

Domagoj Ciković
Mate Gambiraža
Lucija Medić
Ariana Milašinčić

0036487757
0036491608
0036491660
0036492925

SEMINARSKI RAD - SPVP



Fakultet elektrotehnike i računarstva, Sveučilišta u Zagrebu
Zavod za elektroničke sustave i obradbu informacija
Sveučilište u Zagrebu

DKU-IRIM .

zadaci za učenike



- Δ Namijenjeno edukatorima i učenicima
- Δ mBot, micro:bit, mBlock program
- Δ Uparivanje mBota i micro:bita
- Δ Korištenje senzora za boju i UV senzora

Sažetak

Projekt DKU-IRIM predstavlja digitalne edukacijske materijale koji se koriste kao podloga za provedbu radionica robotike, organizaciju natjecanja i za djecu koja se žele iskušati u složenijim zadacima. Ideja projekta je korištenjem edukativnih robota potaknuti razvoj digitalne i znanstvene pismenosti, tehnoloških i ostalih kompetencija u okviru STEM područja za mlade. Projekt je izveden u suradnji s IRIM-om, a materijali će biti objavljeni na portalu Izradi!.

Edukacijski materijali su namijenjeni profesorima obrazovno-odgojnih ustanova i lokalnih zajednica, kao i učenicima kojima bi materijali bili izrazito jednostavna, funkcionalna i kvalitetna platforma za učenje i rad.

Sadržaj

1. UVOD	4
2. EDUKACIJSKA SREDSTVA	5
2.1. mBot	5
Grafičko programiranje (mBlock)	6
Ultrazvučni senzor	7
Senzor za praćenje linije	7
Senzor za boju	8
2.2. micro:bit	9
3. ZADATCI	11
3.1. Simulacija automobila	11
Simulacija automobila 1	11
Simulacija automobila 2	14
3.2. Upravljanje mBotom pomoću micro:bita	17
3.3. Prolazak kroz labirint s mBotom koristeći 2-3 ultrazvučna senzora	32
1. zadatak: Maze Solver 2	32
Potrebna oprema	32
Programiranje	33
Rješenje	33
Objašnjenja programa	35

Ovaj seminarski rad je izrađen u okviru predmeta „Sustavi za praćenje i vođenje procesa“ na Zavodu za elektroničke sustave i obradbu informacija, Fakulteta elektrotehnike i računarstva, Sveučilišta u Zagrebu.

Sadržaj ovog rada može se slobodno koristiti, umnožavati i distribuirati djelomično ili u cijelosti, uz uvjet da je uvijek naveden izvor dokumenta i autor, te da se time ne ostvaruje materijalna korist, a rezultirajuće djelo daje na korištenje pod istim ili sličnim ovakvim uvjetima.

GLAVNI PROGRAM.....	35
PRATI LINIJU	35
ZAOKLIČI PREPREKU.....	37
2. zadatak: Maze Solver 1.....	37
Potrebna oprema	37
Programiranje	37
Rješenje	37
ZAKLJUČAK.....	39
4. LITERATURA.....	40

1. Uvod

STEM aktivnosti su alat za postizanje važnih kompetencija: ključnih vještina kao što su vještine učenja, rješavanja problema, suradnje, komunikacije, kao i osobina ličnosti kao što su znatiželja, inicijativa, upornost, prilagodljivost, društvena i kulturološka svjesnost. Stoga ne čudi da su STEM aktivnosti sve više zastupljenije u obrazovno-odgojnim ustanovama i lokalnim zajednicama.

IRIM kroz pokret Croatian Makers uvodi STEM aktivnosti u obrazovno-odgojne ustanove i lokalne zajednice te s dosegom od preko 100.000 djece najveći je takav izvankurikularni pokret u Hrvatskoj [1]. Na digitalnom portalu Izradi! dostupni su edukacijski materijali za profesore i učenike [2].

Povećanje zanimanja za STEM područje zahtijeva razvoj novih edukacijskih materijala. Stoga se u sklopu ovog projekta predstavljaju tri osmišljena zadatka: Simulacija automobila, Maze Solver i Upravljanje mBotom pomoću micro:bita. Osmišljeni zadatci su popraćeni uputama i rješenjima. Svaki od zadataka se sastoji od dva dijela. Prvi dio je jednostavnija verzija zadatka, dok je drugi dio nešto složenija izvedba.

2. Edukacijska sredstva

Prilikom izrade projekta koristila su sljedeća edukacijska sredstva: mBot i micro:bit. Za programiranje mBota se koristio program mBlock, dok se za programiranje micro:bit-a koristio program MakeCode.

2.1. mBot

mBot je cjelokupno rješenje koji daje praktično iskustvo u programiranju, elektronici i robotici. Riječ je o edukacijskom robot setu (Slika 1), kojemu Arduino predstavlja srce programa. Rad s mBot robot setom inspiriran je Scratch 2.0. programom koji se može povezati s mBot uređajem koristeći Bluetooth ili 2.4GHz wireless modul. Ovaj jednostavan za sastavljanje mBot edukacijski robot set pruža neograničene mogućnosti učenja [3].

Sadrži 38 sastavnih dijelova koji se mogu složiti u 10 minuta, a jedinstveno obojani RJ25 priključci za žice osiguravaju više vremena za programiranje i kreativnost.

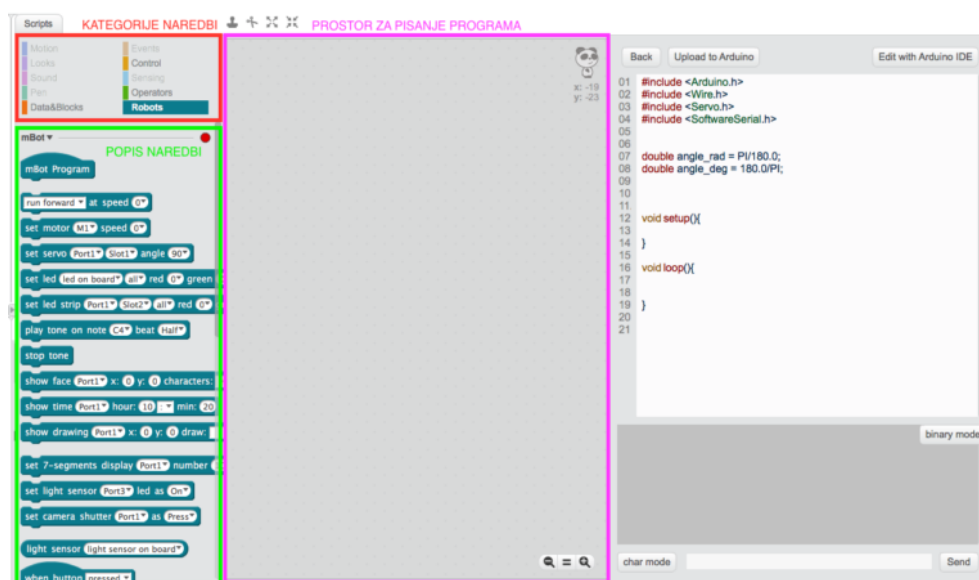


Slika 1: mBot edukacijski robot set

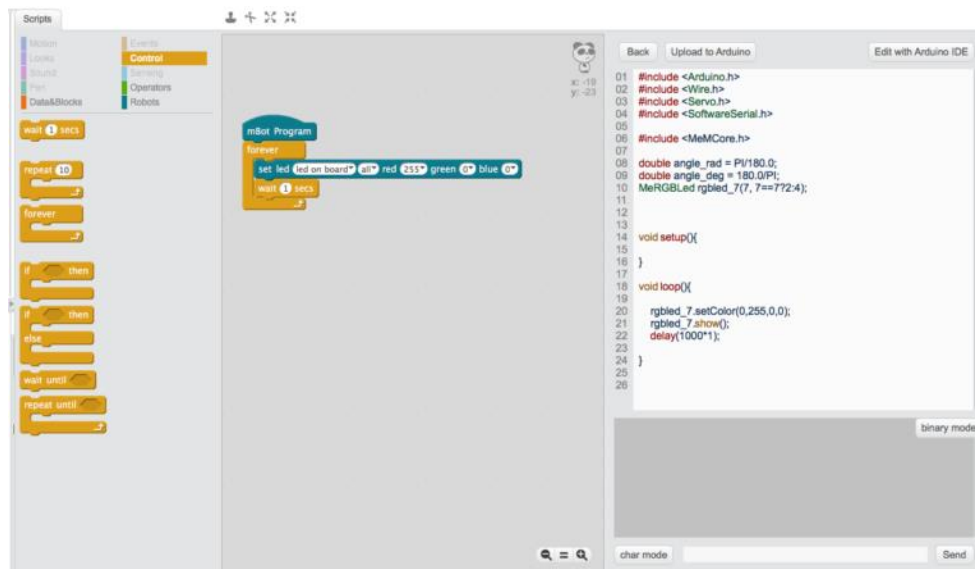
Grafičko programiranje (mBlock)

„Povuci i ispusti“ računarno-grafički softver mBlock razvijenog na temelju Scratch 2.0. omogućuje brzo učenje programiranja, kontrolu robota i realizaciju sve više funkcija koje mBot ima [3]. Širenjem svoga portfelja s Arduino nastavcima, moguće je prebacivanje s grafičkog na jezično kodiranje u Arduino modu. Naredbe se izvršavaju jedna po jedna od vrha programa prema dnu, pa je zbog toga važno da su naredbe spojene.

Kako bi prebacivanje programa na robota bilo moguće robot mora biti spojen na računalo putem USB kabela te sklopka na robotu mora biti uključena. Također, robot mora biti povezan s mBlock programom preko opcije Connect – Serial port. Povezivanje s mBlock programom potrebno je napraviti svaki put kada se robot priključi na računalo.



Slika 2: mBlock Desktop sučelje sa naznačenim sekcijama



Slika 3: Primjer mBlock programa

Ultrazvučni senzor

Jedan od senzora koji se koristio prilikom izrade zadatka je UV senzor. Ultrazvučni se senzor nalazi na prednjoj strani robota i kablom je spojen na jedan od portova jezgre robota. UV senzor radi tako da odašilje zvuk koji se zatim odbija od raznih predmeta i vraća robotu. Ovisno o tome koliko je vremena potrebno da se zvuk vrati, robot može odrediti udaljenost od samog predmeta koji se ispred njega nalazi. UV senzor se u zadacima koristi za zaobilazanje prepreka.



Slika 4: Ultrazvučni senzor

Senzor za praćenje linije

Senzor za praćenje linije je također jedan od senzora koji se koristi u raznim zadacima. Senzor se nalazi s donje strane robota zajedno s trećim kotačićem te je kablom spojen na jezgru robota. Senzor se sastoji od dva optička senzora te oba mogu razlikovati crnu i bijelu boju. Na senzoru se nalaze i dvije LED diode s gornje strane senzora. Svaka od njih pokazuje stanje optičkog senzora. Ako LED dioda svijetli plavom bojom, senzor je detektirao bijelu boju, odnosno LED dioda ne svijetli ako je ispod senzora crna podloga. Kada se oba optička senzora nalaze na crnoj podlozi tada se prilikom očitavanja senzora u programu dobije vrijednost 0. Kada se lijevi optički senzor nalazi na crnoj podlozi, a desni na bijeloj podlozi, uzima se vrijednost 1. U obratnom slučaju, vrijednost je 2. Zadnji slučaj, kada su oba optička senzora na bijeloj podlozi, odnosno izašli su s putanje linije, očitava se vrijednost 3.

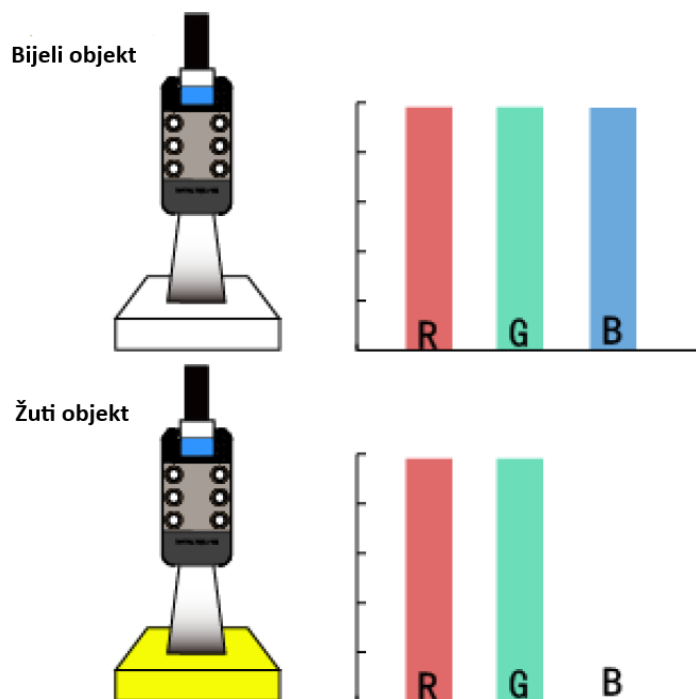


Slika 5: Senzor za praćenje linije

Senzor za boju

Osim prethodno navedenih senzora, koristio se i senzor za boju (Slika 7). U nastavku je dano pojašnjenje rada samog senzora.

Svjetlost vidljiva ljudskom oku sastoji se od tri osnovne boje – crvene, zelene i plave (R, G, B). Kada svjetlost obasja objekte, razlika u svojstvima površine objekata uzrokovat će različite apsorpcijske i refleksijske efekte [4]. Senzor za boju određuje boju površine objekta otkrivanjem intenziteta refleksije triju primarnih boja (Slika 6).



Slika 6: Primjer rada senzora za boju

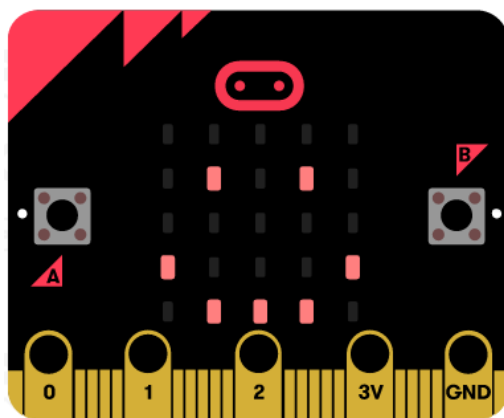
Senzor za boju prepoznaje 6 boja – crnu, bijelu, crvenu, zelenu, plavu i žutu. Nalazi se ispod robota, a preporučena visina na kojoj bi se senzor trebao nalaziti u odnosu na površinu je između 1.5 i 3 cm. Za komunikaciju koristi I2C protokol te se na upravljačku pločicu mBota (mCore) povezuje RJ25 kabelom na bilo koji port. Vrijeme potrebno da senzor prepozna promjenu boje iznosi 160 ms.



Slika 7: Senzor za boju

2.2. *micro:bit*

micro:bit je tzv. „mini-računalo“. Ono sadrži 3 tipke pomoću koji možemo komunicirati sa samim micro:bit-om. Dvije tipke nalaze se s prednje strane micro:bit-a i one služe za izvršavanje željenih naredbi. Treća tipka nalazi se sa stražnje strane i služi za ponovno pokretanje programa na micro:bitu-u. Ovo „mini-računalo“ također ima zaslon na kojem se mogu ispisivati brojevi, slova ili možemo sami odrediti što će biti prikazano. Zaslon se sastoji od 5x5 polja LED dioda. micro:bit sadrži i ulazne pinove na koje možemo dovoditi razne signale i mjeriti iste. Uređaj se može napajati pomoću baterija ili USB kablom putem računala. Osim navedenih značajki pravog računala, micro:bit sadrži i nekoliko senzora. To u akcelerometar, senzor za mjerenje intenziteta svjetlosti te senzor temperature. micro:bit podržava bluetooth i radio komunikaciju te može „razgovarati“ s drugim uređajima.



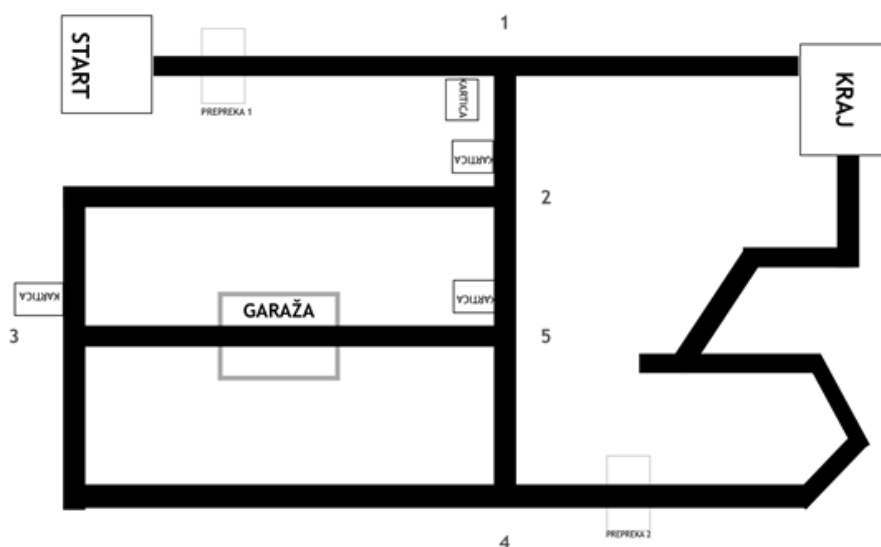
Slika 8: micro:bit

3. Zadatci

3.1. Simulacija automobila

Potrebna oprema: mBot (dva DC motora, 4x AA baterije, kućište za baterije, ultrazvučni senzor, senzor za praćenje linije), senzor za boju, staza (Slika 9) i prepreka.

Za programiranje mBota koristi se programski alat mBlock (verzija 3.4.12).



Slika 9: Staza koja se koristi u zadatcima Simulacija automobila 1 i 2

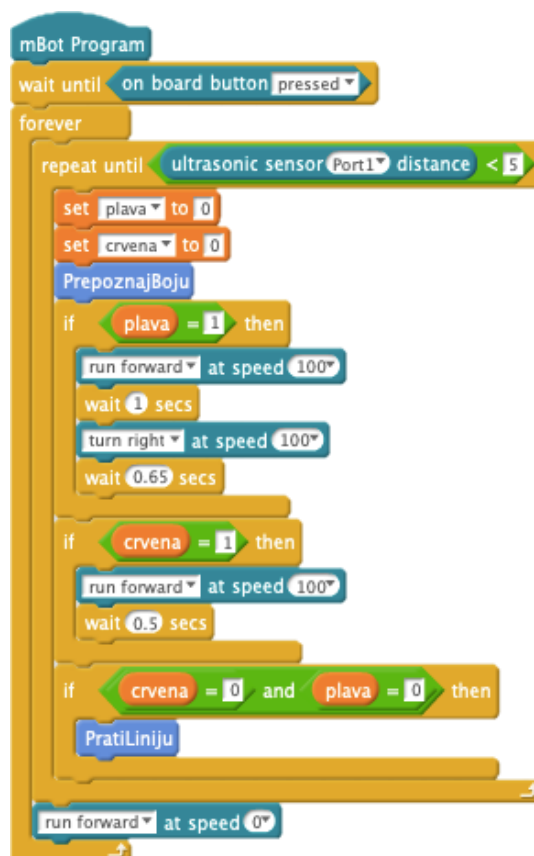
Simulacija automobila 1

Opis zadatka: U ovom zadatku pomoću mBot-a potrebno je simulirati vožnju i parkiranje automobila. Pri nailasku na raskrižje skreće ovisno o boji koja je postavljena neposredno prije raskrižja. Detekcijom crvene boje, mBot na raskrižju nastavlja voziti ravno. Ako je detektirana plava boja, mBot skreće desno. Ako se na raskrižju ne nalazi kartica za boju, mBot nastavlja pratiti liniju s lijeve strane te ako se nalazi na raskrižju automatski skreće lijevo ili nastavlja voziti ravno.

mBot se postavlja na poziciju Start te se pokreće pritiskom na dugme. Vozi se prateći liniju s lijeve strane. Na raskrižju 1 i 2 mBot treba

skrenuti desno (plava boja) te nastavlja pratiti liniju do sljedećeg raskrižja, raskrižja 3, gdje treba nastaviti voziti pravo. Prilikom dolaska na raskrižje 4 mBot automatski skreće lijevo te istu radnju ponavlja na raskrižju 5. Nakon obavljenih radnji mBot dolazi do garaže gdje se parkira. Parkiranje je izvedeno tako da se mBot zaustavlja kada detektira prepreku na udaljenosti od 5 cm. Parkiranjem mBota vožnja je završena.

Glavni program: mBot je potrebno postaviti na Start poziciju. Prilikom paljenja mBot čeka pritisak na tipku da bi krenuo s izvođenjem programa. Variable *plava* i *crvena* se postavljaju u 0, te se poziva funkcija *PrepoznajBoju*. Ako funkcija detektira boju potrebno je ovisno o detektiranoj boji odraditi skretanje na raskrižju. Boje koje se koriste su plava i crvena. Ako je prepoznata plava boja, mBot prvo ide ravno 2 sekunde brzinom 100, a potom skreće desno 0.65 sekundi brzinom 100. Ako je riječ o crvenoj boji mBot se kreće ravno 0.5 sekundi, a ako nije prepoznata nijedna od navedenih boja, mBot prati liniju. Vožnja se odvija sve dok ultrazvučni senzor na *portu 1* ne detektira prepreku na udaljenosti (5 cm) koja predstavlja garažu. U tom slučaju mBot se zaustavlja, čime je završen prvi zadatak (slika 10).



Slika 10: Glavni program

Praćenje linije: Senzor za praćenje linije nalazi se na *portu 3*. Ako senzor za praćenje linije ima vrijednost 2, vozi ravno. Ako su oba senzora na svijetloj podlozi (vrijednost 3), robot skretanjem u desno traži tamnu podlogu kako bi došao u situaciju da vrijednost senzora bude 2 i da robot nastavi s vožnjom naprijed. U situacijama kad je robot s lijevim senzorom na tamnoj, a s desnim na svijetloj podlozi (vrijednost 1) ili s oba senzora na tamnoj podlozi (vrijednost 0), robot skretanjem ulijevo dolazi do položaja kad će senzor ponovno očitati vrijednost 2 i kretati se ravno (slika 11).



Slika 11: Funkcija za praćenje linije

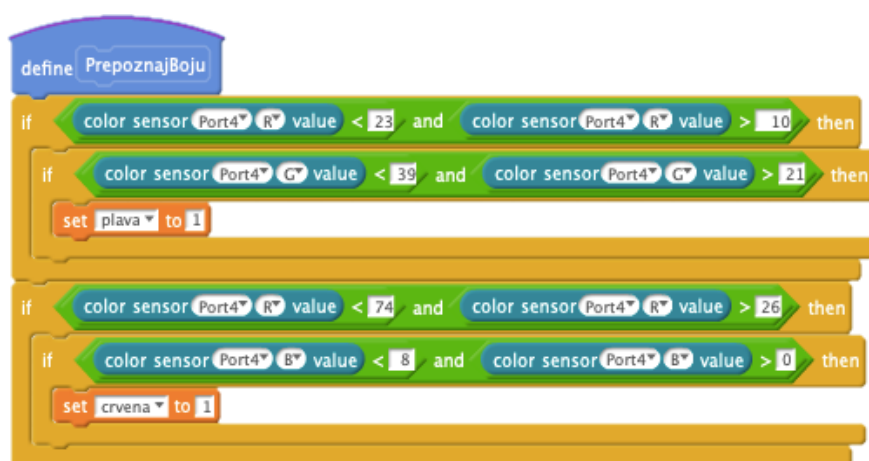
Prepoznavanje boje: RGB vrijednosti boja su određene očitavanjem senzora za boju u različitim osvjetljenjima, te su dobivene granične vrijednosti koje su prikazane u tablici.

Tablica 1: RGB vrijednosti crvene i plave boje

BOJA	[Rmin, Rmax]	[Gmin, Gmax]	[Bmin, Bmax]
PLAVA	[10, 23]	[21, 39]	[25, 38]

CRVENA	[26, 74]	[1, 14]	[0, 8]
--------	-----------	-----------	---------

Prepoznavanje boja je napravljeno preko očitavanja RGB vrijednosti sa senzora za boju (*port 4*). Plava boja se prepoznaje tako da je R vrijednost iz intervala [10, 23], a G vrijednost iz intervala [21, 39]. U slučaju crvene boje R poprima vrijednost iz intervala [26, 74] dok je B vrijednost iz intervala [0, 8] (Slika 12).



Slika 12: Funkcija za prepoznavanje boje

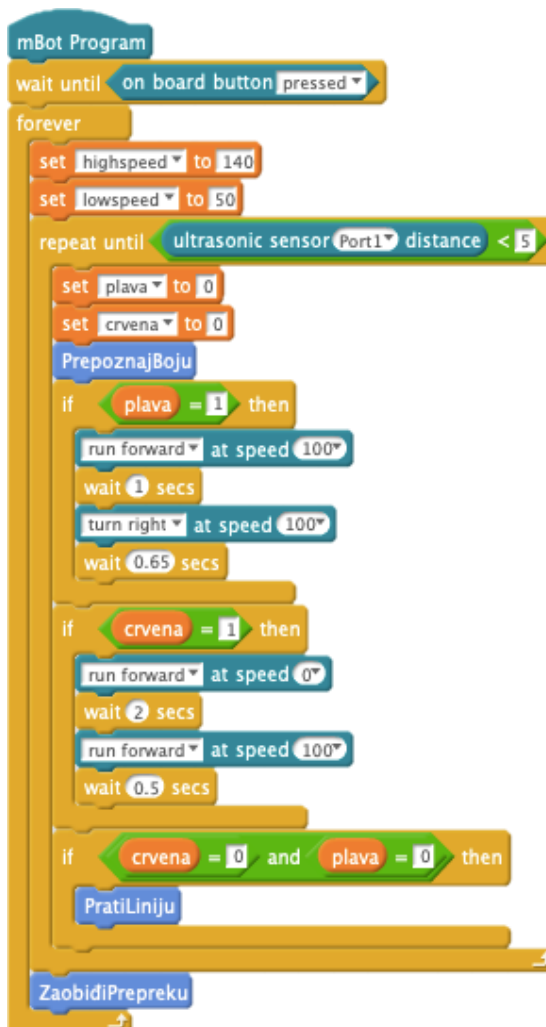
Simulacija automobila 2

Opis zadatka: Ovaj zadatak predstavlja nadogradnju prvog zadatka (Simulacija automobila 1). Potrebno je simulirati vožnju i parkiranje automobila kao i zaobilazanje prepreke. Pri nailasku na raskrižje skreće ovisno o boji koja je postavljena neposredno prije raskrižja. Detekcijom crvene boje, mBot na raskrižju nastavlja voziti ravno. Ako je detektirana plava boja, mBot skreće desno. Ako se na raskrižju ne nalazi kartica za boju, mBot nastavlja pratiti liniju s lijeve strane te ako se nalazi na raskrižju automatski skreće lijevo ili nastavlja voziti pravo.

mBot se postavlja na poziciju Start te se pokreće pritiskom na dugme. Vozi se prateći liniju s lijeve strane sve dok ne naiđe na prepreku. Pri nailasku na prepreku 1 mBot se treba zaustaviti 5 cm prije na 2 sekunde te zaobići prepreku s desne strane. Na raskrižju 1 mBot treba skrenuti desno (plava boja) te nastavlja pratiti liniju do sljedećeg

raskrižja, raskrižja 3, gdje se treba zaustaviti na 2 sekunde i nastaviti voziti pravo (crvena boja). Prilikom dolaska na raskrižje 5 mBot ponavlja istu radnju kao na raskrižju 3. Na raskrižju 4 mBot automatski skreće lijevo. Nakon prolaska kroz raskrižje 4 mBot treba zaobići prepreku 2 na isti način kao što je zaobišao prepreku 1. Nakon zaobilaženja prepreke 2 mBot nastavlja s vožnjom sve dok ne dođe do pozicije Kraj, koja predstavlja kraj vožnje.

Glavni program: mBot je potrebno postaviti na Start poziciju. Prilikom paljenja mBot čeka pritisak na tipku i započinje s praćenjem linije. Na samom početku postavljaju se varijable highspeed (140) i lowspeed (50) koje se koriste u funkciji ZaobiđiPrepreku. Potom se varijable plava i crvena se postavljaju u 0, te se poziva funkcija PrepoznajBoju. Ako funkcija detektira boju potrebno je ovisno o detektiranoj boji odraditi skretanje na raskrižju. Boje koje se koriste su plava i crvena. Ako je prepoznata plava boja, mBot prvo ide ravno 1 sekundu brzinom 100, a potom skreće desno 0.65 sekundi brzinom 100. Ako je riječ o crvenoj boji mBot se zaustavlja na 2 sekunde, a zatim ide ravno 0.5 sekundi. Ako nije prepoznata nijedna od navedenih boja, mBot prati liniju. Vožnja se odvija sve dok ultrazvučni senzor na portu 1 ne detektira prepreku na određenoj udaljenosti (5 cm). Kada se detektira prepreka započinje se proces zaobilaženja pozivom funkcije ZaobiđiPrepreku. Nakon zaobilaženja, mBot nastavlja s vožnjom. Zadatak je završen dolaskom mBota na poziciju Kraj (Slika 13).



Slika 13: Glavni program

Praćenje linije: izvedeno isto kao i u poglavlju *Simulacija automobila 1* (slika 11).

Prepoznavanje boje: izvedeno isto kao i u poglavlju *Simulacija automobila 1* (slika 12).

Zaobilaženje prepreke: Jednom kada se detektira prepreka robot se zaustavlja na 2 sekunde (automatsko kočenje). Zatim započinje sa zaobilaskom prepreke tako da se mBot okrene desno brzinom *highspeed* (140) 0.5 sekundi. Potom se robotu postavlja brzina desnog motora (M2) na *highspeed* (140) dok mu se brzina lijevog motora (M1) postavlja na *lowspeed* (50). Brzine motora ostaju takve sve dok je vrijednost senzora za praćenje linije (*port3*) manja od 3, odnosno sve dok mBot ne detektira crnu liniju (Slika 14).



Slika 14: Funkcija za zaobilaženje prepreke

3.2. Upravljanje mBotom pomoću micro:bita

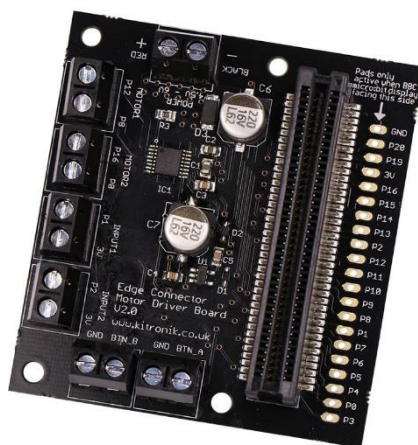
Potrebno:

- mBot
- DC motor 2 komada
- Motor driver

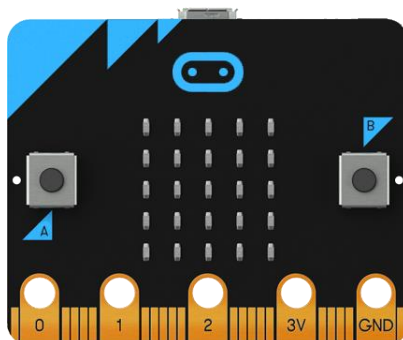
- micro:bit 2 komada
- 4xAA baterije
- Kućište za baterije
- Žice



Slika 15: mBot



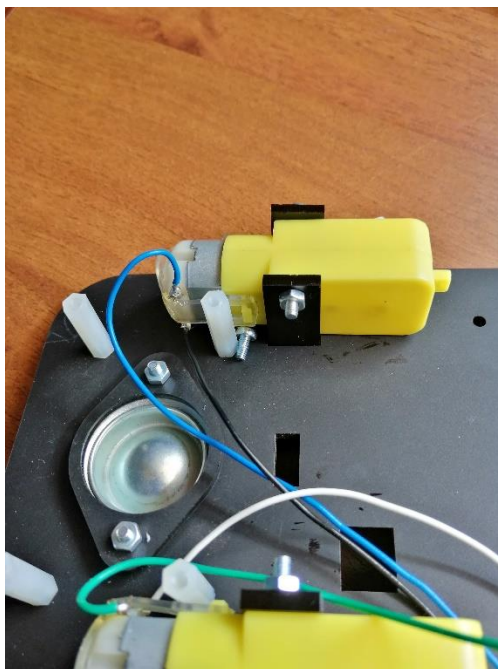
Slika 16: Motor driver



Slika 17: micro:bit

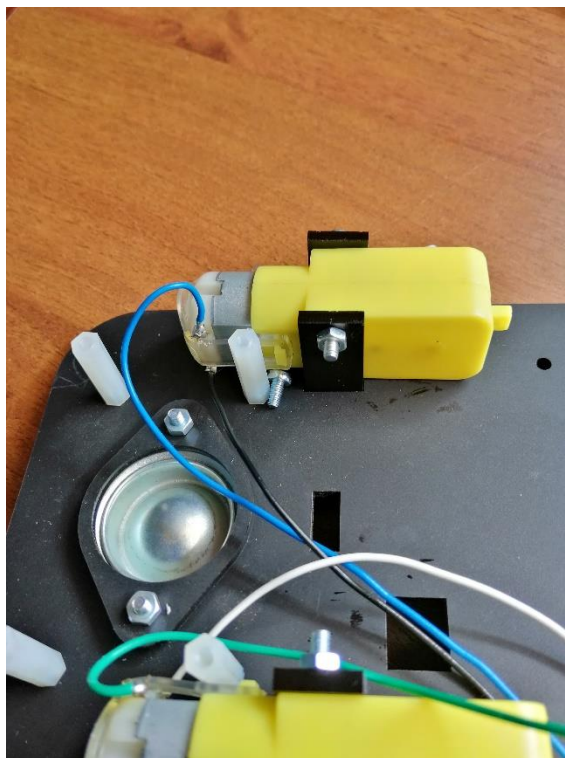
U ovom zadatku se neće koristiti Arduino pločica mBota te ju je potrebno maknuti s mBota. Na to mjesto ugradit će se motor driver za micro:bit i kućište za baterije.

DC motori se pomoću žica spajaju na motor driver na sljedeći način. Desni motor se pomoću žica, u ovom slučaju crne i plave žice, spaja na motor driver na konektor pod nazivom MOTOR2. Crna žica, koja se nalazi s donje strane DC motora, spaja se na pin P16, dok se plava žica, spojena na gornju stranu DC motora, spaja na pin P0.



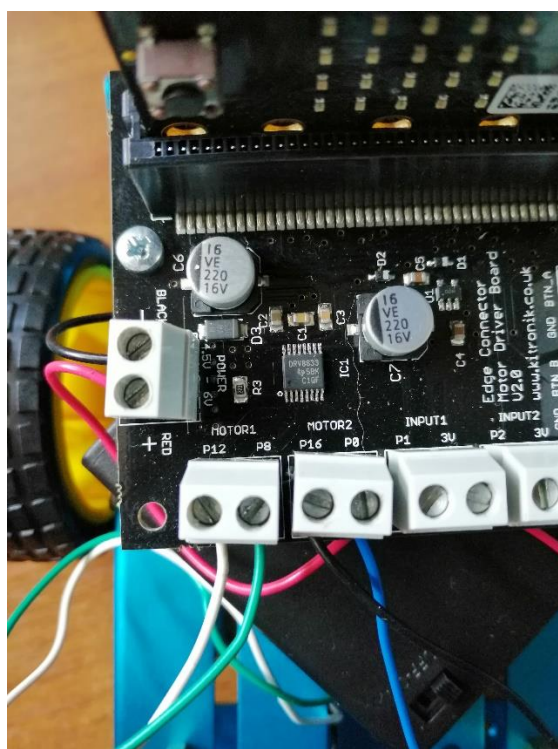
Slika 18: Desni DC motor

Na jednak način spaja se lijevi DC motor. Zelena žica se nalazi s gornje strane motora i nju spajamo na konektor pod nazivom MOTOR1 na pin P8. Bijela žica se nalazi s donje strane i nju spajamo na pin P12.



Slika 19: Lijevi DC motor

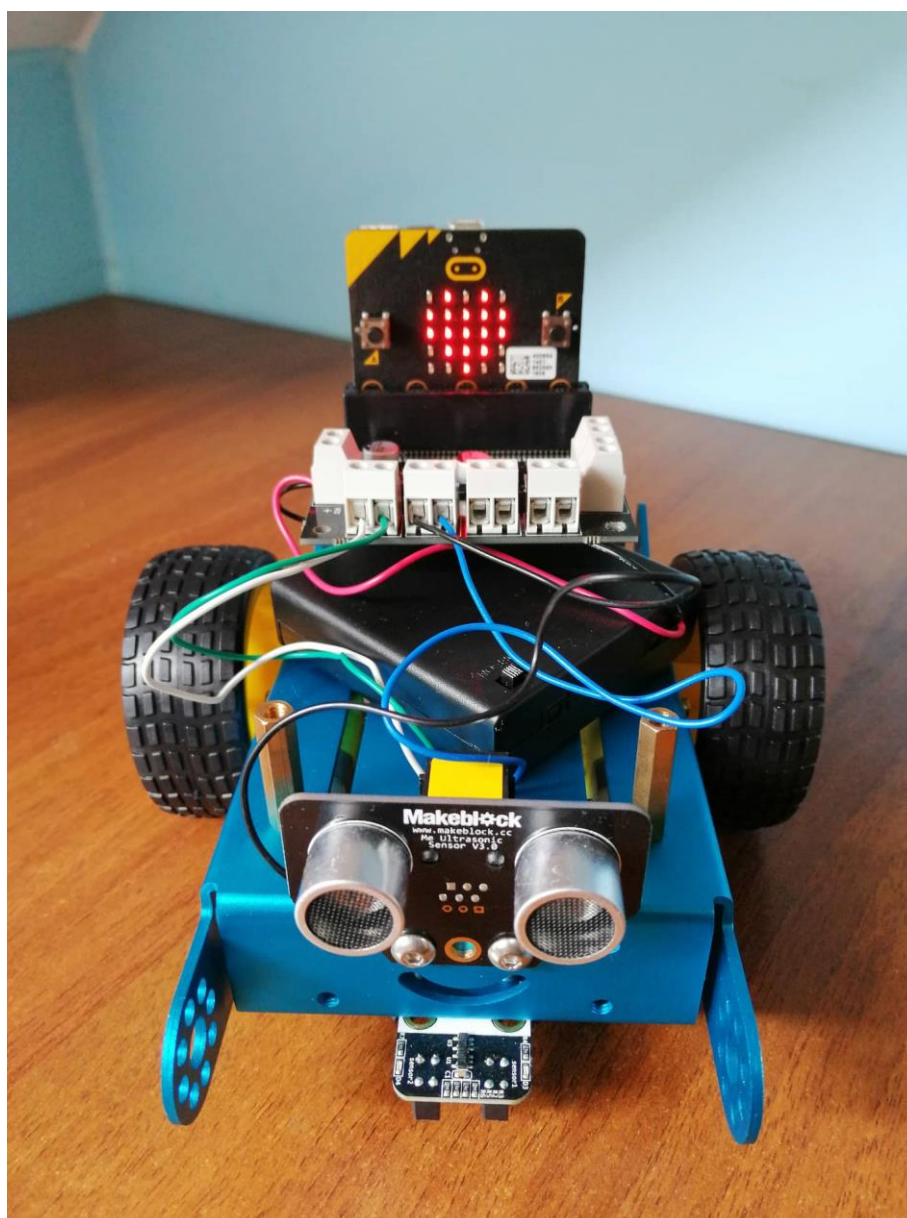
Za napajanje motor drivera i micro:bita koriste se 4 AA baterije koje su smještene u kućištu. Kućište se spaja pomoću žica na konektor pod nazivom POWER. Crna žica se spaja na pin – BLACK, a crvena na pin + RED.



Slika 20: Raspored pinova na motor driver

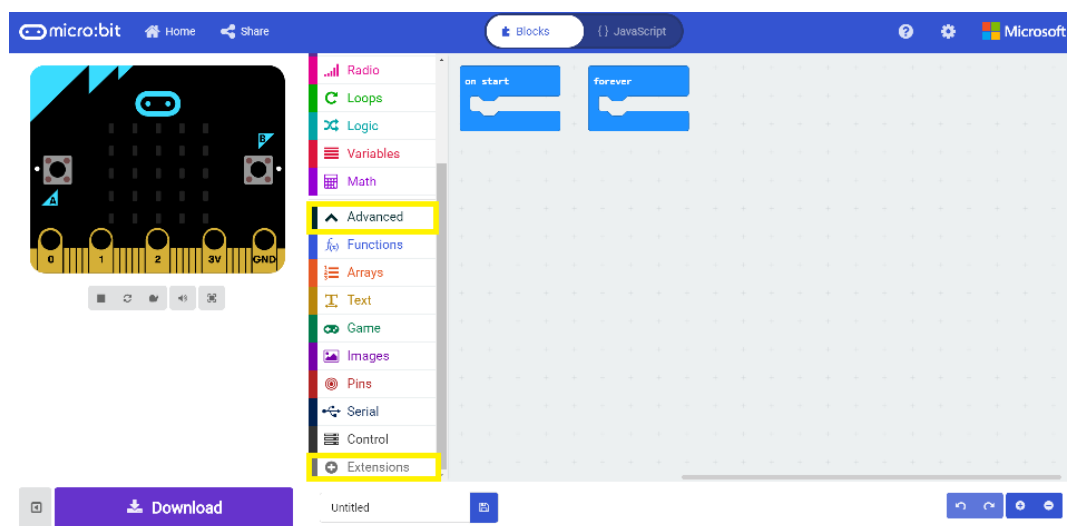
Prilikom spajanja žica na konektore potrebno je prvo otpustiti priključke pomoću odvijača, zatim gurnuti žica u odgovarajuće otvore i dobro zategnuti. Obratiti pozornost da prilikom spajanja žica baterije nisu u kućištu. Preporučuje se prvo postaviti DC motore i kućište na njihove pozicije te zatim spojiti pomoću žica.

Kada micro:bit stavljamo na motor driver potrebno ga je licem okrenuti prema konektorima za ispravan rad.



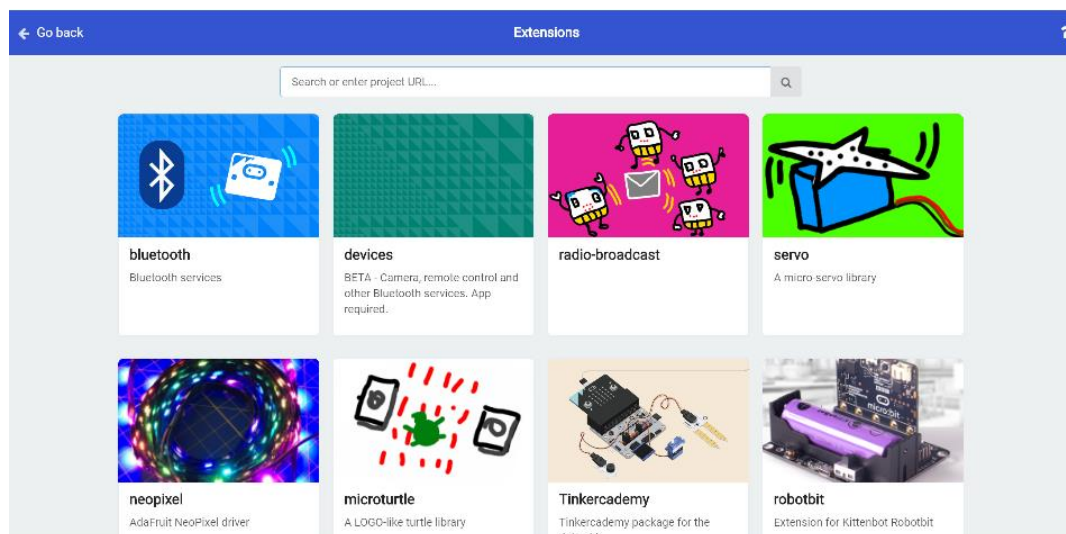
Slika 21: mBot upravljan pomoću micro:bita

Programiranje : Za programiranje micro:bita koristi se Microsoft MakeCode. Prilikom programiranja potrebno je uključiti ekstenziju za upravljanje motor driverom. To ćemo napraviti tako da u glavnom izborniku naredbi kliknemo na Advanced i zatim pronađemo opciju Extensions (na slici 8 označeno žutom bojom).

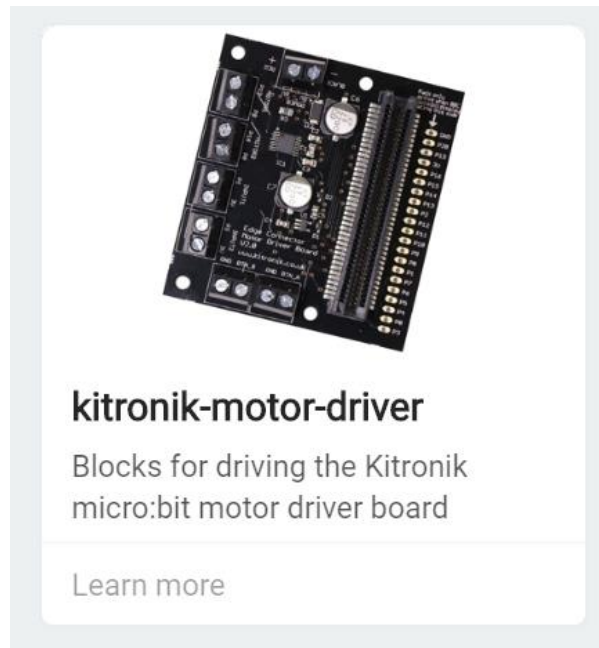


Slika 22: Opcija Extensions u izborniku za naredbe

Klikom na Extension otvara se nova stranica (slika 9) u gdje u polje za pretraživanje moramo upisati „kiktronik-motor-driver“ i odabrati ekstenziju kao što je prikazano na slici 10. U glavni program se vraćamo klikom na „Go back“ u gornjem lijevom kutu.



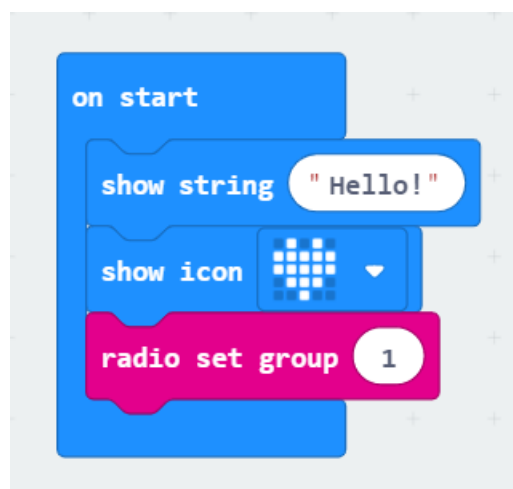
Slika 23: Stranica za biranje ekstenzija



Slika 24: Ekstenzija za motor driver

micro:bit kao upravljač:

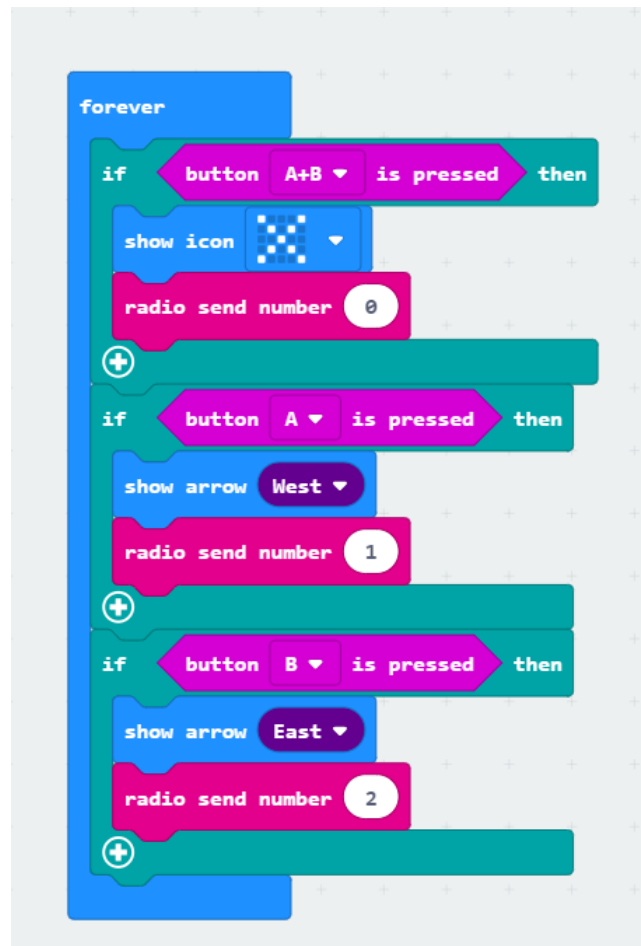
1. Kada uključimo micro:bit želimo da on uspostavi komunikaciju s drugim micro:bitom. To ćemo učiniti tako da iz naredbi za radio komunikaciju iskoristimo funkciju „radio set group“ i postavimo neki broj, u ovom slučaju to je broj 1. Tako ćemo odrediti s kojom grupom može naš micro:bit komunicirati. Broj koji postavimo mora biti jednak na oba micro:bita inače komunikacija neće biti ostvarena.



Slika 25: Dio programa za ostvarivanje radio komunikacije

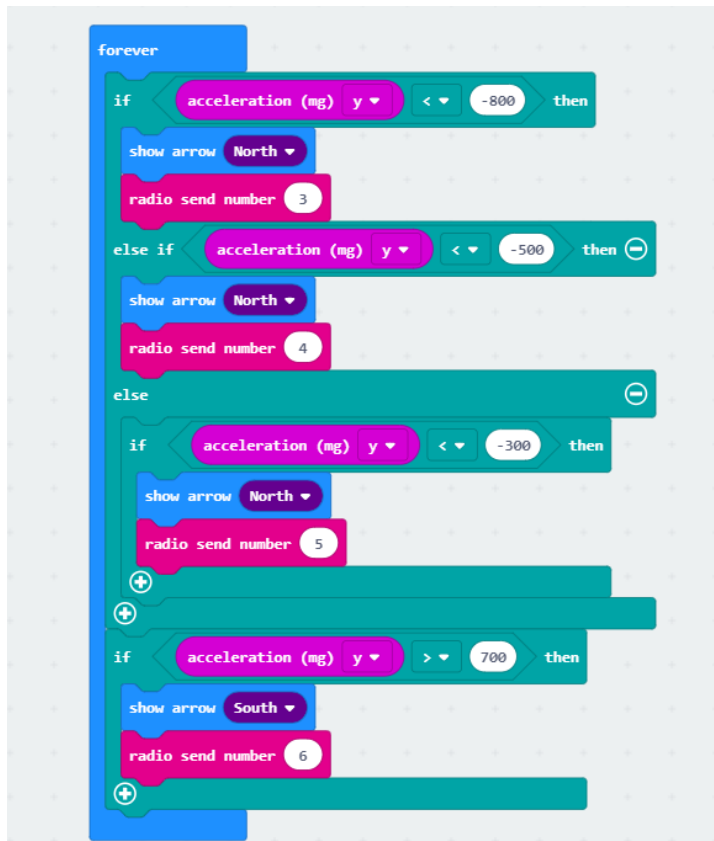
2. Prvi micro.bit koristit će se kao upravljač. Koriste se dvije forever petlje. U prvoj forever petlji želimo upravljati mBotom da on ide lijevo, desno ili da se on zaustavi. To možemo ostvariti tako da provjeravamo koja je tipka pritisnuta. Ako pritisnemo tipku A neka mBot ide lijevo, ako je pritisnuta tipka B neka mBot ide desno, a ako su istovremeno pritisnute tipke A i B neka se naš mBot zaustavi. Kako bi micro:bit koji upravlja motorima znao što smo stisnuli na upravljaču moramo putem radio veze poslati neki znak. To ćemo ostvariti tako da iskoristimo funkciju „radio send number“. Upravljač tada šalje neki proizvoljni broj koji će micro:bit koji upravlja motorima prepoznati i dalje izvršavati određene funkcije. If petlju koja provjerava jesu li istovremeno pritisnute tipke A i B potrebno je postaviti na početak, jer u suprotnom će robot prvo skrenuti lijevo pa desno i tek će se onda zaustaviti. Tako ćemo definirati da micro:bit šalje brojeve :

- 0 za zaustavljanje
- 1 za lijevo
- 2 za desno
- 3 za naprijed brzinom 100
- 4 za naprijed brzinom 50
- 5 za naprijed brzinom 30
- 6 za nazad

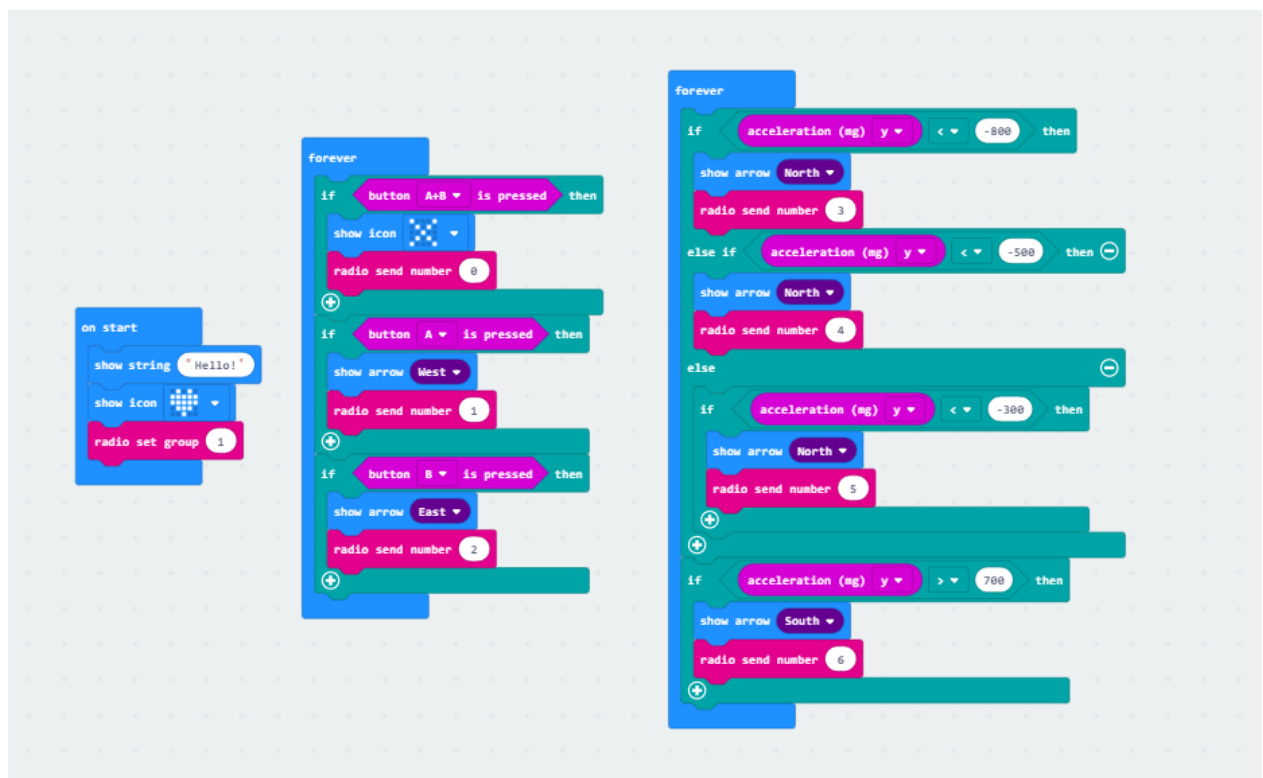


Slika 26: Program za upravljanje lijevo, desno, stani

3. U drugoj forever petlji koristi će se funkcija koja mjeri nagib pomoću g sile kako bi upravljali robotom da se kreće naprijed ili nazad. Kada je micro:bit nagnemo prema dolje želimo da robot krene naprijed, a kada ga nagnemo prema gore on će ići nazad. Moramo odrediti granicu kod koje želimo da robot krene te ispitati uvjet kada je ta granica pređena. To se ostvaruje na način prikazan u sljedećem primjeru. Brzinu koju želimo postići ostvarit ćemo odabirom različitih nagiba.



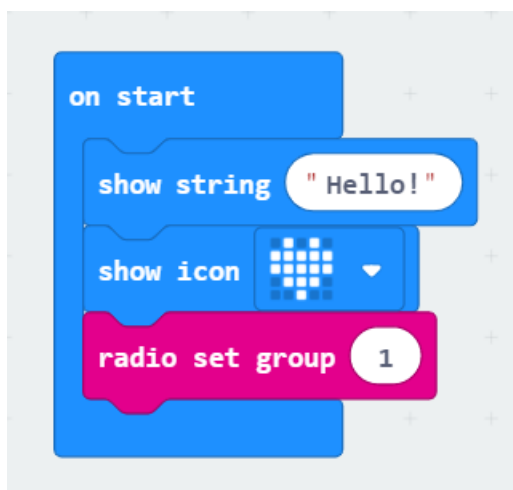
Slika 27: Program za upravljanje brzinom motora i kretanje naprijed/natrag



Slika 28: Cijeli program za micro:bit kao upravljač

micro:bit kao prijamnik:

1. Kao i kod prvog micro:bita moramo uspostaviti komunikaciju kao što je već navedeno.



Slika 29: Program za ostvarivanje komunikacije

2. Kako ovaj micro:bit koristimo kao svojevrsni prijamnik podataka, u izborniku za radio komunikaciju koristit ćemo funkciju „on radio received“ koja radi tako da podatak koji je poslan s drugog micro:bita spremi u neku varijablu, koja je u ovom slučaju nazvana BROJ. U varijablu broj spremaju se vrijednosti koje smo poslali s micro:bit upravljača na način kako smo definirali :
 - 0 za zaustavljanje
 - 1 za lijevo
 - 2 za desno
 - 3 za naprijed brzinom 100
 - 4 za naprijed brzinom 50
 - 5 za naprijed brzinom 30
 - 6 za nazad

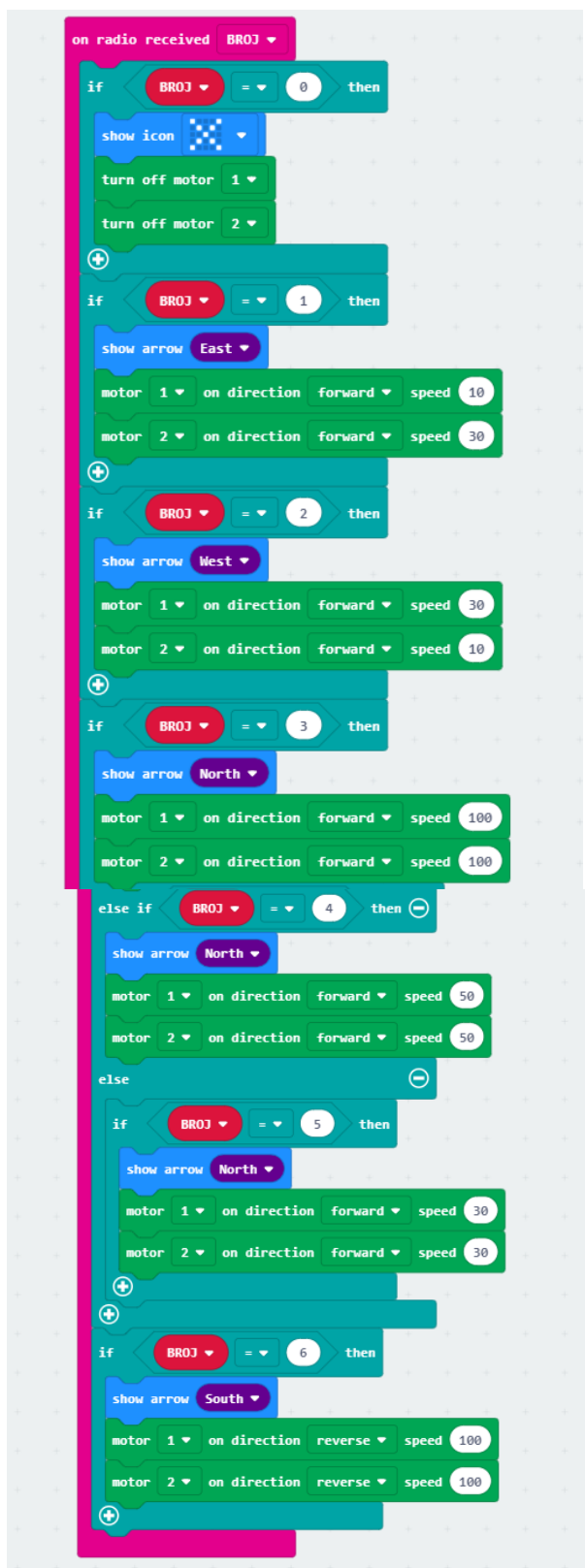
Zatim je potrebno kreirati novu varijablu BROJ putem izbornika Variables, koju zatim uspoređujemo.

Ako je varijabla BROJ jednaka 0 iz ekstenzije za motor driver koristimo funkciju „turn off motor“ i potrebno je u padajućem izborniku izabrati 1 i 2 da se motori zaustave.

Dalje, ako je varijabla BROJ jednaka 1 tada je potrebno podesiti brzine motora da bi mBot skrenuo ulijevo. Da bi robot skrenuo lijevo potrebno je podesiti brzine motora tako da je brzina desnog motora veća od brzine lijevog. Kako je nama desni motor, motor 2, a lijevi motor, motor 1 potrebno je namjestiti da je brzina motora 2 veća od brzine motora 1. Za preciznije okretanje robota izabrati manje brzine.

Jednako vrijedi i za skretanje za desno, no u ovom slučaju motor 1 mora imati veću brzinu.

Za kretanje naprijed ili unatrag uspoređujemo varijablu BROJ s 3 ako želimo da motor ide naprijed, odnosno s 6 ako želimo da se kreće unatrag. Razlika između ove dvije petlje je samo da u izborniku funkcije za upravljanje motorima odaberemo forward za naprijed ili reverse za unatrag. Jednako vrijedi i za određivanje brzine motora. U određeno polje upišemo samo broj od 0 do 100 za brzinu koju želimo postići.



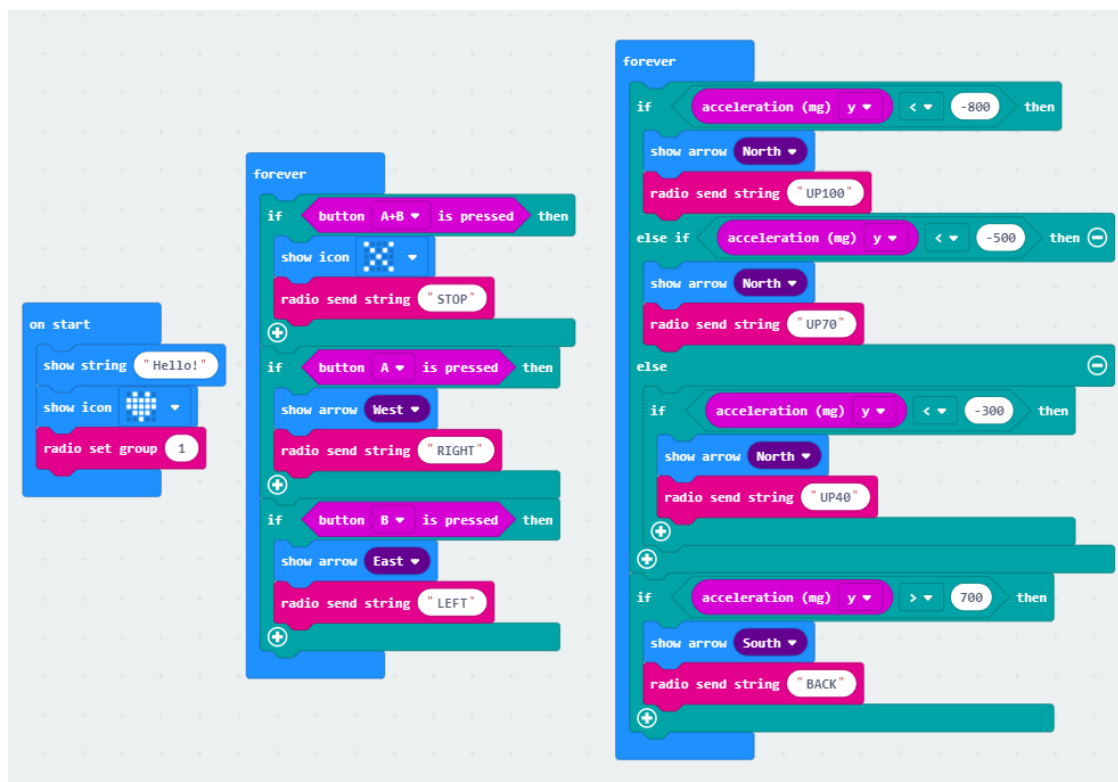
Slika 30: Upravljanje motorima preko primljenih brojeva

Programiranje pomoću stringa:

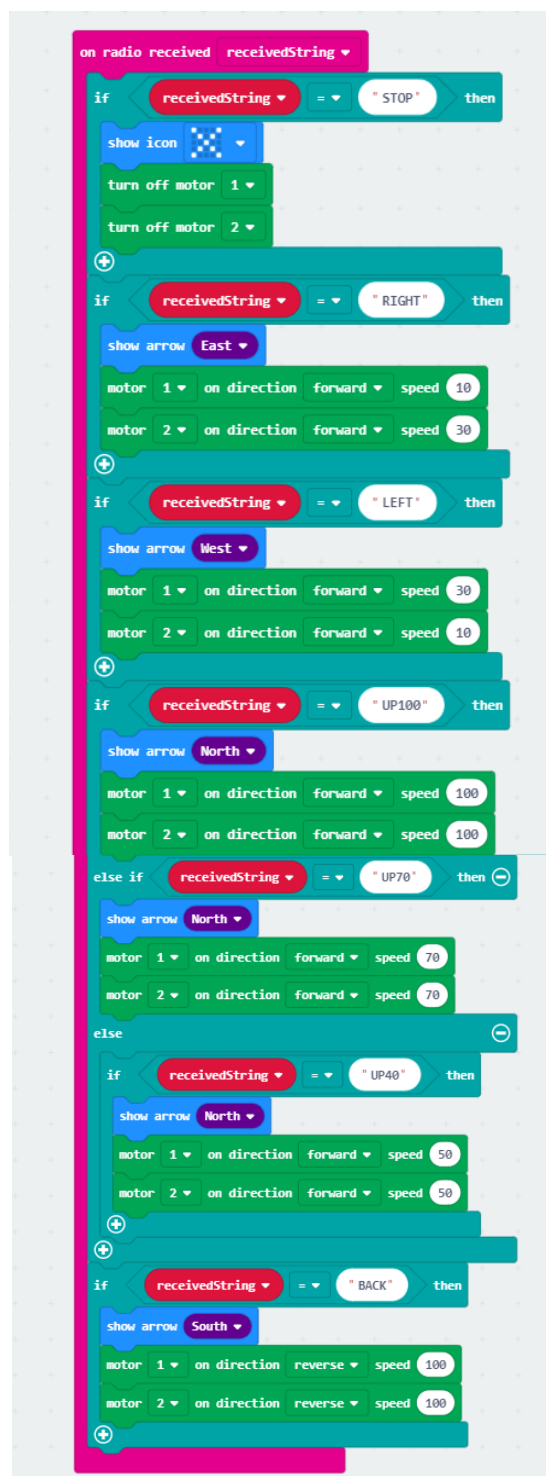
Postoji alternativan način kojim možemo programirati naše micro:bite. U ovom načinu koristit ćemo znakove koje šaljemo umjesto brojeva. Tako ćemo definirati niz znakova koje ćemo slati :

- 'STOP' za zaustavljanje
- 'LEFT' za lijevo
- 'RIGHT' za desno
- 'UP100' za naprijed brzinom 100
- 'UP70' za naprijed brzinom 70
- 'UP40' za naprijed brzinom 40
- 'BACK' za nazad

Iz izbornika se koriste 'Text' naredbe, tj. naredba u koju možemo upisivati tekst. Za razliku od prethodnog koda moramo koristiti naredbu 'on radio received' i u izborniku se odabire receivedString varijablu u koju se sprema poslani znakovni niz. Varijablu receivedString zatim moramo usporediti s definiranim znakovnim nizovima koju upisujemo u naredbu za tekst. Gotov kod se nalazi u nastavku.



Slika 31: Program za micro:bit kao upravljač

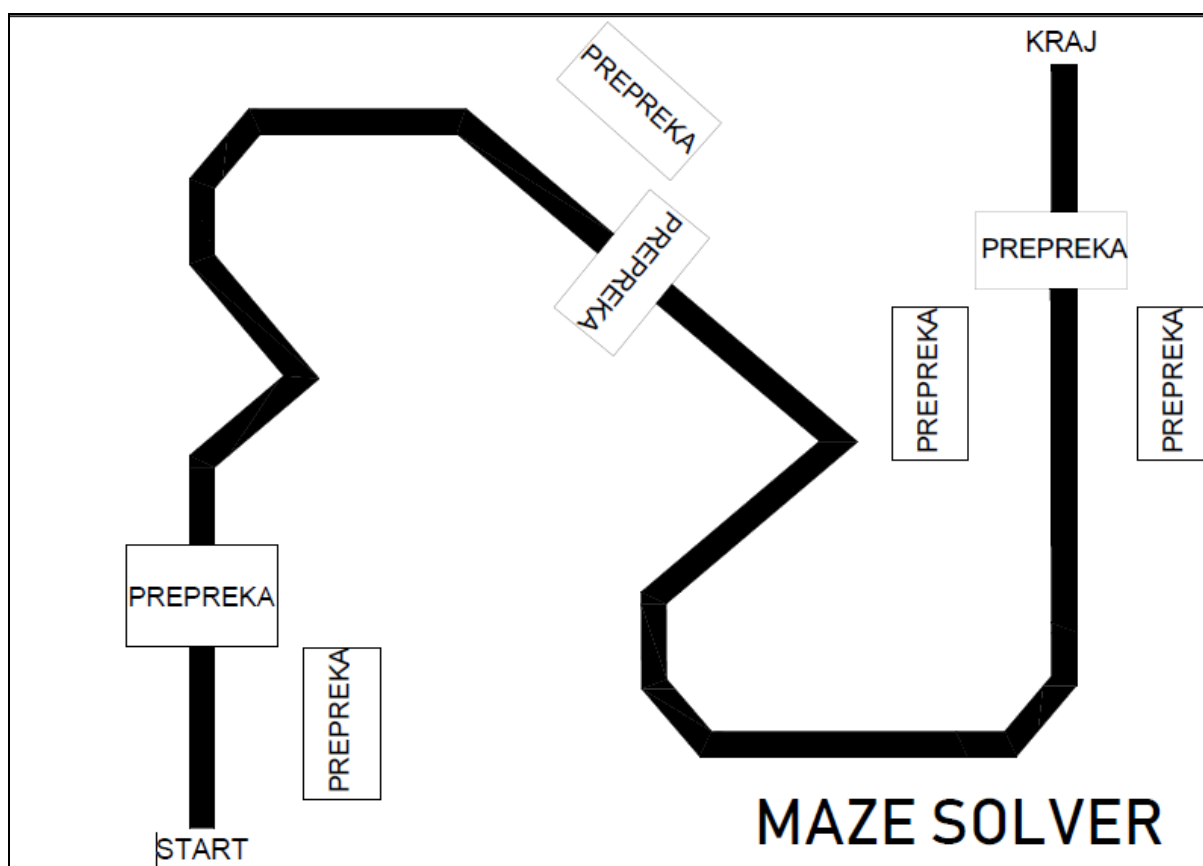


Slika 32: Program za micro:bit prijamnik

3.3. Prolazak kroz labirint s mBotom koristeći 2-3 ultrazvučna senzora

1. zadatak: Maze Solver 2

Prolazak mBota kroz fizički labirint s 3 ultrazvučna senzora.



Slika 33. Staza s fizičkim preprekama

Opis zadatka: Složi robota koji će pratiti crtu sve dok ne dođe do prepreke. Koristeći svoja 3 ultrazvučna senzora odrediti će na koju stranu mora zaobići prepreku ispred sebe. Ako dođe do prepreke ispred sebe, mora koristiti senzor s lijeve i desne strane da odluči s koje strane će zaobići prepreku.

Prvu prepreku mora zaobići s lijeve strane, drugu prepreku s desne strane dok kada dođe do treće ovisno da li će biti s lijeve ili desne strane mora zaobići prepreku ispred sebe.

Potrebna oprema

1. mBot

2. DC motor 2 komada
3. 4x AA baterije
4. Kućište za baterije
5. 3x ultrazvučni senzor

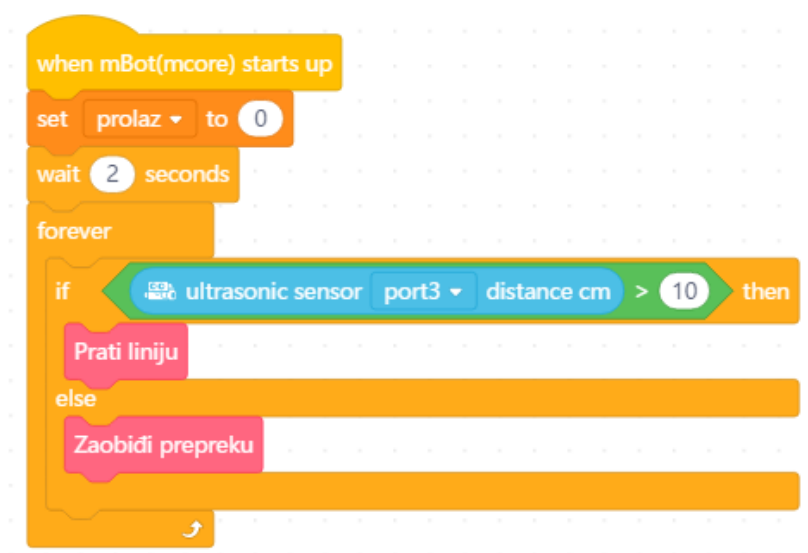


Slika 34. mBot

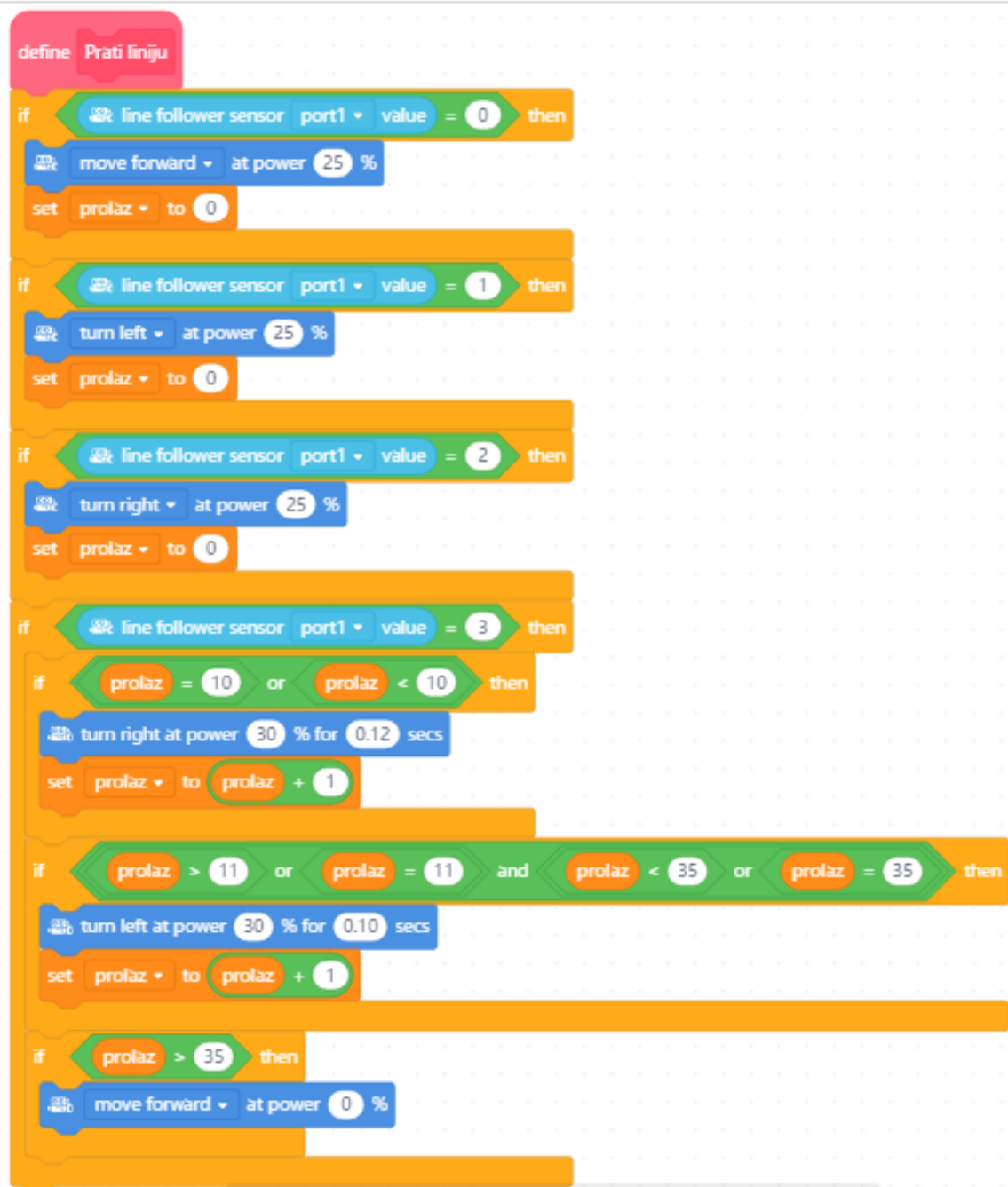
Programiranje

Za programiranje mBota koristi se *mBlock5*.

Rješenje



Slika 25. Glavni program



Slika 36. Program za praćenje linije



Slika 37. Program za zaobilaženje prepreke

Objašnjenja programa

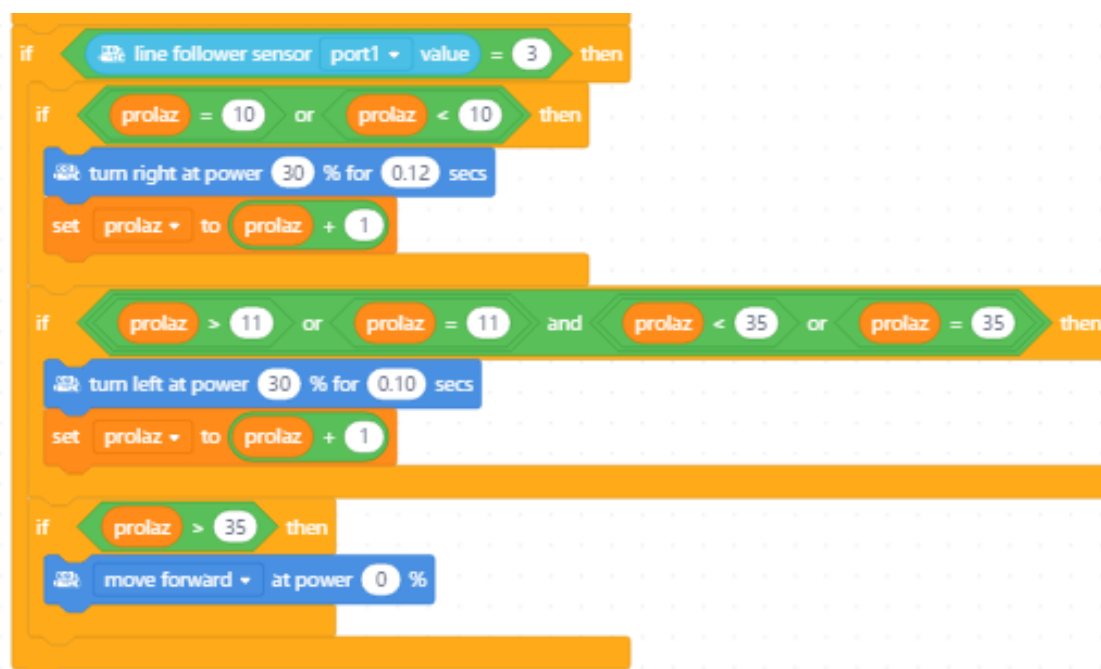
GLAVNI PROGRAM

Kada se mBot upali, inicijaliziraj varijablu *prolaz* na 0, pričekaj 2 sekunde i onda ovisno o senzoru za udaljenost (*port 3*) koji traži prepreke ispred sebe prati crtu ili zaobidi prepreku. Prepreka ispred mBota mora biti na udaljenosti manjoj od 10 cm. Ako nema prepreke ispred uđi u potprogram *Prati liniju*, a ako ima uđi u potprogram *Zaobidi prepreku*.

PRATI LINIJU

Senzor za praćenje linije nalazi se na *portu 1*. On daje 4 vrijednosti. Ako je vrijednost koju on daje 0 to znači da se mBot nalazi točno iznad linije i on može nastaviti voziti ravno. Ako je vrijednost koju on daje 1 to znači da se mBot nalazi s desne strane linije (nije točno iznad linije) i onda se mBot mora pomaknuti udesno. Ako je vrijednost 2, mBot je s lijeve strane linije i mBot se mora pomaknuti ulijevo.

Sada dolazimo do najgoreg slučaja, a to je kada senzor daje vrijednost 3. To znači da je on došao do bijele površine. I to bi trebalo značiti da je došao do kraja. No u nekim slučajevima to nije tako. Na primjer, u slučaju kada je preoštar zavoj robot zna „zapesti“ na bijeloj površini i ne prepoznati dalje crnu liniju. U tom slučaju koristimo se malim trikom. Kada robot dođe do bijele podloge pomaknut će se udesno i ulijevo i pritom provjeravati ako postoji crna linija ispod njega. To se radi ovim dijelom programa (Slika 38.):



Slika 38. Dio programa koji provjerava da li je robot stvarno stigao do cilja

Koristimo se varijablom *prolaz* kojoj je na početku programa dodijeljena vrijednost 0. Ovaj program funkcionira na način da se robot prvo pomakne 10 puta udesno i svaki put za 0.12sec. Prvi svakom okretaju za 0.12sec udesno dodaje se varijabli *prolaz* vrijednost 1 i provjerava ako je naišao na crnu crtu. Ako je neka nastavi dalje ju pratiti i opet resetiraj varijablu *prolaz* na 0. Ako nije neka se ponovno okrene sve dok ne dođe do druge if-naredbe koja ga okreće ulijevo 24 puta i to svaki put za 0.10sec. Pri svakom okretaju, opet se dodaje varijabli *prolaz* 1. Ako je

došao do linije, nastavlja ju pratiti i varijabla se vraća na 0. Ako nije dolazi se do treće if-naredbe koja kaže robotu da stane.

ZAObiDI PREPREKU

Kada robot dođe do prepreke mora očitati vrijednosti s senzora koji mu se nalaze s lijeve i desne strane. Senzor s lijeve strane nalazi mu se na *portu* 2, a senzor s desne strane na *portu* 4. Ako mu senzor s desne ili lijeve strane očitava vrijednost manju od 10cm onda mora zaobići prepreku ispred sebe sukladno tome. Ako je s desne strane prepreka, robot će se okrenuti ulijevo, malo pomaknuti naprijed (trajanje = 0.3sec, da si olakša zaobilaženje) i krenuti u krug oko prepreke sve dok ne dođe do crne linije koju će prepoznati senzor za praćenje linije na *portu* 1. To zaobilaženje radi se s naredbom: „left wheel turns at power 60%, right wheel at power 20%“. Ista stvar je i ako je prepreka s lijeve strane – okrene se udesno, malo pomakne naprijed i krene u krug oko prepreke ispred njega sve dok ne dođe do crne linije.

2. zadatak: Maze Solver 1

Prolazak mBota kroz fizički labirint s 2 ultrazvučna senzora.

Potrebna oprema

1. mBot
2. DC motor 2 komada
3. 4x AA baterije
4. Kućište za baterije
5. 3x ultrazvučni senzor

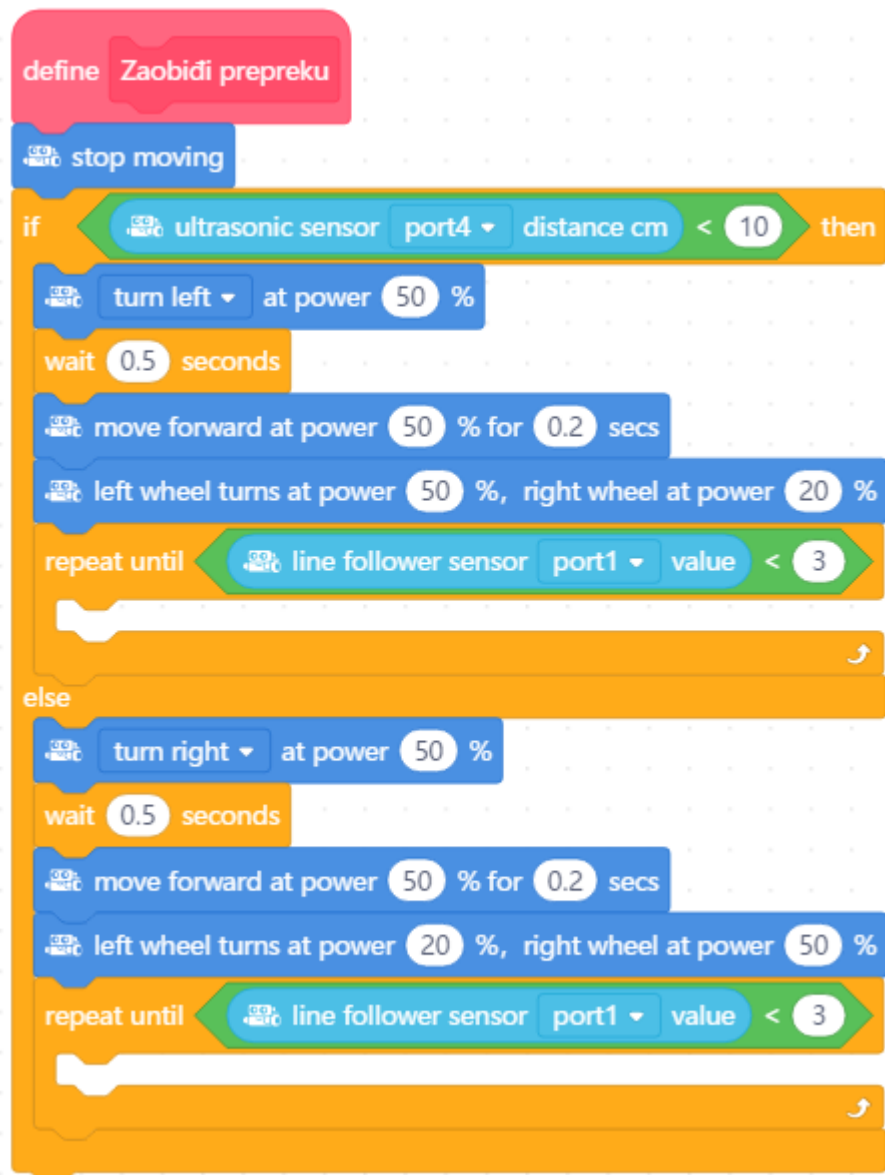
Programiranje

Za programiranje mBota koristi se *mBlock5*.

Rješenje

Zadatak je zapravo jednak kao i prošli samo što mu se potprogram *Zaobići prepreku* razlikuje. U ovom slučaju koristimo senzor koji će pratiti postoji li prepreka ispred robota i da li postoji prepreka s desne strane robota. Ako senzor vidi prepreku s desne strane onda zaobići prepreku ispred sebe s lijeve strane, a inače s desne (opisano se vidi u if-else naredbi). Robot zaobilazi prepreku na isti način kao i u prethodnom zadatku. Ovisno s koje strane je prepreka se okrene, pomakne se malo

naprijed, i krene u krug zaobilaziti oko prepreke sve dok ne naiđe na crnu liniju.



Slika 39. Program Zaobiđi prepreku - ZAD 2.

Zaključak

Edukacijski materijali koji su osmišljeni predstavljaju dobru podlogu za provedbu radionica robotike i natječaja. Kroz 3 predstavljena zadatka osigurano je stjecanje znanja o nekoliko senzora (ultrazvučni senzor, senzor za praćenje linije, senzor za boju). Osim znanja o senzorima, razvijaju se i programerske vještine poput analize problema, pronalazak nekoliko rješenja te odabir onog koji najbolje odgovara zadanom problemu.

Osim što su namijenjeni profesorima i ostalom edukacijskom osoblju, osmišljene materijale mogu koristiti i djeca koja se žele iskušati u nešto složenijim problemima ili pak roditelji koji svoju djecu žele izložiti STEM aktivnostima.

Sljedeći koraci koji bi se mogli odviti u budućnosti su osmišljavanje novih zadataka te unaprjeđenje predstavljenih. Još jedan od prijedloga za budući razvoj je korištenje novih edukacijskih sredstava.

4. Literatura

- [1] Institut za razvoj i inovativnost mladih. URL: <https://croatianmakers.hr/hr/projekti-irim-a/>
- [2] Digitalni portal Izradi!. URL: <https://izradi.croatianmakers.hr/>
- [3] Edukacijski materijali iz robotike. URL: <https://izradi.croatianmakers.hr/robotika/>
- [4] Uvodno o mBotu. URL: <https://izradi.croatianmakers.hr/project/uvodno-o-mbotu/>
- [5] Senzor za boju. URL: <https://izradi.croatianmakers.hr/project/senzor-za-boju-2/>
- [6] UV senzor. URL: <https://izradi.croatianmakers.hr/project/mbot-osnove-ultrazvucni-senzori/>
- [7] Senzor za praćenje linije. URL: <https://izradi.croatianmakers.hr/project/mbot-osnove-senzori-za-pracenje-linije/>
- [8] Babić, Stjepan; Finka, Božidar; Moguš, Milan. Hrvatski pravopis. Zagreb: Školska knjiga, 1995.