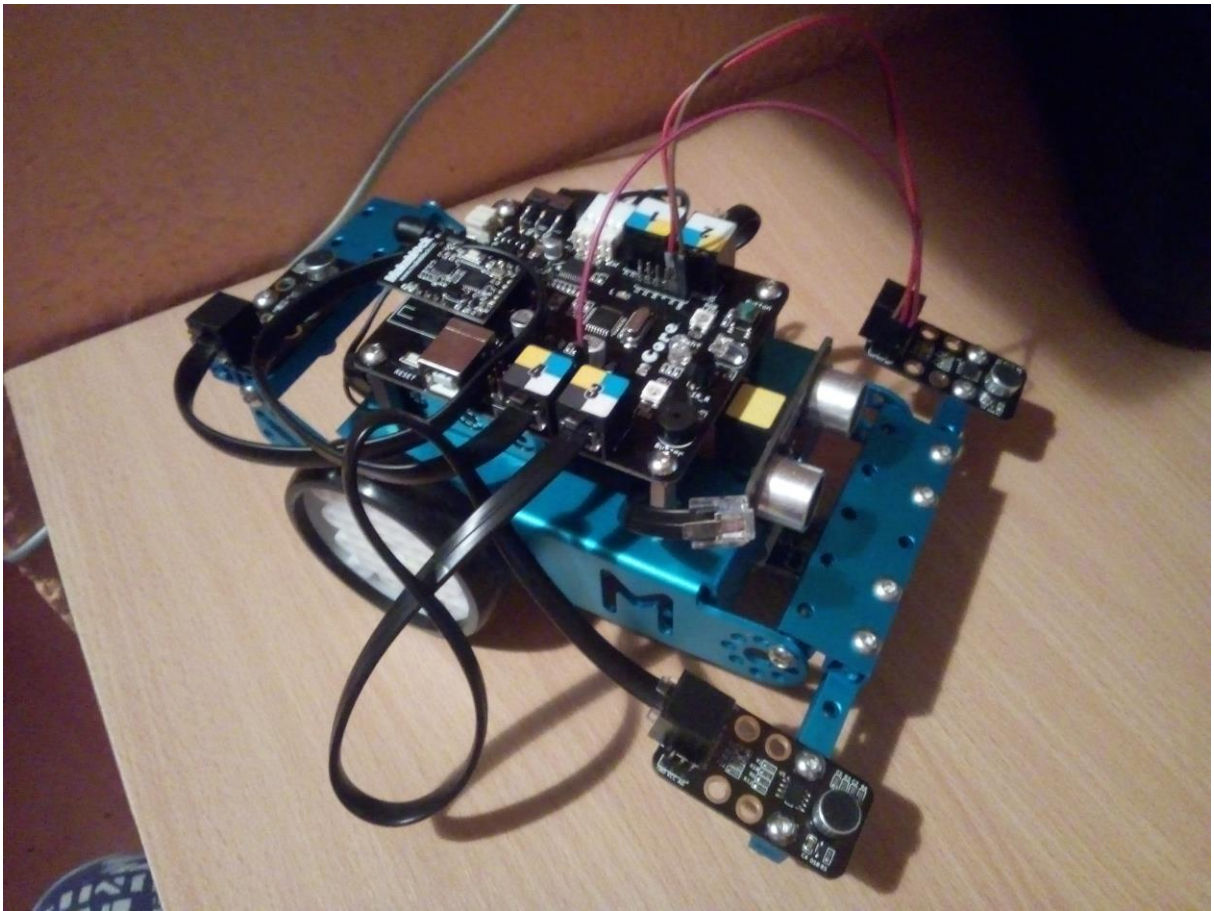


Izvor: autor

Oba položaja pokazivala su različite rezultate tokom implemetacije. Senzori su u vertikalnom položaju pokazivali veće razlike u vrijednostima intenziteta zvuka kada se izvor nalazio sa strane robota, što je olakšalo kalibriranje robota na zahtjeve pokretanja i skretanja. Horizontalni položaj je imao bolju pokrivenost u svim smjerovima te je u područjima pokrivenosti lijevog i desnog mikrofona postojalo preklapanje koje je korisno za određivanje smjera zvuka u sustavu trilateracije. S obzirom na navedene prednosti oba položaja na kraju je odabran horizontalni položaj; razlika u lakoći kalibriranja u vertikalnom položaju u odnosu na horizontalni nije toliko značajna dok su prednosti horizontalnog položaja bitnije za izračunavanje smjera zvuka.



Slika 17 - kompletno instalirano rješenje



Izvor: autor

Konačni rezultat je potpuno hardversko rješenje Robota pratioca zvuka prema ustanovljenom dizajnu robotskog rješenja, prikazan na slici 17.

Sveukupni dijelovi korišteni u izradi su mBot robot set, široke i tanke grede te vijci i matice iz mBot seta, Me Sound Sensori iz Makeblock Interactive Light and Souns seta dodatka za mBot, RJ 25 kabeli i žice. Popis svih korištenih dijelova prikazan je u tablici 11.

Tablica 11 - ukupni djelovi

Ukupni Alati i djelovi	Količina
MBot robot set	1
MBot set široka greda	2
MBot set tanka greda	2
MBot set vijci	8
MBot set matice	8
Makeblock Me Sound Sensor	3
RJ25 kabel	2
Žice	3

Izvor: autor

### 4.3. Programiranje robota

Programiranje robota izvedeno je prema ustanovljenom dizajnu programskog rješenja prikazanom na slici 11 i unutar tekstualnog okvira Kod 4. Robot je programiran u Arduino IDE razvojnom okruženju. Arduino.h biblioteka sadrži sve objekte i funkcije za komuniciranje i upravljanje Arduino mikrokontrolerskim pločama.

Najprije su se dodale sve potrebne Arduino i Makeblock biblioteke. Biblioteka Wire.h omogućava komunikaciju putem I<sup>2</sup>C i TWI protokola. SoftwareSerial.h biblioteka omogućava serijsku komunikaciju s Arduinom na digitalnim pinovima. Pomoću te biblioteke moguće je koristiti više softverskih serijskih portova s brzinama do 115200 bitova po sekundi (bps). Na kraju, MeMCore.h biblioteka sadrži sve druge biblioteke potrebne za komunikaciju i upravljanje svim komponentama na mCore ploči. Biblioteke se u Arduino kod dodaju „#include“ direktivom. Poziv biblioteka unutar Arduino koda prikazana je unutar tekstualnog okvira Kod 5.

#### Kod 5 - biblioteke

```
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>
#include <MeMCore.h>
```

Izvor: autor



Unutar MeMCore biblioteke nalazi se MeDCMotor biblioteka s objektima za definiciju i funkcijama za upravljanje mBot motora. Objekt motora se definira naredbom „MeDCMotor imeMotora(uint8\_t port);“ gdje je port priključak na koji se definirani motor spaja. Motori se pokreću i postavlja im se snaga funkcijom „MeDCMotor::run(int speed)“ gdje je argument funkcije brzina (speed u pozivu funkcije), odnosno snaga koja se motoru postavlja, a zaustavljaju se funkcijom „MeDCMotor::stop()“ koja postavlja snagu na nulu. Funkcije se pozivaju naredbama „imeMotora.run(brzina);“ i „imeMotora.stop();“. U kodu programskog rješenja definirani su lijevi (motorL) i desni (motorR) motor kako je prikazano unutar tekstualnog okvira Kod 6.

Kod 6 - inicijalizacija objekata motora

```
MeDCMotor motorL(9);  
MeDCMotor motorR(10);
```

Izvor: autor

Kako bi se pristupilo Me Sound Sensor mikrofonima moraju se definirati pinovi koji se koriste te načini rada istih. Definiranje arduino pinova izvodi se unutar „setup()“ funkcije. Pinovi i njihovi načini rada definiraju se funkcijom „pinmode(pin, načinRada);“ gdje se pod „pin“ definira koji se pin koristi, a način rada može biti „INPUT“ ili „OUTPUT“. Budući da se podaci sa Me Sound Sensor mikrofona dohvaćaju, način rada postavlja se na „INPUT“. U skladu s dizajnom i implementacijom robotskog rješenja (slika 10) koriste se analogni pinovi A1, A2 i A3. Definirane su i globalne varijable sensorL, sensorD i sensorS u kojima će se držati vrijednosti dohvaćene sa senzora, kako je prikazano unutar tekstualnog okvira Kod 7.

Kod 7 - inicijalizacija varijabli i setup() funkcija

```
int sensorL, sensorD, sensorS;

void setup(){
  pinMode(A1,INPUT); //stražnji
  pinMode(A2,INPUT); //lijevi
  pinMode(A3,INPUT); //desni
}
```

Izvor: autor

U sklopu sa funkcijama definiranim tokom analize problema (slika 1, slika 2) definirane su tri funkcije koje će obavljati prikupljanje informacija sa senzora, izračunavanje smjera zvuka te pokretanje i zaustavljanje robota prema dizajnu programskog rješenja. Funkcija za prikupljanje informacija sa senzora definirana je kao „slušajZvuk“. SlušajZvuk dohvaća vrijednosti sa senzora zvuka i vraća pozitivnu ili negativnu logičku vrijednost ovisno o tome jesu li vrijednosti veće od zadane (375, prema pseudokodu unutar tekstualnog okvira Kod 2). Tokom testiranja senzora ustanovilo se da je stražnji senzor osjetljiviji od prednjih, zbog čega je potrebno smanjiti njegove vrijednosti. Budući da stražnji senzor u jednakim uvjetima vraća otprilike 35% veće vrijednost od ostalih, njegove vrijednosti potrebno je smanjiti za približno 25%. Funkcija za izračunavanje smjera zvuka definirana je kao „izračunajSmjer“. IzračunajSmjer uspoređuje vrijednosti na sva tri senzora i ovisno o rezultatu vraća informaciju u kojoj od četiri zone definirane dizajnom se zvuk nalazi. Funkcija za pokretanje robota definirana je kao „slijediZvuk“. SlijediZvuk ovisno o rezultatu izračunavanja smjera postavlja snage motora u skladu s dizajnom i pokreće robota u smjeru zvuka. Funkcije za postavljanje snage motora definirane su unutar MeDCMotor biblioteke i pozivaju se naredbom „imeMotora.run(brzina);“. Funkcija za zaustavljanje robota nazvana je „stani“. Implementacije funkcija “slušajZvuk”, “izračunajSmjer”, “slijediZvuk” i “stani” unutar Arduino koda prikazane su unutar tekstualnih okvira Kod 8, Kod 9, Kod 10 i Kod 11.

#### Kod 8 - funkcija slusajZvuk()

```
bool slusajZvuk(){
  senzorL=analogRead(A2);
  senzorD=analogRead(A3);
  senzorS=analogRead(A1)*0.75;
  if(senzorL > 250 || senzorD > 250 || senzorS > 250){
    return true;
  }else{
    return false;
  }
}
```

Kod 9 - funkcija izracunajSmjer()

```
int izracunajSmjer(){
    int rezultat;
    if(senzorL-senzorD>50){
        rezultat=2;
    }else if(senzorD-senzorL>50){
        rezultat=3;
    }else{
        if((senzorS)>max(senzorL, senzorD)){
            rezultat=4;
        }else{
            rezultat=1;
        }
    }
    return rezultat;
}
```

Izvor: autor

## Kod 10 - funkcija slijediZvuk()

```
void slijediZvuk(int rezultat){
  switch (rezultat) {
  case 1:
    motorL.run((9)==M1?-100:100);
    motorR.run((10)==M1?-100:100);
    break;
  case 2:
    motorL.run((9)==M1?-100:-100);
    motorR.run((10)==M1?-100:100);
    break;
  case 3:
    motorL.run((9)==M1?-100:100);
    motorR.run((10)==M1?-100:-100);
    break;
  case 4:
    if(senzorL>senzorD){
      motorL.run((9)==M1?-100:-100);
      motorR.run((10)==M1?-100:100);
    }else if(senzorL<senzorD){
      motorL.run((9)==M1?-100:100);
      motorR.run((10)==M1?-100:-100);
    }
    delay(150);
    break;
  }
}
```

Izvor: autor

#### Kod 11 - funkcija stani()

```
void stani(){
  motorL.run((10)==M1?-(0):(0));
  motorR.run((10)==M1?-(0):(0));
}
```

Izvor: autor

Na kraju, sve tri funkcije pozivaju se unutar Arduino glavne funkcije prema dizajniranom algoritmu i slijednom dijagramu prikazanom na slici 11. Implementacija unutar Arduino koda prikazana je u tekstualnom okviru Kod 12. Kompletan Arduino kod korišten u implementaciji priložen je na kraju rada.

#### Kod 12 - glavna loop() funkcija

```
void loop(){
  while(slusajZvuk())
  {
    int rezultat=izracunajSmjer();
    slijediZvuk(rezultat);
  }
  stani();
}
```

Izvor: autor

## 5. Uporaba sustava

Sustav se koristi na način da se robot postavi u početnu poziciju na dovoljno širokoj horizontalnoj ravnoj površini, robot se zatim upali i u njegovoj blizini se stvori izvor kontinuiranog zvuka. Površina na kojoj se robot kreće treba biti dovoljno široka kako bi se robotu omogućilo kretanje i okretanje, preporučena minimalna širina je priližno 54 cm, ili tri dužine robota. Izvor zvuka mora biti kontinuiran zbog načina na koji ga robot prati; robot se kreće samo dok čuje zvuk. Također, izvor zvuka se mora nalaziti u blizini robota i biti dovoljno glasan. Razlika između detektiranog intenziteta zvuka na sensorima robota smanjuje se udaljavanjem od robota, što otežava određivanje smjera izvora zvuka na većim udaljenostima. Nakon što je robot upaljen, postavljen u početni položaj i u njegovoj okolini se nalazi dovoljno glasan izvor zvuka, robot će se okretati i kretati u smjeru izvora.

## 6. Zaključak

Robot pratilac zvuka uspješno je implementiran. Sustav funkcionira onako kako se i prvobitno zamislilo; mBot reagira na zvukove u svojoj blizini, određuje iz kojeg smjera zvuk dolazi te je sposoban kretati se prema izvoru.

Sveukupna izgradnja robota trajala je priližno 13 sati. Na analizu problema i prijašnjih rješenja utrošeno je 2 sata. Dizajniranje rješenja trajalo je približno 4 sata, od čega su 2 utrošena na definiranje korištenih dijelova i njihove specifikacije, 1 sat na dizajn hardverskog rješenja i 1 sat na dizajn programskog. Implementacija je trajala najduže, približno 7 sati, od čega je konstrukcija robota bila jednostavna i potrajala samo 30 minuta, dok se ostalo vrijeme radilo na izradi, testiranju i prepravljanju programskog rješenja.

Pojavilo se nekoliko problema tokom implementacije. U početku koristila su se samo dva senzora zvuka, zbog čega robot nije bio sposoban odrediti dolazi li zvuk srijeda ili straga. Problem se riješio nakon analize prijašnjih rješenja dodavanjem trećeg mikrofona prema primjeru Blindbota. Sljedeći problem bilo je postojanje samo dva RJ25 konektora na mCoreu dok je bilo potrebno spojiti tri senzora zvuka. Problem se riješio spajanjem prva dva mikrofona preko RJ25 konektora i trećeg mikrofona direktno na Arduino pinove. Tokom pisanja programskog koda pojavio se problem s određenim Makeblock bibliotekama i funkcijama koje su bile loše implementirane ili zastarjele, namjenje starijim verzijama Arduino IDEa. Problem je riješen zaobilazanjem spomenutih funkcija i korištenjem drugih, sličnih funkcija koje su davale jednak rezultat. Drugi problem proizašao je iz činjenice da motori koji pokreću robota stvaraju dovoljno glasan zvuk da zatvore robota u beskonačnu petlju kretanja u krug. Problem se riješio podizanjem donje granice vrijednosti senzora koje se registriraju kao vanjski zvuk.

Konačno rješenje može biti poboljšano, umjesto korištenja razlike u intenzitetu zvuka pri određivanju smjera izvora može se koristiti mjerenje kašnjenja u vremenu detekcije zvuka između sva tri mikrofona. Također, područje oko robota moglo se podijeliti na više zona umjesto samo na 4. Te dvije promjene rezultirale bi sustavom sposobnim za preciznije određivanje smjera izvora zvuka.



## POPIS KRATICA I AKRONIMA

- CPU - Centraj Processing Unit
- MIT - Massachusetts Institute of Technology
- AI – Artificial Intelligence
- IDE - Integrated Development Environment
- USB - Universal Serial Bus
- STEM - Science, Technology, Engineering, Mathematics
- PWM - Pulse Width Modulation
- ICSP - In-Circuit Serial Programming
- SCL - Serial Clock Line
- SDA - Serial Data Line
- LDR - Light Dependent Resistor
- RISC - Reduced Instruction Set Computer
- MIPS – Million Instructions Per Second
- EEPROM - Electrically Erasable Programmable Read-Only Memory
- SRAM - Static Random-Access Memory
- RTC - Real Time Counter
- USART - Universal Synchronous and Asynchronous Receiver-Transmitter
- ADC – Analog-to-Digital Converter
- SPS – Samples Per Second
- I<sup>2</sup>C - Inter-Integrated Circuit
- TQFP - Thin Quad Flat Package
- QFN/MLF - Quad Flat No-lead/Micro Lead-Frame
- SPI - Serial Peripheral Interface
- GUI - Graphic User Interface
- NC - Not Connected

## POPIS LITERATURE

### Internet

1. Encyclopaedia Britannica, Robot, lipanj 2018.,  
<https://www.britannica.com/technology/robot-technology>
2. Black P., Sound Following Rover, Lipanj 2018.,  
<http://randomagglomeration.blogspot.hr/p/sound-following.html>
3. Shah A., D., Blindbot (Sound Follower Robot), lipanj 2018.,  
[https://drive.google.com/file/d/0B\\_9bbqvqxo2MTDhrcXQyU0NZVkk/view](https://drive.google.com/file/d/0B_9bbqvqxo2MTDhrcXQyU0NZVkk/view)
4. GISGEography, Trilateration vs Triangulation – How GPS Receivers Work,  
<https://gisgeography.com/trilateration-triangulation-gps/>
5. Jensa, Flashgamer, Makeblock mBot review, 2. studeni 2016.,  
<http://flashgamer.com/hardware/comments/makeblock-mbot-review>
6. Atmel, ATmega328/P datasheet, studeni 2016.,  
[http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P\\_Datasheet.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf)
7. Makeblock, Me Sound Sensor overview, 15. studeni 2017.,  
[https://media.digikey.com/pdf/Data%20Sheets/Makeblock%20PDFs/Me\\_Sound\\_Sensor\\_Web.pdf](https://media.digikey.com/pdf/Data%20Sheets/Makeblock%20PDFs/Me_Sound_Sensor_Web.pdf)
8. Texas Instruments, LMx58 Low-Power, Dual-Operational Amplifiers, revizija, studeni 2014.,  
<http://www.ti.com/lit/ds/symlink/lm158-n.pdf>

## POPIS TABLICA

Tablica 1 - specifikacije mBota .....	9
Tablica 2 - specifikacije mCore-a.....	10
Tablica 3 - konekcije mCore RJ25 pinova .....	10
Tablica 4 - povezivanja mBot portova .....	11
Tablica 5 - specifikacije ATmega328 mikrokontrolera.....	12
Tablica 6 - pinovi Me Sound Sensor mikrofona .....	14
Tablica 7 - tehničke specifikacije Me Sound Sensor mikrofona.....	14
Tablica 8 - povezivanje Arduino i mikrofon pinova .....	20
Tablica 9 - djelovi 1.....	27
Tablica 10 - djelovi 2.....	28
Tablica 11 - ukupni djelovi.....	31

## POPIS SLIKA

Slika 1 - osnovni dijagram funkcija.....	2
Slika 2 - konačni dijagram funkcija.....	4
Slika 3 - Sound Following Rover .....	5
Slika 4 - Blindbot.....	6
Slika 5 - primjer trilateracije.....	7
Slika 6 - Me Sound Sensor shema .....	13
Slika 7 - Arduino IDE prozor .....	16
Slika 8 – osnovne funkcionalnosti.....	18
Slika 9 - shema robotskog rješenja .....	19
Slika 10 - shema povezivanja hardwarea .....	20
Slika 11 - dijagram toka .....	21
Slika 12 - podjela na zone i primjena trilateracije .....	23
Slika 13 - mBot.....	26
Slika 14 - instalacija 1 .....	27
Slika 15 - instalacija 2 .....	28
Slika 16 - usporedba područja osjetljivosti vertikalno (slika lijevo) i horizontalno (slika desno) postavljenih senzora zvuka.....	29
Slika 17 - kompletno instalirano rješenje .....	30

## POPIS KODOVA

Kod 1 – pseudokod 1 .....	21
Kod 2 – pseudokod 2 .....	22
Kod 3 – pseudokod 3 .....	24
Kod 4 – pseudokod 4 .....	25
Kod 5 - biblioteke .....	32
Kod 6 - inicijalizacija objekata motora.....	33
Kod 7 - inicijalizacija varijabli i setup() funkcija.....	34
Kod 8 - funkcija slusajZvuk().....	35
Kod 9 - funkcija izracunajSmjer() .....	36
Kod 10 - funkcija slijediZvuk() .....	37
Kod 11 - funkcija stani().....	38
Kod 12 - glavna loop() funkcija .....	38

## POPIS PRILOGA

Prilog 1 - Arduino kod.....	45
-----------------------------	----

## Prilog 1 - Arduino kod

```
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>
#include <MeMCore.h>

MeDCMotor motorL(9);
MeDCMotor motorR(10);

int senzorL, senzorD, senzorS;

void setup(){
  Serial.begin(9600);
  pinMode(A1,INPUT); //stražnji ss3
  pinMode(A2,INPUT); //lijevi ss1
  pinMode(A3,INPUT); //desni ss2
}

bool slusajZvuk(){
  senzorL=analogRead(A2);
  senzorD=analogRead(A3);
  senzorS=analogRead(A1)*0.75; // stražnji senzor je osjetljiviji od prednjih, potrebno je
  smanjiti njegove vrijednosti
  if(senzorL > 250 || senzorD > 250 || senzorS > 250){
    return true;
  }else{
    return false;
  }
}
```

```
int izracunajSmjer(){
    int rezultat;
    if(senzorL-senzorD>50){
        rezultat=2;
    }else if(senzorD-senzorL>50){
        rezultat=3;
    }else{
        if((senzorS)>max(senzorL, senzorD)){
            rezultat=4;
        }else{
            rezultat=1;
        }
    }
    return rezultat;
}
```

```

void slijediZvuk(int rezultat){
  switch (rezultat) {
  case 1:
    motorL.run((9)==M1?-(100):(100));
    motorR.run((10)==M1?-(100):(100));
    break;
  case 2:
    motorL.run((9)==M1?-(-100):(-100));
    motorR.run((10)==M1?-(100):(100));
    break;
  case 3:
    motorL.run((9)==M1?-(100):(100));
    motorR.run((10)==M1?-(-100):(-100));
    break;
  case 4:
    if(senzorL>senzorD){
      motorL.run((9)==M1?-(-100):(-100));
      motorR.run((10)==M1?-(100):(100));
    }else if(senzorL<senzorD){
      motorL.run((9)==M1?-(100):(100));
      motorR.run((10)==M1?-(-100):(-100));
    }
    delay(150);
    break;
  }
}

void stani(){
  motorL.run((10)==M1?-(0):(0));
  motorR.run((10)==M1?-(0):(0));
}

```

```
void loop(){  
  
    while(slusajZvuk()  
    {  
        int rezultat=izracunajSmjer();  
        slijediZvuk(rezultat);  
    }  
    stani();  
    delay(100);  
}
```