

# MIRANDA ROBOTSKI SIMULATOR

Priručnik robotike za mentore i učenike

Osnovne upute za rad u Edison robotu



Hrvatski robotički savez, Zagreb  
Udruga za robotiku „Inovatic“, Pmf - Split  
Izradio: Jozo Pivac, prof.

Split, 2020.

**Sadržaj:**

1. Predgovor.....	3
2. Uvod u Edison robota .....	4
2.1. Osnovne karakteristike robota .....	4
2.2. Karakteristike robota unutar simulatora .....	6
3. Osnovne naredbe za upravljanje robotom .....	9
4. Naredbe za led žaruljice .....	13
5. Ispis poruka u simulatoru .....	15
6. Upravljanje robotom pomoću senzora .....	17
6.1. Senzor za prepoznavanje linije .....	18
6.2. Senzor za praćenje linije .....	21
6.3. Senzor za prepoznavanje objekta .....	24

## 1. Predgovor

Roboti su sve više dio svakodnevnog života mlađih, posebno mlađe životne dobi, bilo kod kuće da se koriste kao sredstvo za igru i zabavu ili da se koriste kao edukacijski alati unutar pojedinih školskih i izvanškolskih aktivnosti. Roboti mogu imati veliki potencijal po pitanju edukacije djece jer sadrže zabavnu platformu unutar koje djeca mogu učiti razne sadržaje i razvijati korisne vještine poput razvijanja motoričkih sposobnosti, kreativnosti, logičkog promišljanja, kritičkog pristupa rješavanju problema, suradnji i timskom radu. Učenje uz pomoć raznih robotičkih alata pruža djeci interdisciplinarni pristup učenju jer unutar robotičkih aktivnosti postoje elementi drugih predmeta poput matematike, fizike, informatike, tehničke kulture te sličnih područja prirodnih i tehničkih znanosti. Također robotika sadrži važnu socijalnu komponentu koja se razvija kroz potrebno suradničkog učenje koje je preduvjet za uspješan rast i razvoj te prilagođavanje pojedinca potrebama suvremenog društva.

Iako su robotika i STEM aktivnosti sve popularnije i potrebitije u obrazovanju mlađih, na žalost su nedostupne mnogim učenicima u obrazovanju zbog raznih ograničenja poput nedostatka potrebne opreme, prostora i vremena unutar školskog plana i programa. Stoga nam je želja ovim priručnikom omogućiti svim učiteljima i učenicima kojima je zbog pojedinih ograničenja nedostupna fizička oprema da kroz rad u ovom robotskom simulatoru učine robotiku dostupnom te im na takav način olakšamo uključenje u aktivnosti iz robotike.

Također svi oni učitelji i mentori raznih tehničkih udruga i škola koji su uslijed nastale pandemije sa Covid- 19 virusom bili prisiljeni obustaviti sav svoj rad i aktivnosti unutar svojih ustanova mogu pomoći simulatora iz robotike nastaviti svoje edukacije i natjecanja u online obliku.

Iako robotski simulatori ne mogu u potpunosti nadomjestiti fizički oblik rada sa robotom postoje određene prednosti učenja robotike uz simulator koje se naravno odlično mogu iskoristiti u već postojećim fizičkim radionicama iz robotike.

Ovdje ćemo istaknuti samo neke pozitivne učinke učenja robotike uz simulator;

- Smanjenje troškove nabavke robotičke opreme te potrošnje i skladištenja baterija.
- Pružanje učenicima rad na daljinu te dostupnost alata i znanja 0/24.
- Smanjuje „strah od neznanja“ posebno pri početnim koracima rada u robotici.
- Skraćuje potrebno vrijeme realizacije određenih zadataka.
- Virtualni 3D svijet u simulatoru pruža učenicima bržu i kreativniju izradu raznih robotskih arena i staza.

Ovaj priručnik je nastao u okviru pokretanja online natjecanja iz robotike od strane Hrvatskog robotičkog saveza i Udruge za robotiku „Inovatic“ pod nazivom *Robotičko simulacijska liga (ROSIL)*, a rezultat je stečenog iskustva i vođenja brojnih radionica i natjecanja u robotici.

Iako je priručnik usmjeren prvenstveno na „Edison“ platformu robota unutar „Miranda“ softvera za simulaciju robota može se većinu obrađenog sadržaja primijeniti i na ostale robotske platforme unutar samog simulatora.

Priručnik je namijenjen prvenstveno učiteljima i učenicima koji rade na popularizaciji robotike unutar svojih udruga, škola i drugih ustanova.

## 2. Uvod u Edison robota

Edison je edukacijski robot namijenjen poučavanju učenika u robotici, STEAM-u i računalnom načinu razmišljanja.

Namijenjen je različitim uzrastima učenika te ima široku mogućnost primjene, može se koristiti već od predškolske dobi pa sve do srednjoškolskog uzrasta.

Karakterizira ga jednostavnost u rukovanju, dobra otpornost na udarce te odlične radne performanse.

Softver za upravljanje je vrlo sofisticiran, prilagođen djeci te ne zahtjeva nikakvo dodatno instaliranje ili podešavanje već je sve moguće na vrlo jednostavan i pristupačan način koristiti preko online sučelja.

Slika 1.1 Fizički izgled Edison robota



Više o Edisonu možete pogledati na službenim stranicama proizvođača: <https://meetedison.com/why-edison-robot/>

### 2.1. Osnovne karakteristike robota

#### Programiranje;

Edison podržava više načina upravljanja (programiranja) ovisno o uzrastu i stupnju predznanja;

- Upravljanje preko „bar kodova“ - za djecu stariju od 4 god.
- Upravljanje preko programskih blokova - za djecu stariju od 7 god.
- Upravljanje unutar Scratch programskog okruženja - za djecu stariju od 10 god.
- Upravljanje unutar Python programskog okuženja - za djecu stariju od 13 god.

Više o mogućnostima programiranja na: <https://meetedison.com/robot-programming-software/>

## Dodatni setovi i dijelovi;

Edison se osim osnovnog modela robotskog vozila može nadograditi sa proširenim paketom različitih lego dijelova s kojima možete dobiti sasvim nove funkcionalne robotske modele.

Slika 1.2. Lego dodaci za Edison robote



## Senzori i motori;

Edison robot koristi određene senzore i motore kako bi se uspješno snalazio u vlastitom okruženju.

Koristi lijevi i desni motor koji su direktno povezani na bočna kola.

Od značajnijih senzora koristi infracrveni (IR) senzor za prepoznavanje blizine sa drugim robotima i objektima, koristi senzore za prepoznavanje različitog intenziteta svjetla, prepoznavanje tamnih linija ili praćenja same linije.

Također sadrži u sebi neke dodatno korisne elemente poput; botuna, led dioda (žaruljice), zvučnika i zvučnog senzora. IR senzor osim za prepoznavanja blizine drugih objekata može se koristiti i za primanje i slanje različitih kodiranih IR poruka za komunikaciju sa drugim robotima.

Više o samim senzorima na: <https://meetedison.com/edison-robots-sensors/>

Slika 1.3. Senzori i elementi robota



## 2.1. Karakteristike robota unutar simulatora

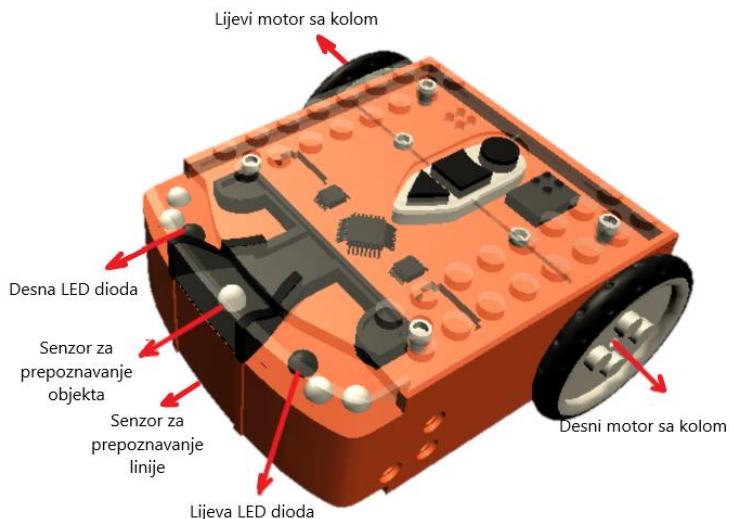
### Fizičke karakteristike robota;

Edison robot u „Miranda“ - robotskom simulatoru vrlo realistično simulira stvarni model Edison robota.

Robot unutar simulatora sadrži veliku većinu dijelova i senzora poput robota u fizičkom obliku;

- Dva motora direktno povezana sa bočnim kolima.
- Senzor za praćenje linije
- Senzor za prepoznavanje objekata
- Dvije prednje led diode (žaruljice)

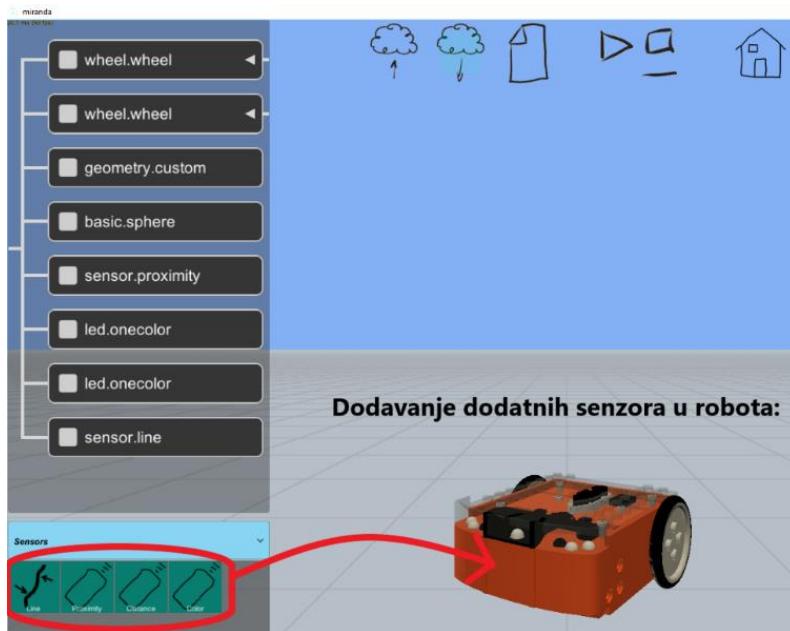
Slika 1.4. Glavni dijelovi i senzori unutar simulatora



Ono što je dodatna prednost simulatora je da se može po potrebi dodavati jedno ili više od navedenih senzora pa primjerice robot može umjesto dvije osnovne led diode imati tri ili više dioda, umjesto jednog senzora za praćenje linije možete sadržavati dva ili više takvih senzora te se također u simulatoru može robotu dodavati oni senzori i elementi koje inače nema niti fizički oblik robota poput;

- Senzora za prepoznavanje boje (*color sensor*)
- Senzora za mjerjenje udaljenosti (ultrasonic sensor)
- RGB led žarulje

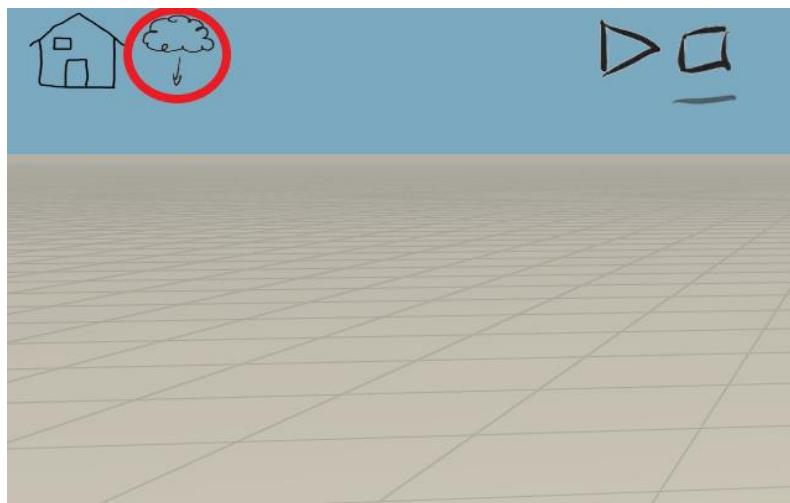
Slika 1.5. Mogućnost nadogradnje robota dodatnim senzorima



#### Programske karakteristike simulatora;

Da bi započeli programiranje robota u simulatoru najprije je potrebno odabrat određenu vježbu iz „Oblaka“.

Slika 1.6. Učitavanje vježbe iz oblaka



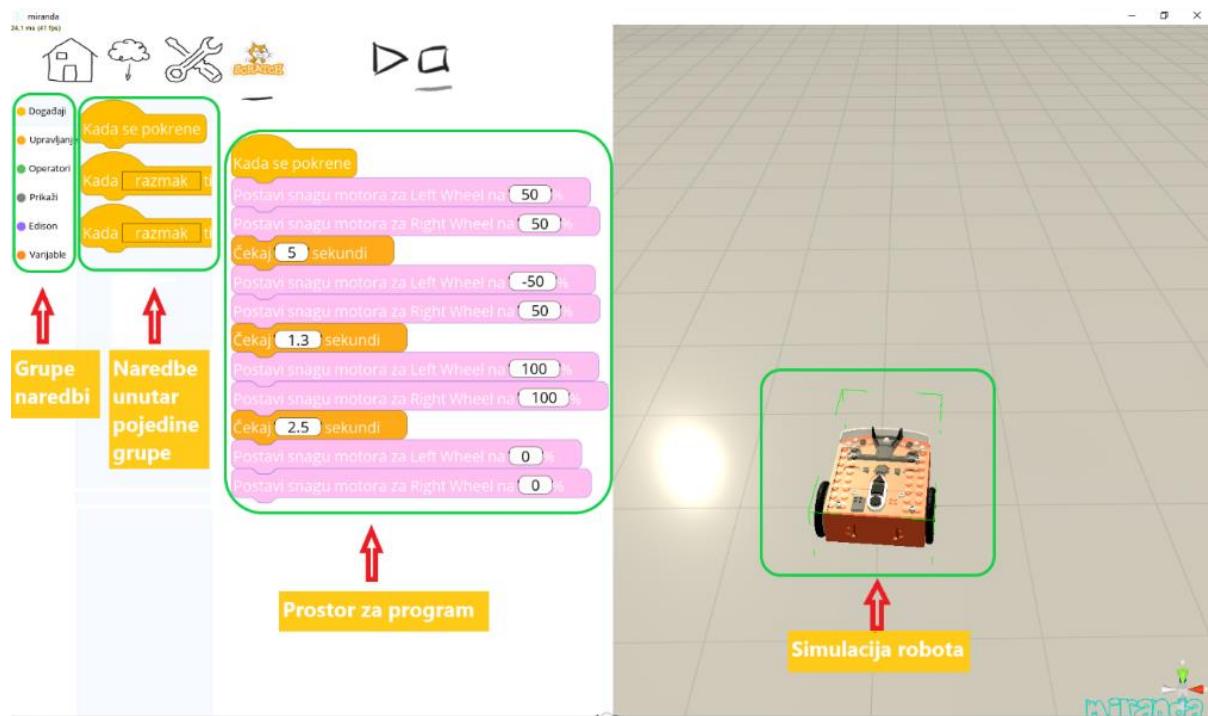
Svaka učitana vježba se sastoji od dva dijela prozora; lijevog dijela prozora predviđenog za izradu programa i desnog dijela prozora predviđenog za simulaciju robota.

Simulacija se pokreće klikom na ikonu oblika trokuta, u gornjem dijelu prozora, a zaustavlja se klikom na ikonu do nje u obliku kvadrata.

Ukoliko se klikne na robota unutar simulacije otvorit će se dio prozor za izradu programa.

Ako se klikne bilo gdje u prostor simulacije prebacujete se u simulaciju sa prikazom simulacije preko cijelog ekrana.

Slika 1.7. Izgled programa i simulacije unutar simulatora



Istovremeno se može imati ekran podijeljen na dva dijela kao što je prikazano na gornjoj slici što omogućuje istovremeno praćenje izvršavanja naredbi programa i praćenje rada robota u simulaciji.

Program koji se napravi u simulatoru se automatski sprema i svaki put učita zadnja verzija programa kada ponovno iz „Oblaka“ odaberete istu vježbu.

Programiranje u simulatoru je podržano u programskom okruženju na bazi Scrtacha i Pytona te se može odabrati programiranje robota u jednom od ta dva programska jezika ukoliko su vam oba ponuđena.

Slika 1.8. Odabir vrste programskog okruženja



Programski jezik Scratch zbog svoje popularnosti i vizualnog oblika naredbi u obliku „naredbenih blokova“ ili „puzla“ je prihvatljiviji za učenike mlađe dobi ili početnike, dokle za nešto starije i naprednije učenike Python bi bio prikladniji odabir programskog jezika gdje se programira unoseći naredbe i funkcije u obliku raznih sintaksi.

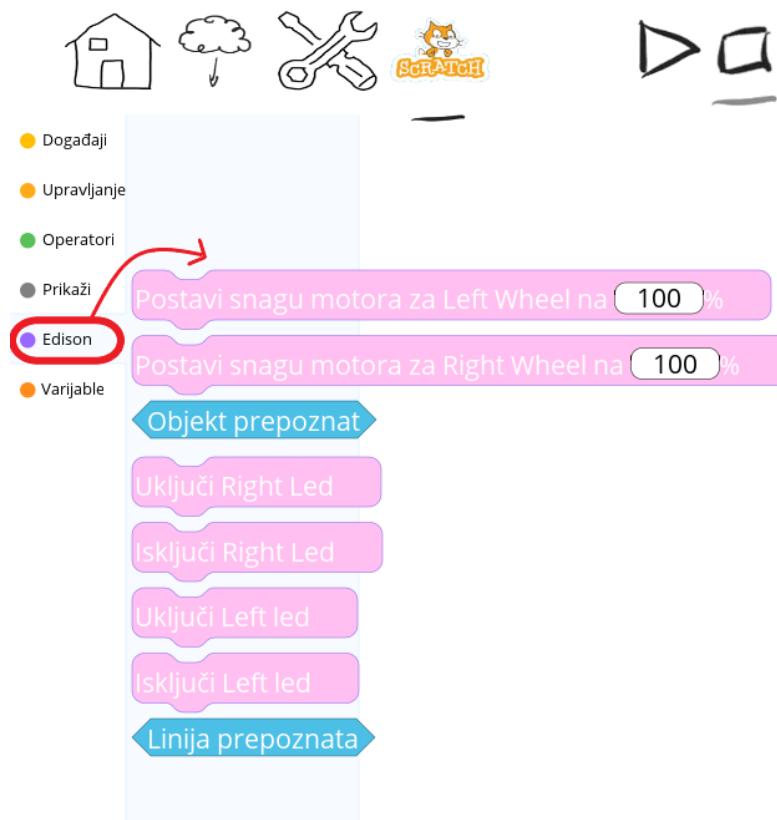
Pošto je ovaj priručnik namijenjen početnicima i temelji se na osnovama u nastavku sva pojašnjena naredbi i programa će biti pokazane unutar Scratch programskog okruženja.

### 3. Osnovne naredbe za upravljanje robotom

Kao što smo već spomenuli programiranje Edison robota je moguće unutar simulatora koristeći programske naredbe bazirane na Scratch programskom jeziku.

Zapravo većinu grupa naredbi su identične onima u Scratchu uz neke dodatne naredbe karakteristične za Edison robota poput naredbi za pokretanje motora, naredbi za prikaz i naredbi za rad sa senzorima.

Slika 1.9. Edison grupa naredbi



Možemo početi najprije od naredbi za pokretanje robota.

Edison robot u sebi sadrži dva motora (lijevi i desni) koja su direktno povezana na kola robota.

Da bi upravljali našim robotom potrebno je odrediti smjer i snagu lijevog i desnog motora.

Pokažimo to na par primjera;

## Primjer 1 - pokretanje robota naprijed

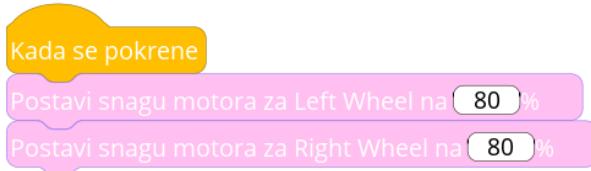
Ako želimo da se naš robot kreće ravnim smjerom naprijed potrebno je oba motora postaviti u isti smjer i istom snagom.

Smjer okretanja motora definiramo pozitivnim ili negativnim brojevima od 0 do 100.

Pozitivan smjer motora će pokretati robota naprijed, a negativan nazad.

Ako želimo da se naš robot kreće naprijed potrebno je snagu oba motora postaviti u jednakе pozitivne vrijednosti.

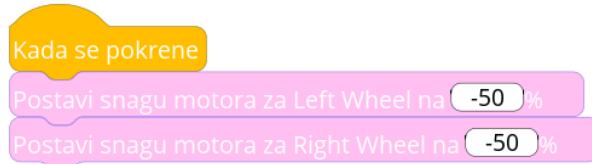
Slika 2.0. Program za pokretanje robota naprijed.



## Primjer 2 - pokretanje robota unazad

Ako želimo se da se naš robot kreće unazad potrebno je da oba motora imaju jednaku negativnu vrijednost snage.

Slika 2.1. Naredbe za pokretanje robota unazad

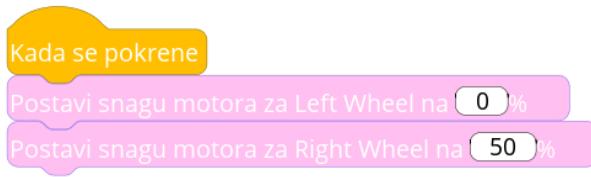


## Primjer 3 - skretanje robota

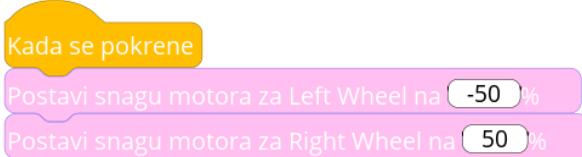
Ukoliko želimo da naš robot skreće u željenom smjeru moguća su dva načina;

1. Način je da se robotu zaustavi jedan motor (postavi u vrijednost „0“), a da se suprotnom motoru postavi određena pozitivna ili negativna vrijednost. U tom slučaju robot će skretati u željenom smjeru, ali sa određenim blagim pomakom prema naprijed ili nazad ovisno o smjeru snage motora kojim se giba.
2. Način je da se robotu postavi jednakе vrijednosti snage motora, ali suprotnog smjera. U tom slučaju će se robot rotirati u željenom smjeru ali bez blagog pomaka unaprijed ili unazad, nego će se rotirati u mjestu, tj. oko svoje osi.

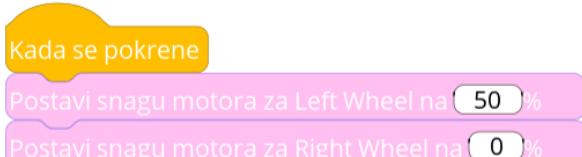
Slika 2.2. Skretanje u lijevo sa blagim pomakom unaprijed



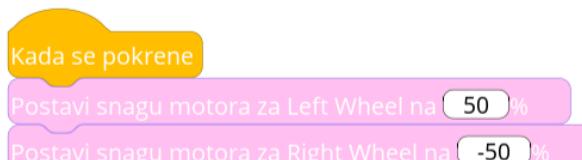
Slika 2.3. Skretanje u lijevo sa rotacijom u mjestu



Slika 2.4. Skretanje u desno sa blagim pomakom unaprijed

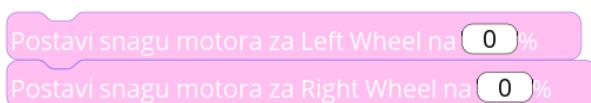


Slika 2.5. Skretanje u desno sa rotacijom u mjestu



Ukoliko želimo da se robot nakon određenog gibanja zaustavi potrebno je dodati naredbe za zaustavljanje, tj. postaviti oba motora u snagu „0“.

Slika 2.6. Naredbe za zaustavljanje robota



Također osim postavljanja određenog smjera i snage motora često je potrebno robotu definirati i vrijeme prijeđenog puta ili rotacije koju želimo sa robotom postići.

Za definiranje dužine puta ili kuta rotacije koristimo naredbu za vrijeme koju u programu postavljamo uvijek nakon naredbi za snagu lijevog i desnog motora.

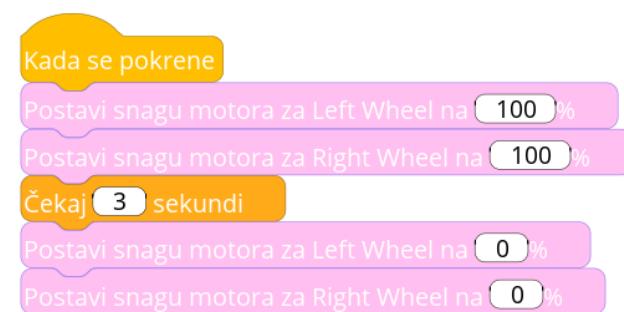
Slika 2.7. Naredba za postavljanje vremena



Naredba za postavljanje vremena ČEKAJ-BROJ-SEKUNDI se nalazi u grupi naredbi UPRAVLJANJE. U naredbu za vrijeme postavlja se određena brojčana vrijednost u sekundama i može biti definirana na najviše dvije decimale, npr. „2.45“ sekundi.

#### Primjer 4 - Postavljanje vremena gibanja robota

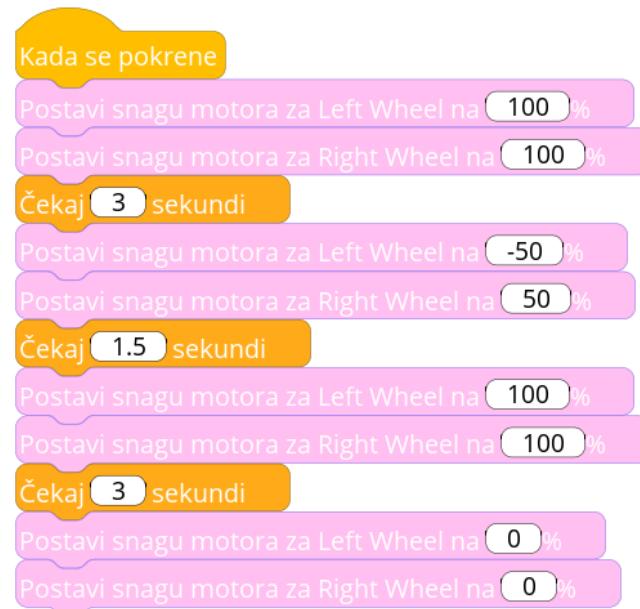
Npr. ako želimo napraviti program koji će pokretati robota naprijed vremenski 3 sekunde i da se nakon toga robot zaustavi naš program bi izgledao ovako;



Za dobiti željenu rotaciju robota pod određenim kutom potrebno je također koristiti naredbu za vrijeme koju postavljamo iza naredbi za snagu i smjer motora.

### Primjer 5 - Postavljanje vremena skretanja robota

Npr. ako želimo nadograditi gornji primjer programa tako da nakon što robot prijeđe put od 3 sekunde da se okreće za otprilike 180° stupnjeva u lijevo i vrati nazad do početnog položaja, naš program bi izgledao ovako;

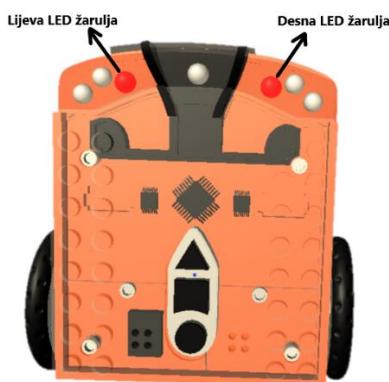


Primjećujemo u programu da smo odabrali vrijednost od „1.5“ sekunde u naredbi za vrijeme skretanja robota. Testiranjem ponašanja robota u simulaciji smo došli do rezultata da je to vrijeme potrebno da bi se robot otprilike zakrenuo za 180°. Naravno da pravilo uvijek vrijedi što je veća snaga motora potrebno vrijeme je manje i obratno.

## 4. Naredbe za led žaruljice

Edison robot sadrži dvije led žaruljice sa prednje lijeve i desne strane. Te mu žaruljice omogućuju određenu signalizaciju. Naredbe za uključivanje i isključivanje led žaruljica možemo pronaći u grupi naredbi „Edison“.

Slika 2.8. Prikaz lijeve i desne led diode (žarulje)



## Primjer 1

Napravimo program koji će istovremeno uključiti lijevu i desnu led žarulju na vrijeme od jedne sekunde i potom ih obje isključiti;



## Primjer 2

Napravimo program koji će tri puta zaredom uključivati i isključivati istovremeno obje led žarulje u vremenskom razmaku od jedne sekunde. Kada želimo u programu određeni broj puta ponavljati iste naredbe poželjno je koristiti naredbu PONOVI-BROJ-PUTA (*Repeat-number-times*);



## Primjer 3

Napravimo program koji će uključivati i isključivati obje led žarulje istovremeno u vremenskom razmaku od jedne sekunde cijelo vrijeme sve dokle je simulator pokrenut. Za postići stalno ponavljanje sljeda naredbi potrebno je koristiti naredbu STALNO PONAVLJAJ (*Repeat/Forever*)



## Primjer 4

Prethodni primjeri upravljanja led žaruljicama unutar robota su bili na automatskom principu izvršavanja.

Ukoliko želimo upravljati led žaruljicama ručno možemo to postići preko tipki sa tipkovnice. Za to je potrebno iz grupe naredbi DOGAĐAJI odabrati sljedeće naredbe; KADA JE- TIPKA- PRITISNUTA (*When key is pressed*) i KADA JE- TIPKA-OTPUŠTANA (*When key is released*).

Napravimo program koji će kada se pritisne tipka „strelice u lijevo“ uključiti lijevu žaruljicu, a kada se pritisne tipka „strelice u desno“ uključiti desnu žaruljicu. Također je potrebno definirati da kada se spomenute tipke otpustete da se određene žaruljice isključe.



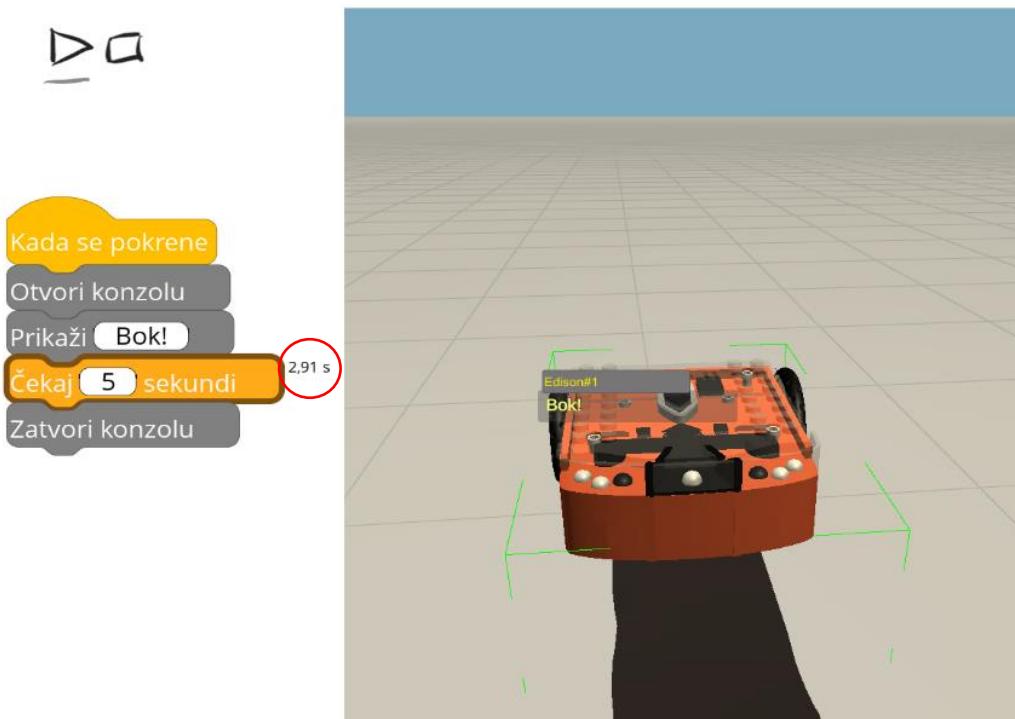
## 5. Ispis poruka u simulatoru

Edison robot nema fizički u sebi ugrađen ekran (*display*) kao što to imaju neki roboti, ali unutar simulacije možemo preko naredbi koristiti robotsku „konzolu“ unutar koje možemo programirati da se ispisuju; tekstualne poruke, varijable ili očitana stanja senzora.

## Primjer 1

Napravimo program koji će kada se robot u simulaciji pokrene ispisati pozdravnu riječ „Bok!“ u trajanju od pet sekundi i nakon toga se poruka više ne prikazuje;

Slika 2.9. Ispis poruke u simulatoru

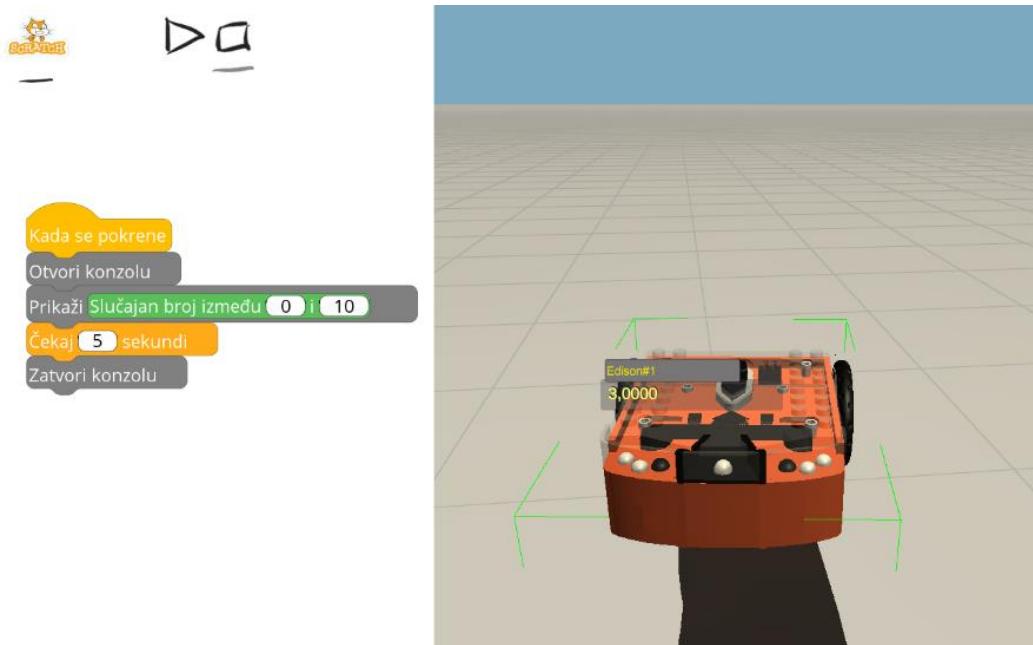


Uočimo da kada se pokrene program, pojavi se uz naredbu za vrijeme i „brojač“ koji odbrojava vrijeme sve dokle vrijeme unutar naredbe ne dosegne vrijednost od „0“ sekundi. Ovo može biti korisno kada želimo točno testirati neko vrijeme potrebno da robot izvrši određenu radnju.

## Primjer 2

Napravimo program koji će slučajnim odabirom izabrati neki broj između 1 i 10 i ispisati ga u konzoli vremenski 5 sekundi. Naredba za SLUČAJAN BROJ se nalazi u grupi naredbi OPERATORI. U našem primjeru program je generirao slučajan broj „3“.

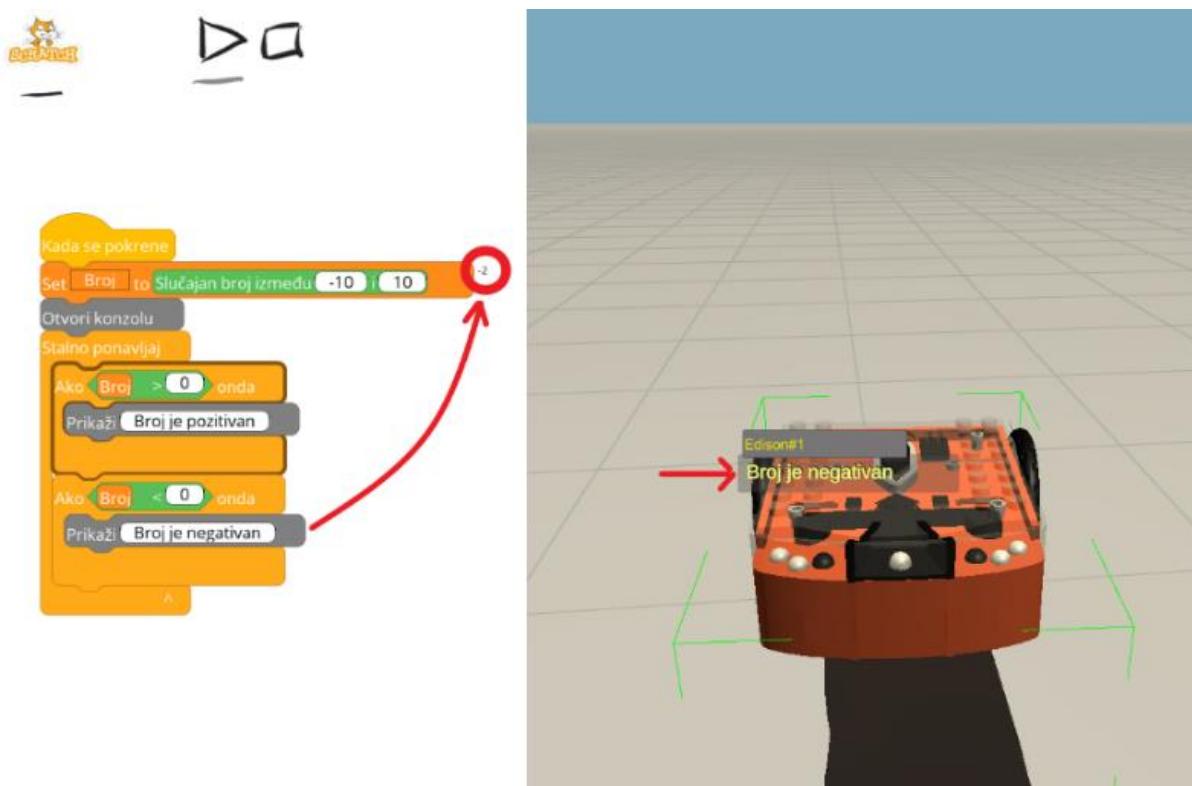
Slika 3.0. Ispis slučajnog broja u simulatoru



### Primjer 3.

Napravimo program koji će slučajnim odabirom izabrati neki broj između -10 i 10 te ispisati u konzoli da li je izabrani broj pozitivan ili negativan.

Slika 3.1. Ispis varijabli u simulatoru



Za potrebe ovog zadatka smo unutar programa koristili varijablu koju smo nazvali imenom „Broj“. Varijable stvaramo i dodajemo unutar grupe naredbi VARIJABLE klikom na „+“.

Zatim odabiremo iz grupe naredbi OPERATORI naredbu za USPOREĐIVANJE VEĆE/MANJE vrijednosti dobivenog broja koju stavljamo kao uvjet unutar naredbi AKO-ONDA.

U našem primjeru slučajnim odabirom smo dobili broj „2“ i u konzoli robota nam se ispisao tekst „Broj je negativan.“

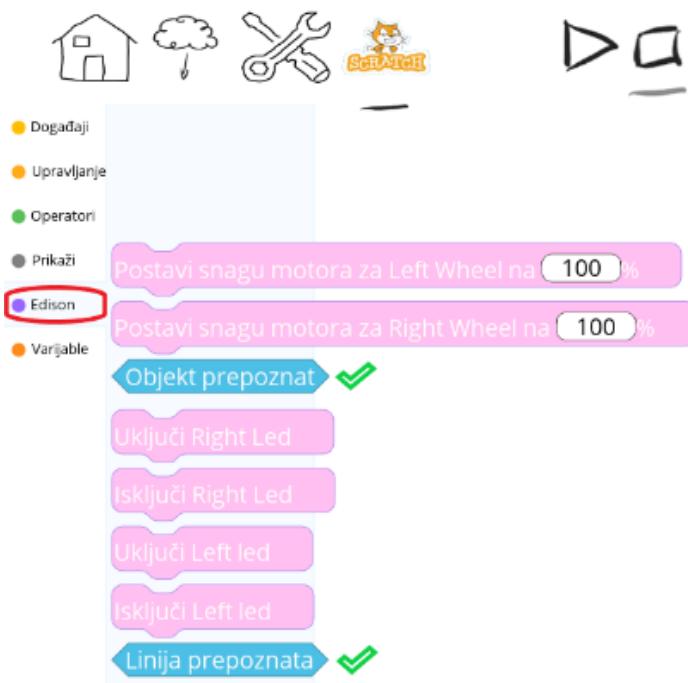
## 6. Upravljanje robotom pomoću senzora

U uvodnom dijelu smo spomenuli kako Edison robot u simulatoru koristi većinu senzora i dijelova kao i robot u fizičkom obliku. Također smo naveli kako se u simulatoru robotu mogu dodavati već postojeći senzori, ali i oni koje Edison inače ne sadrži poput; senzora za prepoznavanje boja (*color sensor*) i ultrazvučnog senzora (*ultrasonic sensor*).

U ovom priučniku ćemo se posvetiti samo onim senzorima koji dolaze sa osnovnom modelom Edison robota unutar simulatora.

U EDISON grupi naredbi se mogu pronaći naredbe za senzore.

Slika 3.2. Naredbe za upravljanje senzorima robota



## 6.1. Senzor za prepoznavanje linije

Edison robot ima sposobnost prepoznavanja tamnih i svijetlih boja. To svojstvo mu omogućuje tzv. „Senzor za prepoznavanje linije“ (*Line detected sensor*) koji se sastoji od dva elementa; LED diode i Fotootpornika.

Funkcija led diode je da svijetlom obasjava podlogu robota nad kojom se robot nalazi.

Fotootpornik služi za prepoznavanje intenziteta reflektirane svjetlosti od podloge.

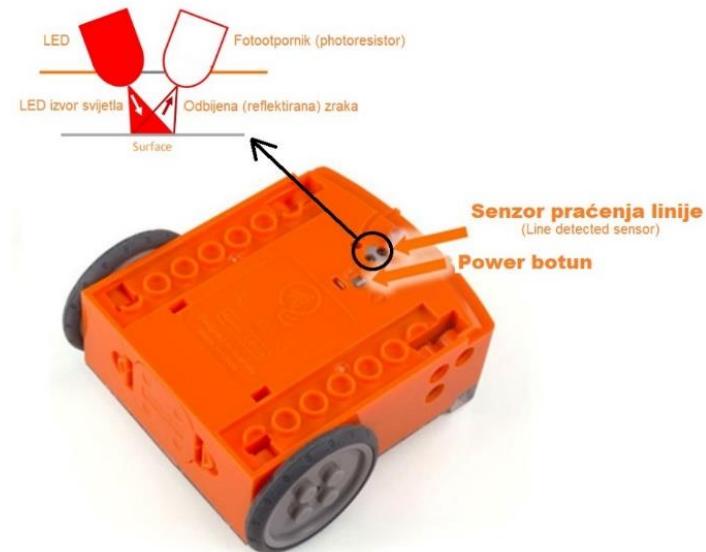
Intenzitet reflektirane svjetlosti neće biti jednak ukoliko se reflektira od tamnije podloge ili od svjetlijе podloge.

Razlog je što se zrake svijetle ne reflektiraju (odbijaju) jednako od tamnije podloge (npr. crna boja) i od svjetlijе podloge (npr. bijela boja). Točnije zrake svijetla se slabije reflektiraju od tamnih podloga nego od svijetlih.

To znači da ukoliko se robot nalazi nad tamnom podlogom (npr. crna linija), zrake svijetla obasjane od led diode neće biti reflektirane od podloge u dovoljnoj mjeri da ih fotootpornik očita te će u tom slučaju senzor zabilježiti da je prepoznao crnu liniju. Međutim ako se robot nalazi nad svjetlijom podlogom ili bojom, zrake svijetla će se većinom reflektirati (odbijati) od podloge i fotootpornik će zabilježiti povećanje svijetla, odnosno da nije prepoznao crnu liniju.

Fotootpornik na ovakav način zapravo radi kao senzor koji prepoznae promjenu intenziteta reflektiranog svijetla od tamnih i svijetlih podloga, odnosno u našem primjeru služi za prepoznavanje crne linije.

Slika 3.3. Izgled donje strane robota



Naredba za prepoznavanje linije je poprilično jednostavna i može se pronaći u grupi naredbi EDISON te izgleda ovako;

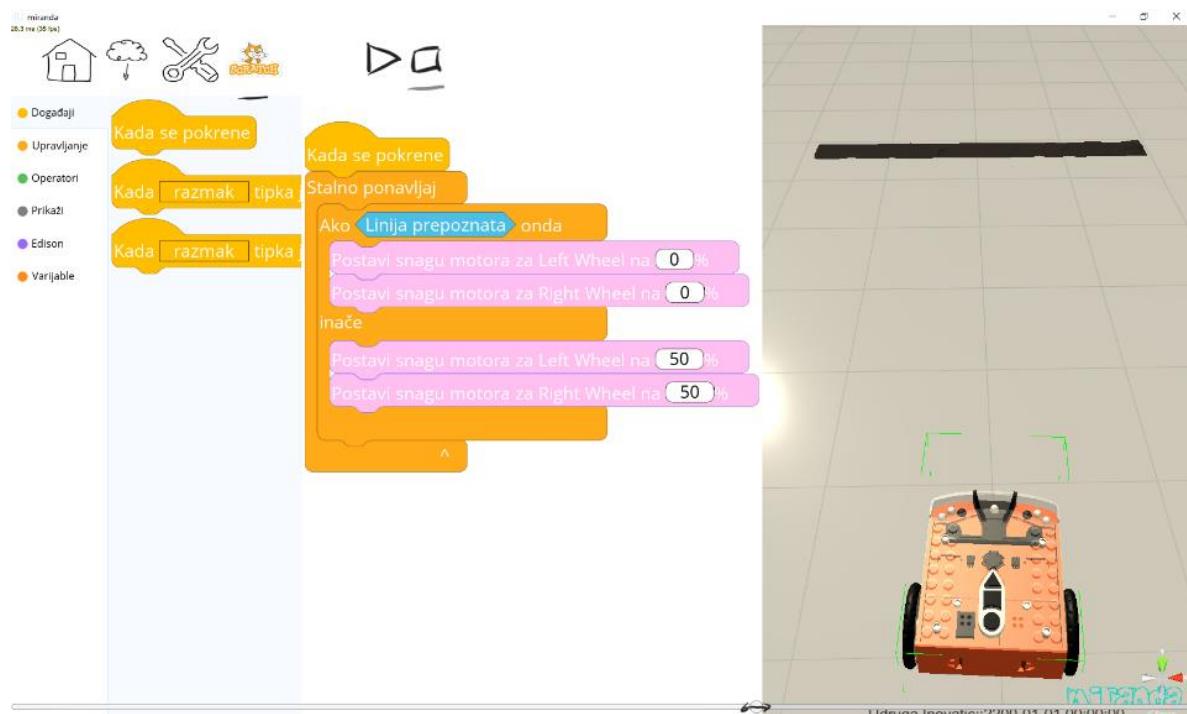
Linija prepoznata

Naredbu za prepoznavanje linije postavljamo unutar konteksta samog programa ovisno o tome što želimo postići senzorom za prepoznavanje linije.

### Primjer 1

Napravimo program koji će pokretati robota naprijed sve dokle robot ne prepozna crnu liniju nakon čega će se zaustaviti.

Slika 3.4. Prepoznavanje crne linije naredbom „Ako-onda-inače“



Unutar prikazanog programa koristimo sljedeće naredbe;

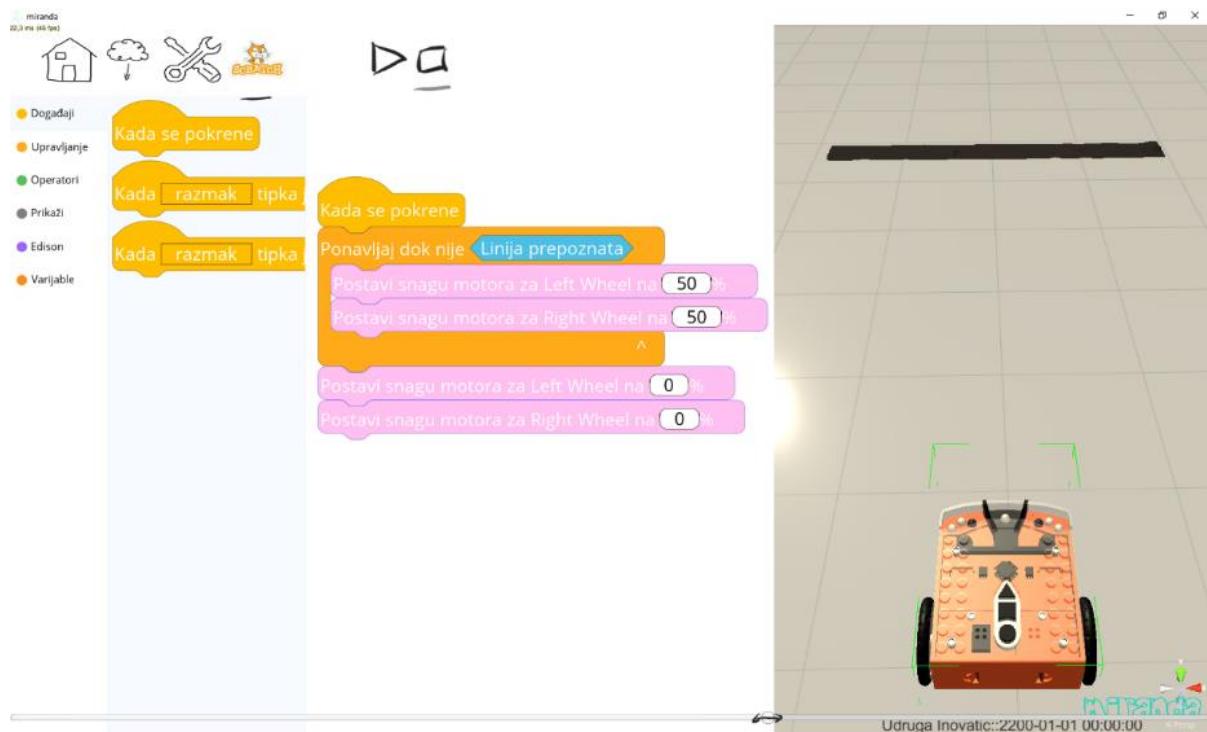
Naredbu STALNO PONAVLJAJ (*repeat/forever*) koristimo uvijek kada želimo da se određeni niz naredbi stalno ponavlja iz početka dokle god je robot pokrenut. U našem programu će se sve naredbe unutar naredbe „ponavljam“ ponavljati ispočetka dokle god ne zaustavimo simulaciju robota.

Naredbu AKO-ONDA-INAČE koristimo jer želimo postaviti određeni uvjet unutar programa (da li je linija prepoznata ili ne) te sukladno ispunjenju uvjeta da se određene naredbe izvršavaju ili ne izvršavaju.

U našem primjeru programa unutar uvjeta naredbe „ako-onda-inače“ smo postavili uvjet programu da ispituje da li je linija prepoznata. Ukoliko je uvjet ispunjen tj. istinit i linija prepoznata robot se zaustavlja pomoću naredbi za zaustavljanje lijevog i desnog motora, a ako uvjet nije ispunjen, tj. linija nije prepoznata robot će se kretati naprijed sve dokle senzor ne prepozna liniju i potom će se zaustaviti.

Također se ovaj isti zadatak mogao riješiti i na malo drugčiji način koristeći umjesto naredbe „Ako-onda-inače“ naredbu PONAVLJAJ-DOK NIJE (*repeat-until*).

Slika 3.5. Prepoznavanje crne linije naredbom „Ponavljam-dok nije“



Unutar naredbe „Ponavljam-dok nije“ se postavljaju one naredbe koje želimo da se stalno ponavljaju **dokle god uvjet nije ispunjen**. Ukoliko uvjet bude ispunjen izvršavaju se one naredbe ispod naredbe „Ponavljam-dok nije“ odnosno sljedeće naredbe u programu.

U našem slučaju dokle god ne bude ispunjen uvjet da je linija prepoznata robot će se kretati naprijed snagom oba motora „50“. Kad se uvjet ispunji i linija bude prepoznata program će postaviti snagu motora u „0“ i robot će se zaustaviti.

## Primjer 2

Napravimo sad program koji će pokretati robota unazad sve dokle senzor za prepoznavanje linije prepoznaće liniju. Nakon što linija ne bude više prepoznata robot će se zaustaviti i ispisati na ekranu „STOP“ te će uključiti obje LED žaruljice.



U našem programu koristimo iz grupe naredbi OPERATORI naredbu za negaciju tj. naredbu **NIJE (not)** koja u kombinaciji sa naredbom PONAVLJAJ DOK **NIJE** predstavlja dvostruku negaciju tih dviju naredbi, što zapravo rezultira naredbom „PONAVLJAJ DOK JE LINIJA PREPOZNATA“. Možemo i pojednostavljeno kazati da u ovom slučaju NIJE i NIJE daju zapravo JE.

Ovako postavljenim naredbama robot će se kretati unazad sve dokle uvjet bude ispunjen tj. dokle je linija prepoznata.

Isto rješenje se moglo dobiti i koristeći kombinaciju naredbi FOREVER i AKO JE-ONDA-INAČE.

## 6.2. Senzor za praćenje linije

Edison robot ima mogućnost praćenja linije pomoću senzora za prepoznavanje linije.

Kako točno radi senzor za prepoznavanje linije smo objasnili u prethodnom poglavlju.

Također smo prethodno naveli da Edison robot u fizičkom obliku kao i u simulatoru sadrži samo jedan senzor za praćenje linije koji se sastoji od jedne led diode i jednog fotootpornika.

Iako se problemi praćenja linija kod pojedinih robota recimo na svjetskim ili većim natjecanjima iz robotike rješavaju uz pomoć dva ili više senzora moguće je većinu staza za praćenje linija riješiti i sa samo jednim senzorom.

## Primjer 1

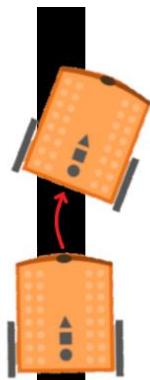
Ovdje ćemo prikazati i pojasniti načine praćenja linije jednim senzorom.

Kada robot sadrži samo jedan senzor za prepoznavanje linije moguća su jedino dva stanja;

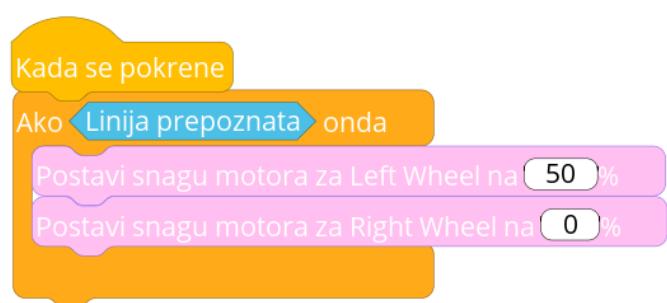
- A) Stanje kada senzor prepoznaje liniju
- B) Stanje kada senzor ne prepoznaje liniju

U stanju (A) možemo programirati robota da se zakreće blago u desno ukoliko mu senzor očita crnu liniju.

Slika 3.6. Robot na crnoj liniji (stanje A)

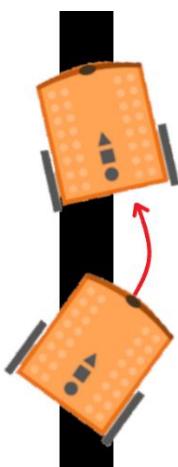


Slika 3.7. Program robota na crnoj liniji (stanje A)



U stanju (B) možemo programirati robota da se zakreće blago u lijevo ukoliko mu senzor ne očita crnu liniju te kako bi ga na takav način vratili ponovno na liniju.

Slika 3.8. Robot na bijeloj podlozi (stanje B)

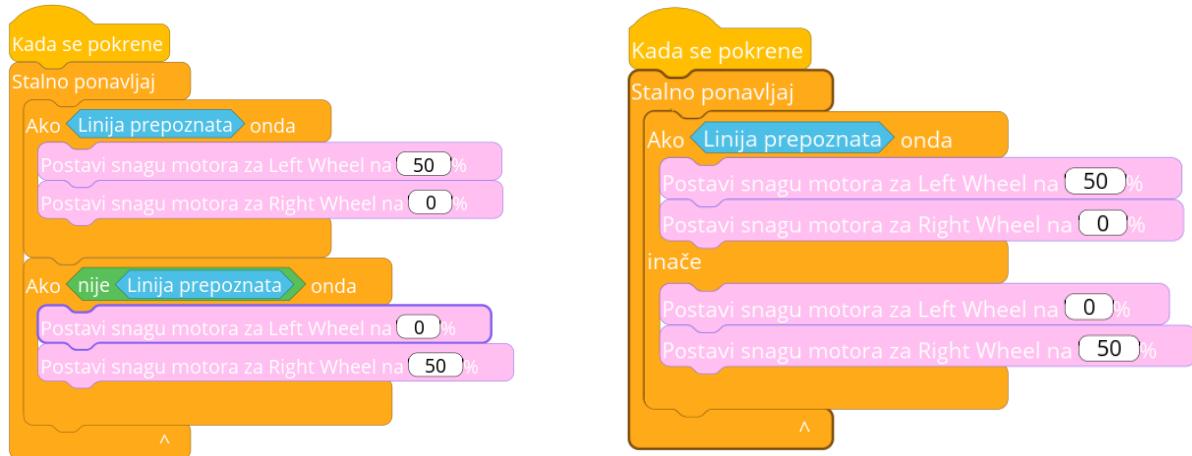


Slika 3.9. Program robota na bijeloj podlozi (stanje B)



Naš gotov primjer programa za praćenje linije može biti programiran na dva načina;

Slika 4.0. Praćenje desnog ruba linije naredbama AKO-ONDA ili naredbom AKO-ONDA-INAČE

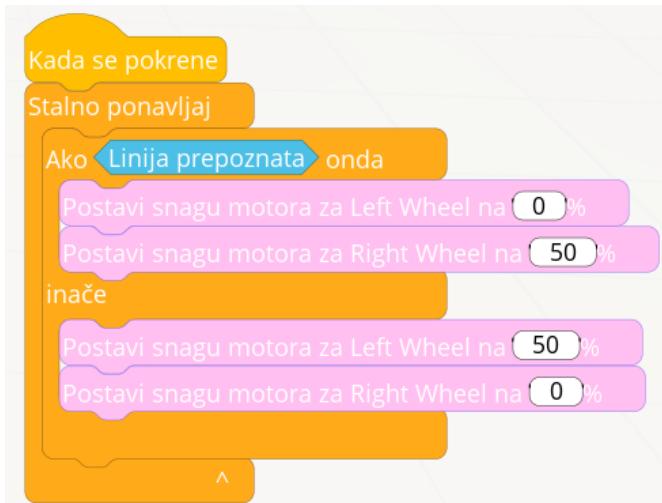


U ovom primjeru smo pokazali kako robot može uspješno pratiti crnu liniju koristeći samo jedan senzor. „Ključ“ ovakvog praćenja je u tome što ukoliko jedan motor miruje (snaga 0), a drugi se pokreće određenom snagom (npr. snaga 50), robot osim što blago skreće u željenom smjeru on se istovremeno i zakrivljeno pokreće unaprijed. Tzv. „krivuljastim praćenjem“ osim ravnih linija robot može uspješno pratiti i razne zakrivljene linije na stazama, pa čak i o ne sa pravim ili oštrim kutom.

## Primjer 2

U prvom primjeru robot je pratio crnu liniju po njenome desnome rubu. Praćenje linije se isto može postići ako želimo da robot prati liniju po njenome lijevome rubu, što ćemo pokazati u ovome primjeru. Izbor hoće li robot pratiti liniju po njenome desnom ili lijevom rubu najviše zavisi o vrsti zadatka i izgledu same staze sa linijama za praćenje.

Slika 4.0. Praćenje lijevog ruba linije naredbom AKO-ONDA-INAČE



Možemo uočiti da je jedina razlika u programu praćenja lijevog ili desnog ruba linije u različito postavljenim snagama lijevog i desnog motora.

### 6.3. Senzor za prepoznavanje objekta

Robot se često unutar različitih robotskih staza susreće sa raznim objektima ili preprekama.

Da bi se robot uspješno zaustavio ispred pojedine prepreke ili je dodatno zaobišao potrebno je programirati robota da uspješno prepozna kada se dovoljno približi određenom objektu ili prepreci.

Robot koristi „Senzor za prepoznavanje objekta“ (*Object detected sensor*) kako bi se uspješno zaustavio ispred pojedinog objekta ili prepreke.

Edison točnije za to koristi Infracrveni senzor (IR senzor) koji šalje i prima infracrvene zrake te na osnovu toga prepozna blisku udaljenost od nekog objekta ili pak drugog robota.

IR senzor ne može mjeriti točnu udaljenost od nekog objekta, kao što to može npr. ultrazvučni senzor (*ultrasonic sensor*), već može samo prepoznati blisku udaljenost ili sami kontakt sa određenim objektom ili drugim robotom.

**Slika 4.1. Položaj IR senzora na robotu**



**Senzor za prepoznavanje objekta  
(IR senzor)**

Naredba za prepoznavanje objekta je poprilično jednostavna kao i kod senzora za prepoznavanje linije te se također može pronaći u grupi naredbi „Edison“ te izgleda ovako;

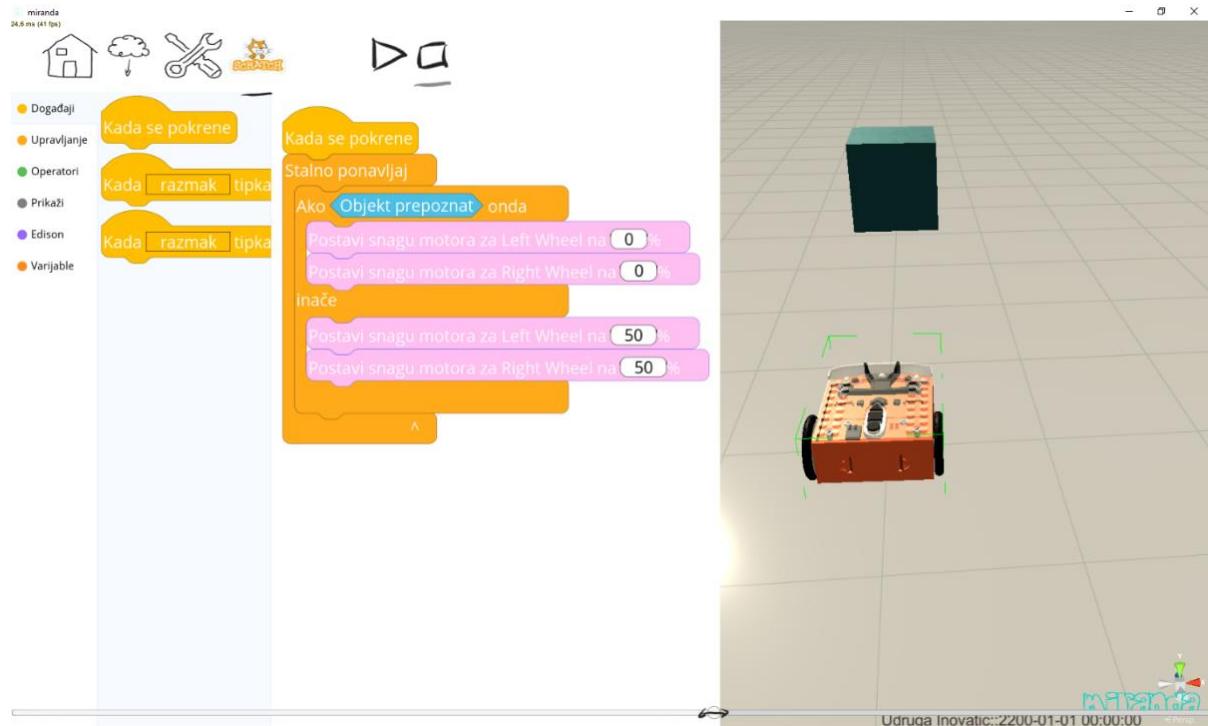
Objekt prepoznat

Naredbu za prepoznavanje objekta postavljamo unutar konteksta samog programa ovisno o tome što želimo postići senzorom za prepoznavanje objekta.

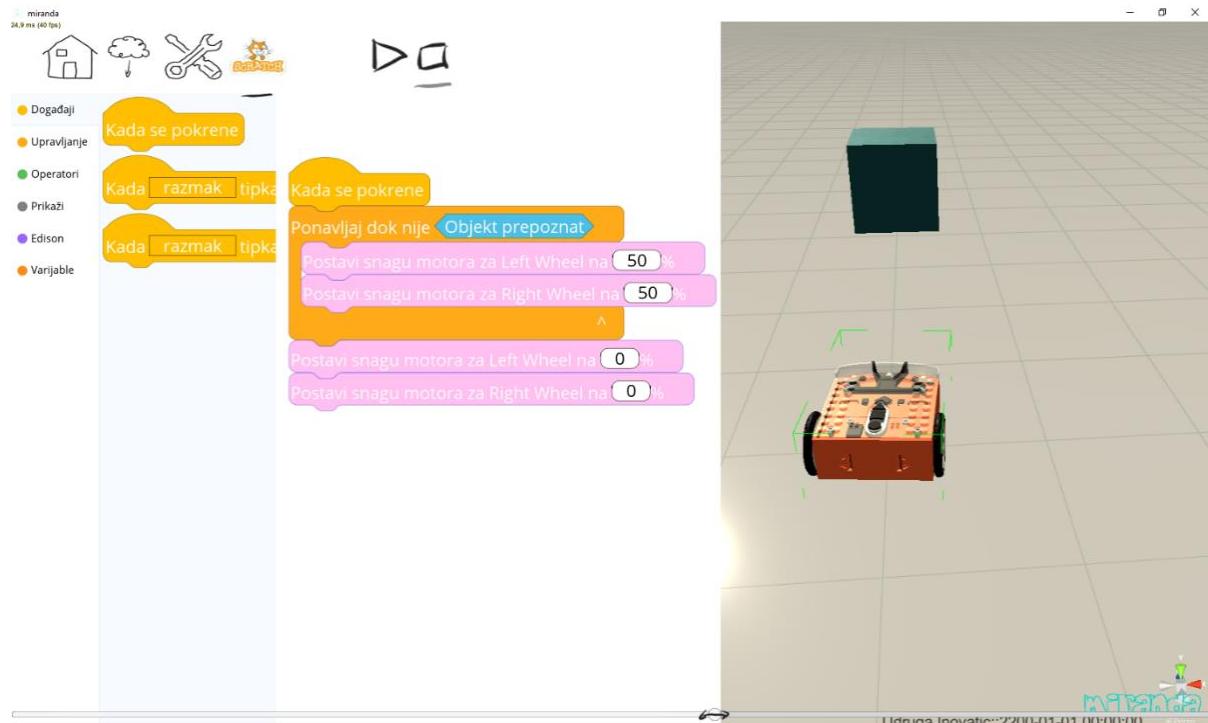
#### Primjer 1

Napravimo program da se robot giba ravno sve dokle ne dođe do nekog objekata te da se zatim pomoću senzora zaustavi. Ovaj program se može riješiti na dva načina. Jedan je pomoću naredbi AKO-ONDA-INAČE, a drugi način je pomoću naredbe PONAVLJAJ-DOK NIJE.

Slika 4.2. Prepoznavanje objekta naredbom AKO-ONDA-INAČE



Slika 4.3. Prepoznavanje objekta naredbom PONAVLJAJ-DOK NIJE



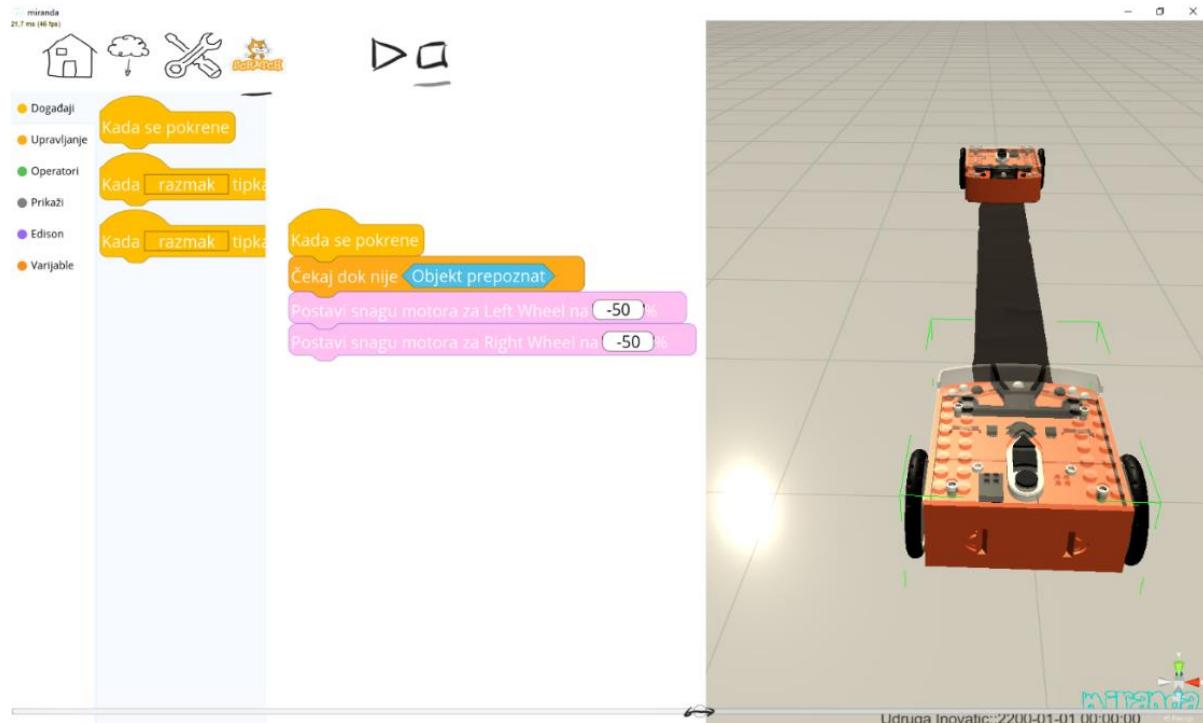
## Primjer 2

U prethodnom primjeru se robot kretao naprijed sve dokle senzorom ne bi prepoznao određeni objekt i potom se zaustavio. Međutim ponekad u zadacima je korisno da robot može prepoznati približavanje nekog drugog objekta ili robota kada se približava prema njemu.

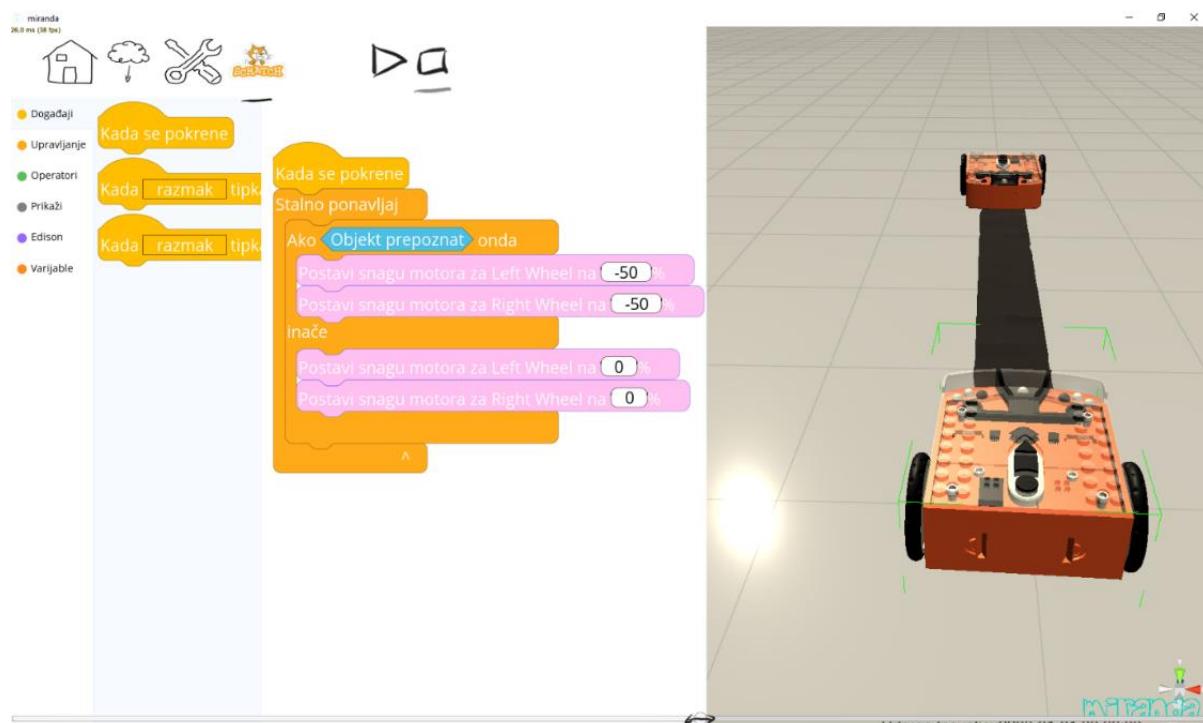
Npr. zadatak može biti da robot miruje sve dokle mu se na malu udaljenost ne približi neki drugi robot. Kako se ne bi sudario sa robotom koji mu se približava robot se kreće unazad dočim mu se drugi robot približi na malu udaljenost.

Ukoliko nam je dovoljno da se uvjet ispituje samo jednom u programu možemo koristiti naredbu ČEKAJ DOK NIJE (*wait until*) iz grupe naredbi UPRAVLJANJE.

Slika 4.4. Prepoznavanje drugog robota naredbom ČEKAJ DOK NIJE



Slika 4.5. Prepoznavanje drugog robota naredbom AKO-ONDA-INAČE

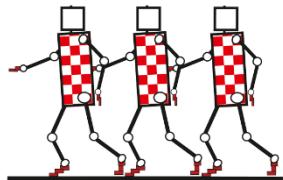


KRAJ 😊

Izradu priručnika omogućili:

Hrvatski robotički savez, Zagreb

<http://hrobos.hr/>



Udruga za robotiku „Inovatic“, Pmf-Split

<http://udruga-inovatic.hr/>



Robotski simulator:

<https://www.miranda.software/>

