



ŠOLSKI CENTER CELJE

Srednja šola za kemijo, elektrotehniko in računalništvo

ROBOT

Raziskovalna naloga

Avtor:

David Podlesnik, E-4.a

Mentor:

Gregor Kramer, univ.dipl. inž

Kazalo

| | | |
|-------|--|----|
| 1. | Povzetek in ključne besede. povzetek cele naloge –hiter pregled naloge | 4 |
| 1.1 | Ključne besede..... | 4 |
| 2. | UVOD..... | 5 |
| 2.1 | Predstavitev naloge | 5 |
| 2.2 | Hipoteze..... | 5 |
| 2.3 | Opis raziskovalnih metod | 5 |
| 3. | Opis delovanja robota | 6 |
| 4. | Potek izdelave | 8 |
| 4.1 | Mikrokontroler Arduino Mega | 9 |
| 4.1.1 | Opis mikrokontrolerja | 9 |
| 4.1.2 | Specifikacije | 11 |
| 4.2 | 4.2 SENZORJI | 11 |
| 4.2.1 | TCS3200 Barvni senzor | 11 |
| 4.2.2 | Bluetooth modul | 12 |
| 4.2.3 | Laser | 14 |
| 4.3 | Regulacija motorjev | 15 |
| 4.3.1 | ČIP L298P | 15 |
| 4.3.2 | Motorji..... | 17 |
| 4.4 | Napajanje..... | 19 |
| 4.4.1 | Litijonske baterije | 19 |
| 4.4.2 | BMS sistem (battery management sistem)..... | 20 |
| 4.5 | Izdelava plastičnih elementov | 20 |
| 4.5.1 | Risanje v programu SolidWorks | 21 |
| 5. | Opis programa mikrokontrolerja..... | 22 |
| 5.1 | Izbira in inicializacija spremenljivk..... | 22 |
| 5.2 | Glavni program – inicializacija vhodno-izhodnih komponent..... | 23 |
| 5.3 | Glavni program – glavna zanka..... | 24 |
| 5.4 | Metodi za določanje barv – color() in colortu()..... | 25 |
| 6. | Razprava | 26 |
| 7. | Zaključek..... | 27 |

| | | |
|----|--------------------------|----|
| 8. | Zahvala | 28 |
| 9. | Viri in literatura | 29 |
| | Izjava | 30 |
| | Pojasnilo | 31 |

Kazalo slik

| | | |
|----------|---|----|
| Slika 1 | Robot | 6 |
| Slika 2 | Blokovna shema | 7 |
| Slika 3 | Oznaka modulov na robotu | 8 |
| Slika 4 | Shema vezja | 9 |
| Slika 5 | Arduino mega | 10 |
| Slika 6 | Čip Atmega 2560 | 10 |
| Slika 7 | TCS 3200 | 12 |
| Slika 8 | Bluetooth modul | 12 |
| Slika 9 | Prikaz napačnega izpisovanja na serijskem vmesniku | 13 |
| Slika 10 | Vezava BL modula | 13 |
| Slika 11 | Laser | 14 |
| Slika 12 | Motor sheeld | 15 |
| Slika 13 | Motor sheeld z oznakami | 15 |
| Slika 14 | Čip L298p | 16 |
| Slika 15 | Vezje znotraj čipa | 16 |
| Slika 16 | Specifikacije Čipa L298P | 16 |
| Slika 17 | Prvi motorji in njihova zgradba | 17 |
| Slika 18 | Oznaka plastičnega delca ki sem ga moral odstraniti | 18 |
| Slika 19 | Prikaz menjave motorjev | 18 |
| Slika 20 | Litijonske baterije | 19 |
| Slika 21 | Graf napetosti in toka skozi polnjenje | 19 |
| Slika 22 | Bms in piklop baterij | 20 |
| Slika 23 | Primer narisane delca 1. v programu SolidWorks | 21 |
| Slika 24 | Primer narisane delca 2. v programu SolidWorks | 21 |
| Slika 25 | Prikaz izdelave plastičnih delcev | 21 |
| Slika 26 | Določanje spremenljivk | 22 |
| Slika 27 | Izvajanje pred programa | 23 |
| Slika 28 | Izvajanje glavne zanke | 24 |
| Slika 29 | Izvajanje metode za merjenje barv | 25 |

1. Povzetek in ključne besede. povzetek cele naloge –hiter pregled naloge

Za raziskovalno nalogo sem raziskoval krmiljenje mobilnega robota. Za nalogo sem se odločil zaradi tekmovanja RoboCup Junior za katerega sem potreboval vozilo z različnimi senzorji in aktuarji. Mobilni robot mora na tekmovanju slediti črni črti in hkrati premagovati določene ovire. Odločil sem se za mobilnega robota z gosenicami, ki ga krmilim z mikrokrmilnikom Arduino Atmel 2560. Nanj sem namestil dva senzorja za zaznavanje barv, laser, Bluetooth modul, krmilnik motorjev in motorja za pogon gosenic ter napajalni sistem. Izdelal sem programsko kodo za sledenje črti in jo preizkusil na poligonu ter kodo za brezžično upravljanje z aplikacijo na pametnem telefonu.

1.1 Ključne besede

- TCS3200- senzor za merjenje barve
- Arduino - mikrokrmilnik
- AT mega 2560
- L2398P – krmilnik motorjev
- Bms – sistem za varno polnjenje in praznjenje baterij
- 3d tiskanje – izdelovanje izdelkov plastičnih mas

2. UVOD

2.1 Predstavitev naloge

Moja želja je bila izdelati robota, ki bo preko barvnih senzorjev sledil črti na poligonu, možna pa bo tudi komunikacija preko bluetooth modula. Poligon ima belo podlago in na njej narisane črne črte. Zastavil sem si cilj, da mora robot po poligonu voziti čim hitreje. Menjava programa za vožnjo po črti in upravljanje preko bluetooth modula je izvedena s tipko. Robot mora začeti z vožnjo takoj, ko vključimo stikalo za vklop. Robot mora biti tudi varen za uporabo zato bom s senzorji preprečil, da se zaleti v morebitno oviro. Robot z vsemi senzorji mora biti cenejši od 100€ oziroma od cene robota Mindstorm LEGO NXT.

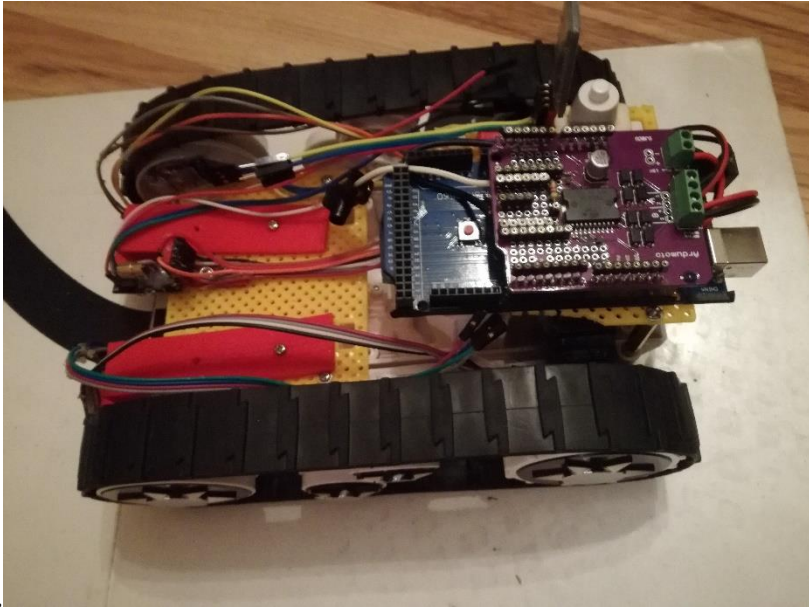
2.2 Hipoteze

- robot sledi črti črne barve in se ustavi pred oviro,
- robot prevozi klanec 40°,
- robota je mogoče brezžično voditi,
- cena robota ne sme znašati več kot 100€.

2.3 Opis raziskovalnih metod

Pri raziskovanju sem najprej določil zahteve za delovanje in nato preiskal tržišče in spletne strani s podobno tematiko. Za potrebe krmiljenja sem izbral mikrokrmilnik Arduino, ki omogoča hitro in poceni implementacijo programske kode.

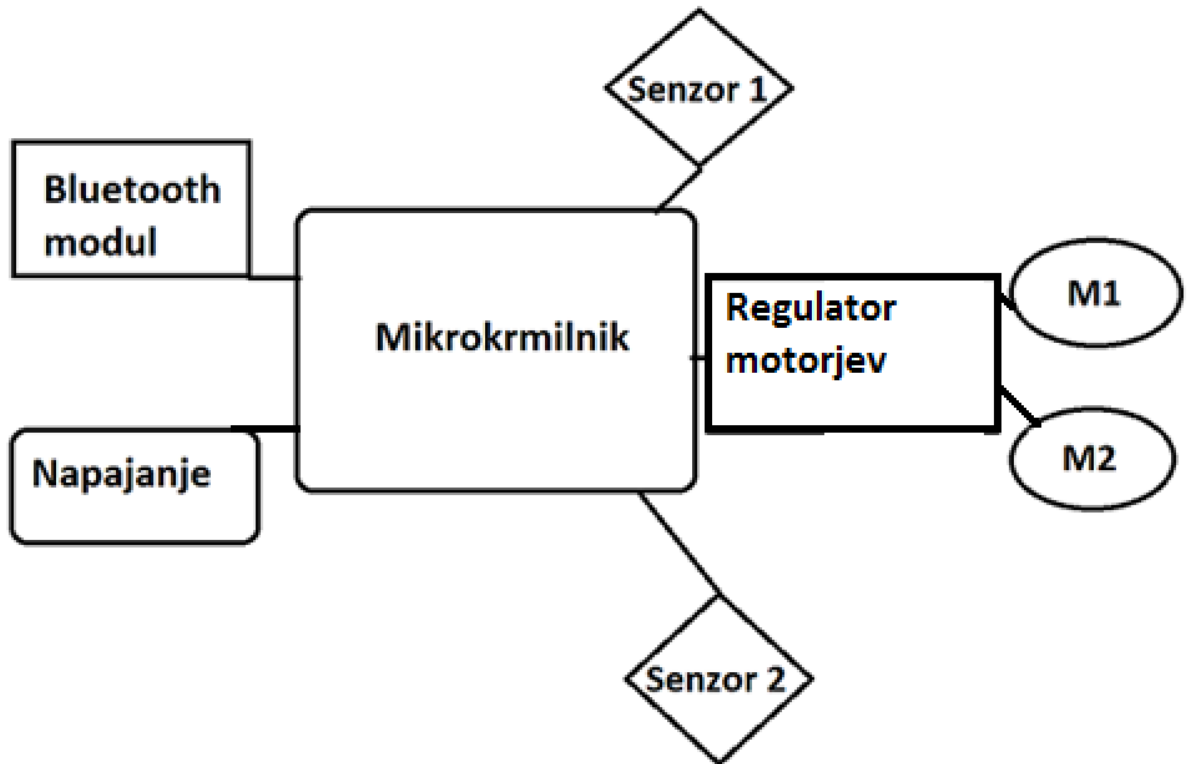
Programiranja sem se naučil samostojno in pri praktičnem delu v šoli. Robotsko vozilo sem kupil preko spleta in za moje potrebe predelal. Za potrebe izdelave namenskih delov robota sem preučil 3D tiskanje in programsko opremo za modeliranje.



Slika 1 Robot

3. Opis delovanja robota

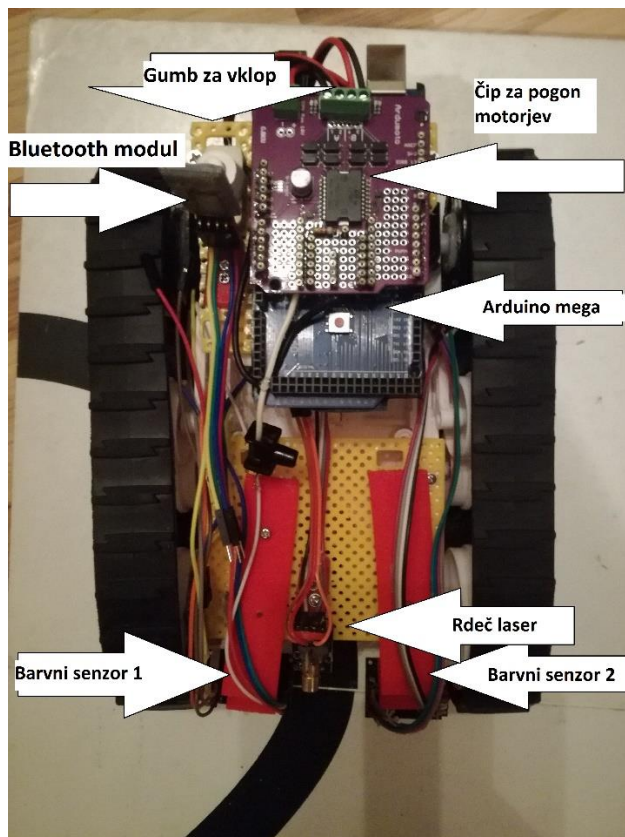
Robota krmilim preko mikrokrmilnika Arduino mega, ki ima vgrajen čip ATmega 2560. Na njem se izvajajo ukazi za izhode in določene funkcije za branje vhodov. Robot ima dva senzorja, ki zaznavata tri različne barve (modro, rdečo in zeleno). Na podlagi teh treh barv in pripadajočih vrednosti robot zaznava oziroma loči črno črto in belo podlago. Za krmiljenje motorja je uporabljeno integrirano vezje L298P. Za daljinsko upravljanje robota, pa je skrbi programska koda za Bluetooth modul. Robot ima senzorje priključene na sprednji levi in desni strani robota. Če robot zazna črno barvo z levim senzorjem se robot ustavi in popravi smer v levo, če pa zazna črno barvo desni sprednji senzor, popravi smer v desno. Če med delovanjem oziroma vožnjo pritisnemo tipko, ki je nameščena na robotu se program zamenja in se ga lahko začne krmiliti z android telefonom preko krmilnika Bluetooth.



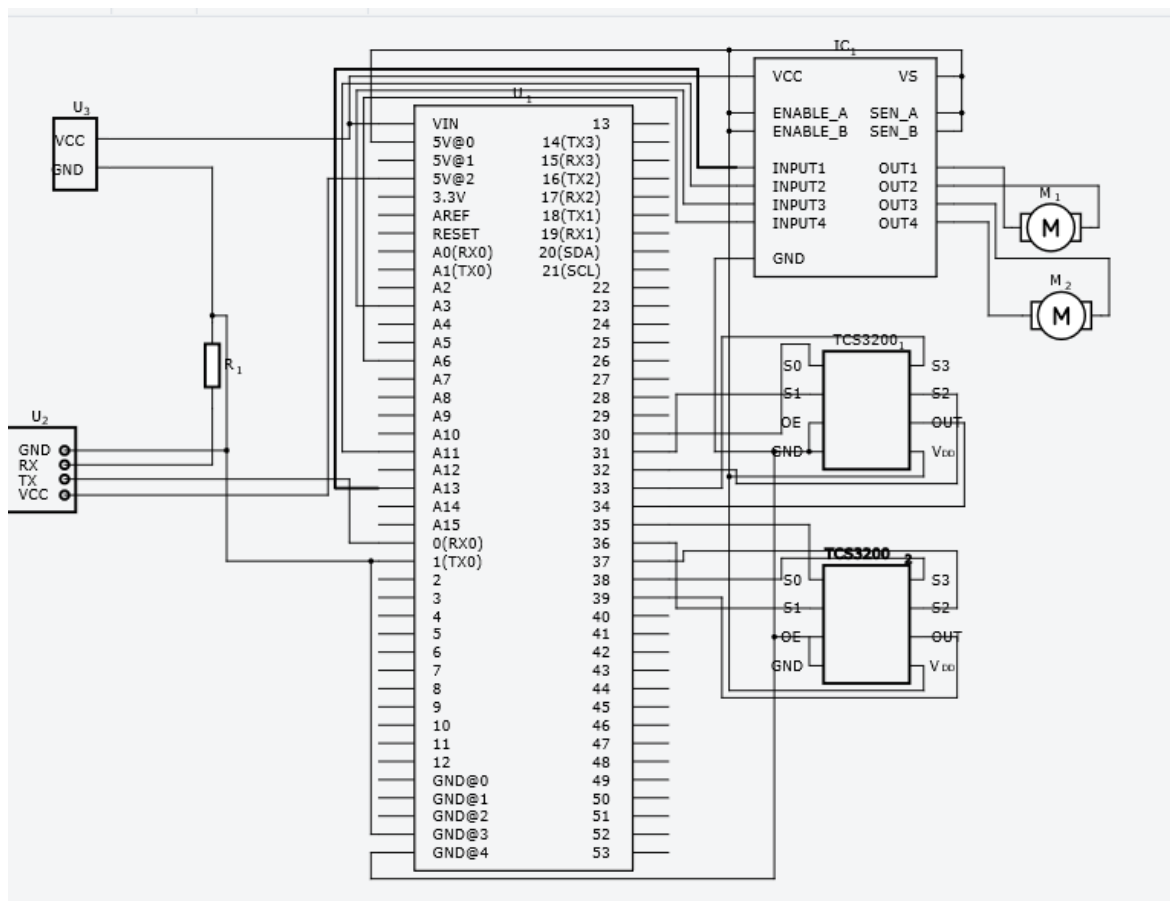
Slika 2 Blokovna shema

4. Potek izdelave

Na težave sem naletel že pri samem naročilu ohišja z motorji saj so bili motorji v njem prehitri in nenatančni za reguliranje. Zaradi tega se je robot, ki je želel slediti črti, peljal prehitro in klub zaznavi barv ni mogel pravočasno zavirati in ustrezno nadaljevati poti. Odločil sem se, da motorje nadomestim z servomotorji, ki sem jim odstranil blokiranje obratov za do 360 stopinj. Za krmiljenje motorjev sem sprva uporabil čip L293d, ki pa je bil tokovno preveč obremenjen. Čip L293d sem nato nadomestil z čipom L298P, ki je vgrajen v ploščo (shield), ki se namesti direktno na mikrokrmilnik. Težave sem imel tudi pri definiranju barvnih senzorjev, ki niso imeli na razpolago zaznavanje bele in črne barve, ampak samo rdečo, modro in zeleno. Pri nalaganju kakršnegakoli programa na krmilnik pa sem moral paziti, da ni bil priklopljen kakšen senzor, ki bi pošiljal signale v serijski vmesnik, ker pri tem ne more čip zaznati, da se nanj nalaga nov program.



Slika 3 Oznaka modulov na robotu

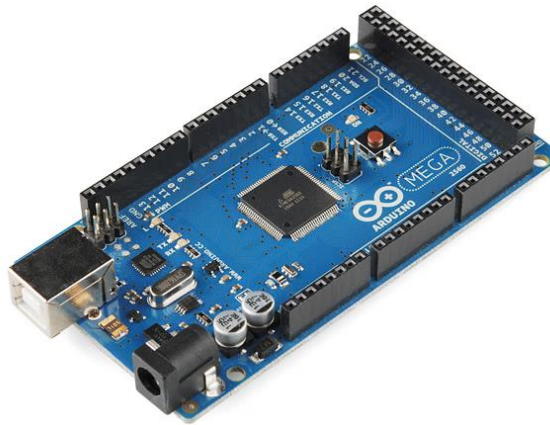


Slika 4 Shema vezja

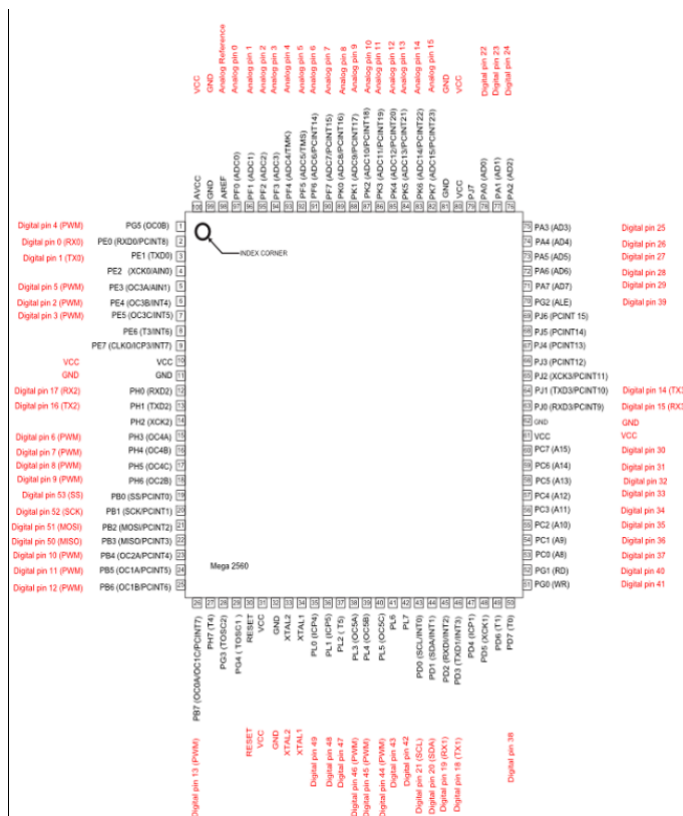
4.1 Mikrokrmilnik Arduino Mega

4.1.1 Opis mikrokontrolerja

Arduino mega je mikrokontroler z vgrajenim čipom ATmega 2560 in deluje z hitrostjo 16 MHz. Mikrokontroler priklopimo preko 2.0 usb kabla na računalnik in v programu določimo logično zaznavanje vhodov in oddajanje signalov na točno določenih nožicah. Za napajanje pa potrebuje od 5 do 12v (najbolj priporočen je priklop na 7-12v). Arduino že ima vgrajen modul, ki čipu poskrbi stabilizirano napajanje na točno 5v. Arduino mega ima na voljo 54 digitalnih nožic kar pomeni da lahko zaznava ali oddaja samo logično enko ali ničlo ter 9 analognih izhodov ter 16 analognih vhodov (devet od teh je lahko analognih izhodov).



Slika 5 Arduino mega



Slika 6 Čip Atmega 2560

4.1.2 Specifikacije

| | |
|-------------------------------------|--------------------|
| <u>Mikrokontroler</u> | <u>Atmega 2560</u> |
| <u>Operacijska napetost čipa</u> | <u>5V</u> |
| <u>Priporočena vhodna napetost</u> | <u>7- 12V</u> |
| <u>Vhodna napetost</u> | <u>5 – 20V</u> |
| <u>Digitalni pini</u> | <u>54</u> |
| <u>Analogni vhodi</u> | <u>16</u> |
| <u>Tokovna obremenitev na pin</u> | <u>40mA</u> |
| <u>Tokovna obremenitev pri 3,3V</u> | <u>50mA</u> |
| <u>Spomin</u> | <u>256 KB</u> |
| <u>Hitrost oscilatorja</u> | <u>16Mhz</u> |

4.2 4.2 SENZORJI

4.2.1 TCS3200 Barvni senzor

Barvni senzor TCS3200 priključimo na napajalno napetost (2.7 do 5.5v). Senzor je zgrajen iz osmih pinov. Pina GND in OE priklopimo na 0V, Vcc na 5v, out je izhod ki je namenjen branju odbite frekvence, ki jo pošljemo skozi pine S0, S1, S2, S3.

| <u>S1</u> | <u>S2</u> | BARVA |
|-----------|-----------|--------|
| LOW | LOW | Rdeča |
| LOW | HIGH | Modra |
| HIGH | LOW | Prazno |
| HIGH | HIGH | Zelena |

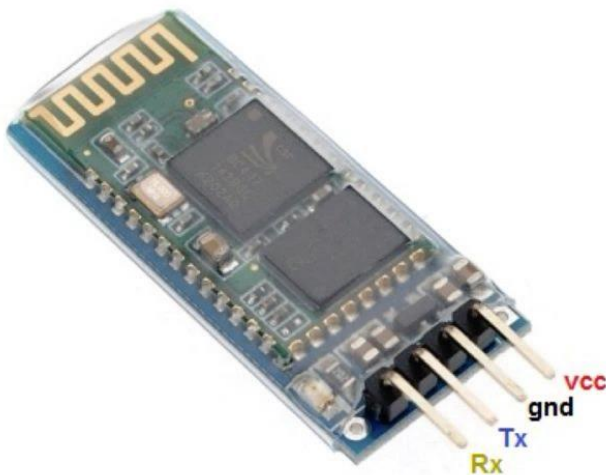
| <u>S0</u> | <u>S1</u> | Odbita frekvenca |
|-----------|-----------|------------------|
| LOW | LOW | 0% |
| LOW | HIGH | 2% |
| HIGH | LOW | 20% |
| HIGH | HIGH | 100% |



Slika 7 TCS 3200

4.2.2 Bluetooth modul

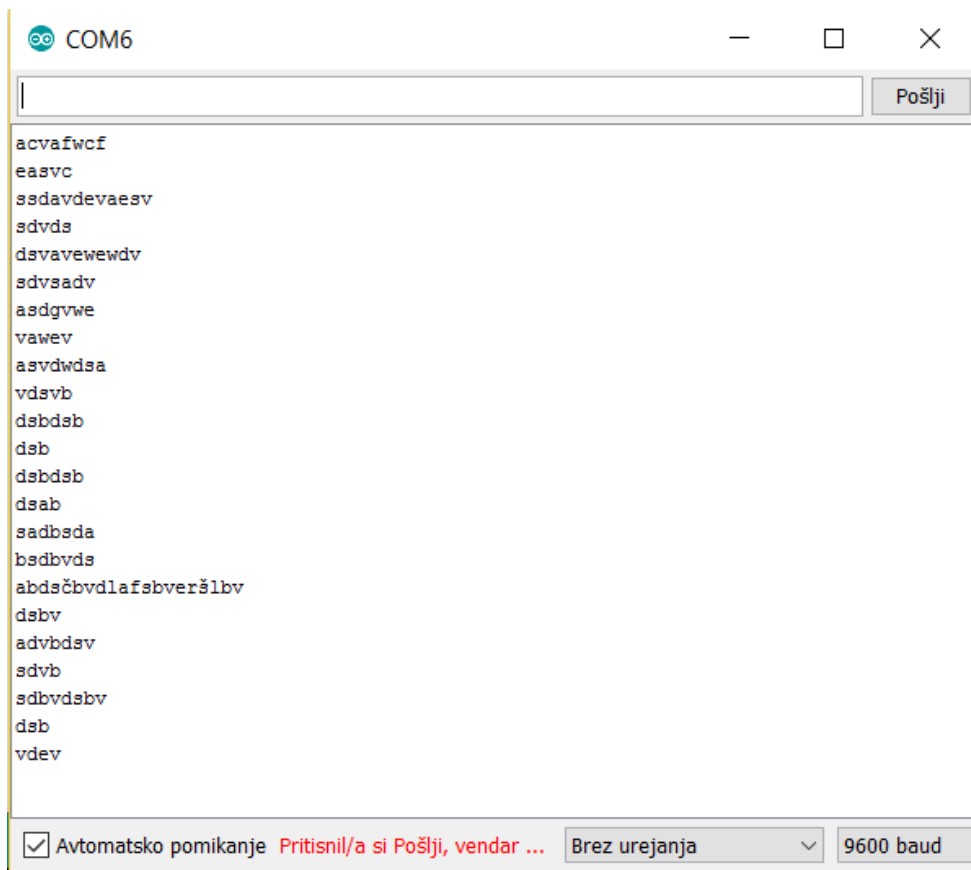
Bluetooth modul ima štiri pine. Na pin Gnd priključimo 0v na pin Vcc priključimo 5v. Pin TX pomeni angleško transmitter (oddajnik) in ga povežemo preko upora na RX pri mikrokrmilniku. Pin RX po angleško reciver (sprejemnik) pa je namenjen sprejemanju signalov iz mikrokrmilnika.



Slika 8 Bluetooth modul

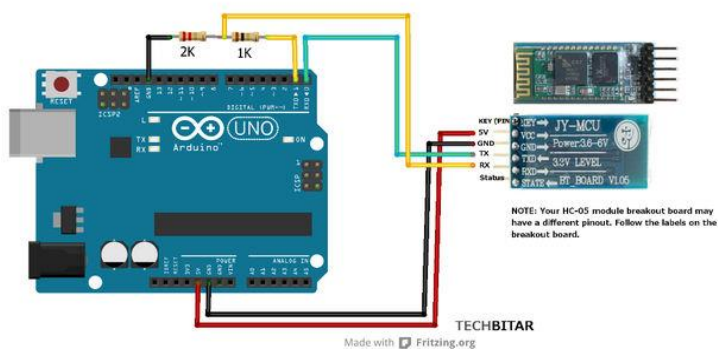
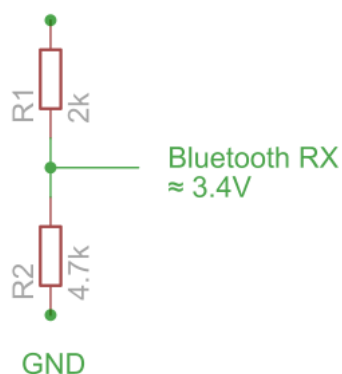
4.2.2.1 Povezava modula Bluetooth na mikrokontroler

Pri povezavi modula z mikrokrmilnikom sem imel najprej težave, kar sem ugotovil, ko sem preveril signale na serijskem vmesniku na programu.



Slika 9 Prikaz napačnega izpisovanja na serijskem vmesniku

Arduino TX

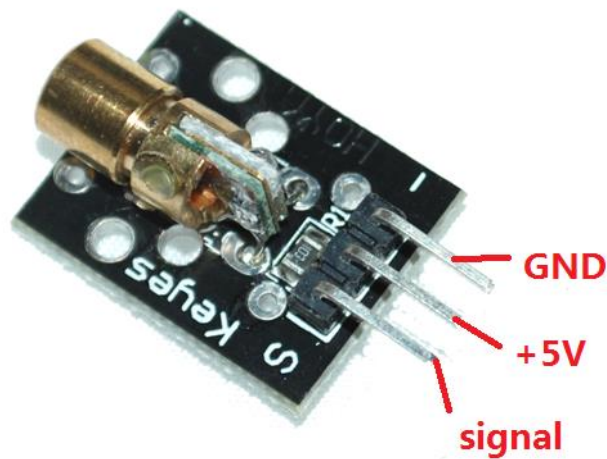


Slika 10 Vezava BL modula

4.2.3 Laser

Za lepši izgled pri nakazovanju smeri vožnje po črti sem uporabil rdeč laser. Modul laserja je zgrajen iz treh nožic, Vcc, Gnd, in S.

Na Vcc mu priklopimo napajanje 5 V, Gnd 0 V, na S pa lahko priklopimo na mikrokrmilnik in ga prižigamo in ugašamo z logičnimi funkcijami. Nožico S sem priklopil direktno na 5V, ker nisem potreboval prižiganja in ugašanja laserja med vožnjo.



Slika 11 Laser

4.3 Regulacija motorjev

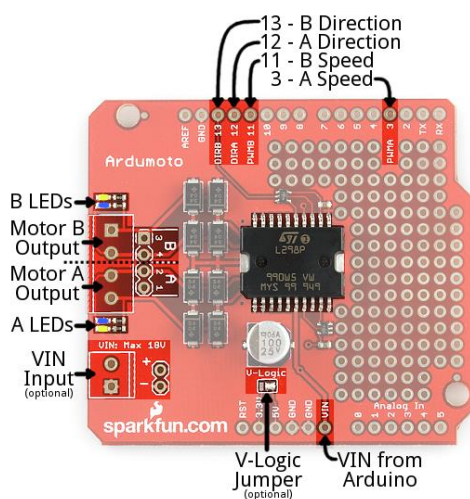
4.3.1 ČIP L298P

Za regulacijo motorjev sem sprva uporabil čip L293d. Čip L293d je namenjen za maksimalne tokove do 1A, zato sem se odločil da za izdelavo uporabim čip L298p ki ima maksimalno tokovno omejitev 2A.

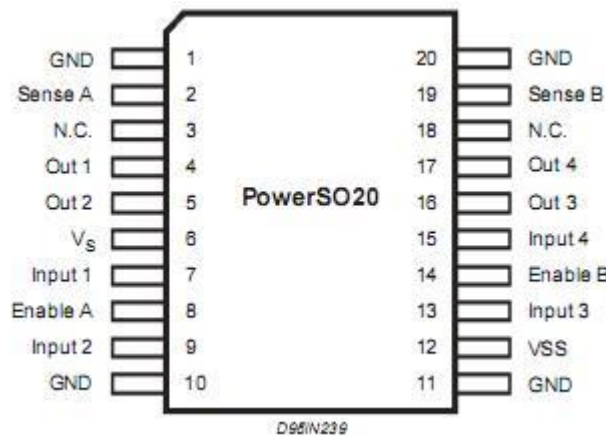


Picture provided by the manufacturer

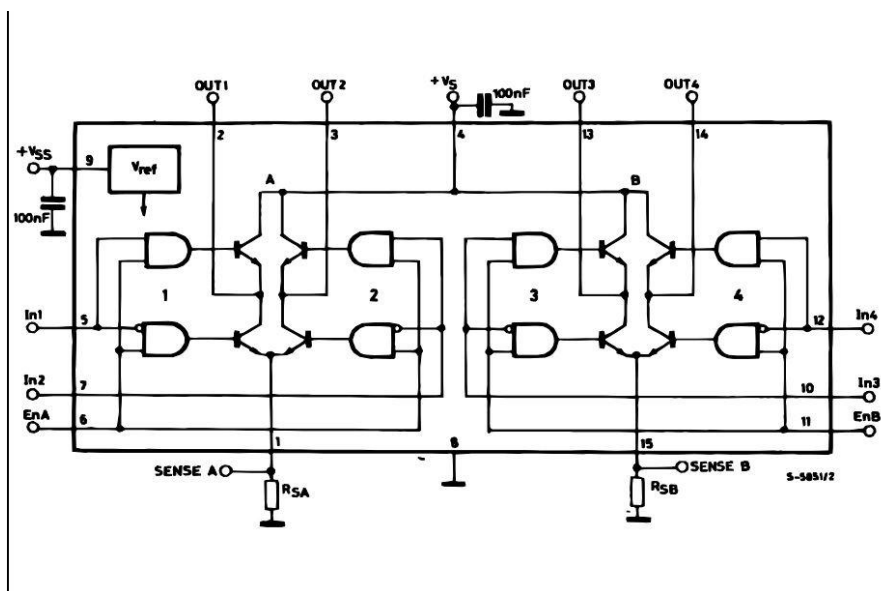
Slika 12 Motor shield



Slika 13 Motor shield z oznakami



Slika 14 Čip L298p



Slika 15 Veze znotraj čipa

4.3.1.1 Specifikacije čipa

| Symbol | Parameter | Value | Unit |
|----------------|---|---------------|-------------|
| V_S | Power Supply | 50 | V |
| V_{SS} | Logic Supply Voltage | 7 | V |
| V_i, V_{en} | Input and Enable Voltage | -0.3 to 7 | V |
| I_o | Peak Output Current (each Channel) - Non Repetitive ($t = 100\mu s$) - Repetitive (80% on -20% off; $t_{on} = 10ms$) - DC Operation | 3 2.5 2 | A A A |
| V_{sens} | Sensing Voltage | -1 to 2.3 | V |
| P_{tot} | Total Power Dissipation ($T_{case} = 75^\circ C$) | 25 | W |
| T_{op} | Junction Operating Temperature | -25 to 130 | $^\circ C$ |
| T_{stg}, T_j | Storage and Junction Temperature | -40 to 150 | $^\circ C$ |

Slika 16 Specifikacije Čipa L298P

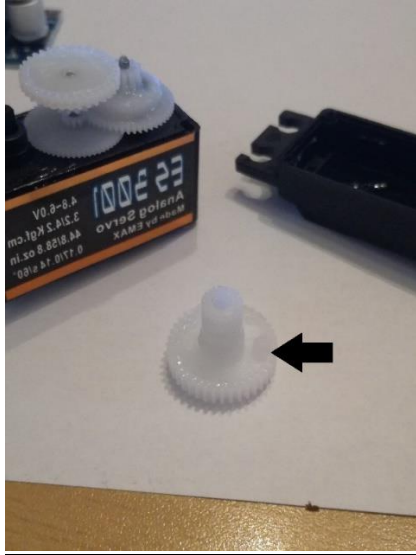
4.3.2 Motorji

Prve motorje sem imel že vgrajene v plastičnem ohišju z gosenicami, vendar sem opazil da pri daljši vožnji robot začne voziti prehitro in kljub zaznavi črte se ne more pravočasno ustaviti in tako pelje skozi njo. Ker hitrosti nisem mogel natančno regulirati z pwm modulacijo, sem moral izdelati nove. Za motorje sem potem uporabil servo motorje, vendar jih je bilo potrebno malo predelati. Servo motorji se vrtijo le za 180 stopinj, zato sem jih razdril in odščipnil plastično varovalo ter odvzel potenciometer. Na motor sem prispajkal dva kabla, ki sem jih povezal na čip za reguliranje motorja.

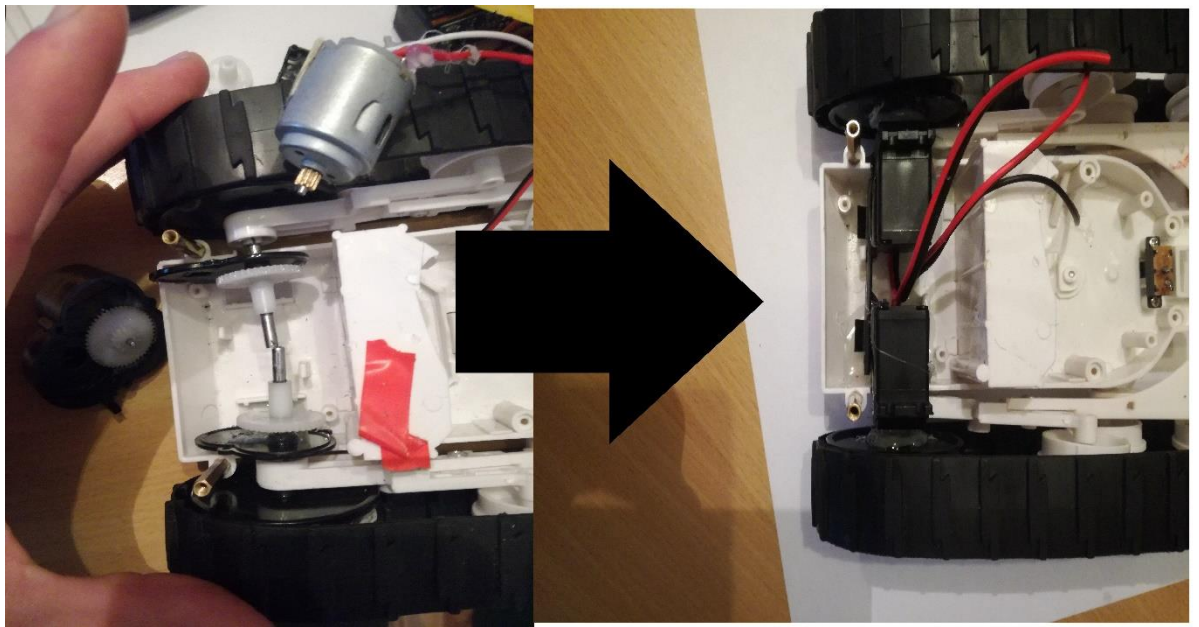


Slika 17 Prvi motorji in njihova zgradba

4.3.2.1 Predelava servomotorja



Slika 18 Oznaka plastičnega delca ki sem ga moral odstraniti



Slika 19 Prikaz menjave motorjev

4.4 Napajanje

4.4.1 Litijonske baterije

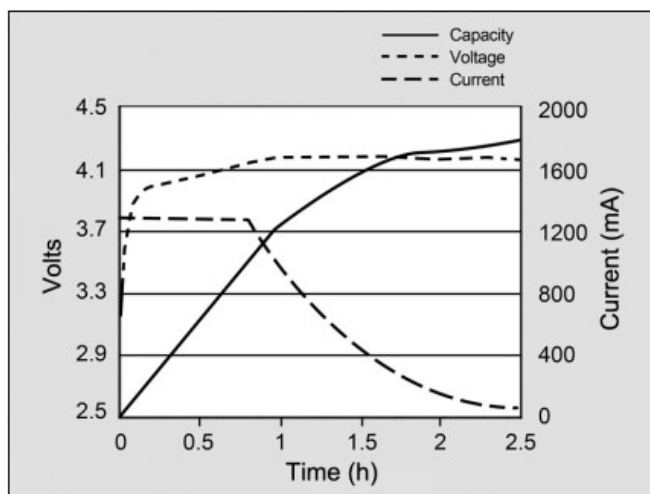
Za napajanje sem se odločil za litijonske baterije zaradi večkratne uporabe in daljšega časa uporabe baterije. Litijeve baterije so baterije, ki imajo povprečno napetost 3,7 V in kapacitivnost 2000mAh. Baterijo z največjo kapaciteto je do sedaj izdelal Panasonic (3500mAh). Litijonske baterije sem dobil iz starih baterijskih vrtljnikov ki se jim je pokvaril BMS. Stare baterije ki ne delajo, imajo nekaj uporabnih baterij, vendar jih je treba razdreti, ločiti nato vsako posebej napolniti in preizkusiti.



Slika 20 Litijonske baterije

4.4.1.1 Napetost in napajanje baterij

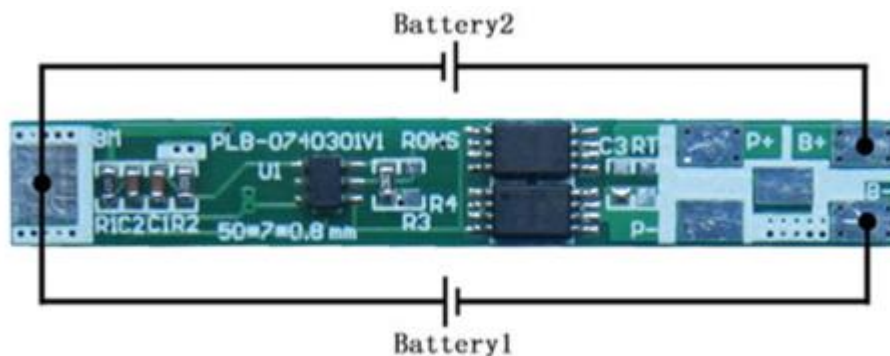
Napetost na vsaki posamezni bateriji ne sme pasti pod 3 V in ne sme preseči 4,2 V, saj se ob tem primeru baterija uniči ob večji napaki tudi vžge. Baterijo začnemo polniti z manjšo napetostjo in jo postopoma glede na naraščanje napetosti na bateriji zvišujemo. Tok pa sprva povečano in postopoma glede na nivo napetosti zmanjšujemo. Tok napajanja se giblje od 0,5 do 5A odvisno od tehnologije posamezne baterije.



Slika 21 Graf napetosti in toka skozi polnjenje

4.4.2 BMS sistem (battery management system)

BMS je sistem za zaščito baterij pred prenapetostjo in preobremenitvijo. BMS drži napetost do 4,2 V nato napajanje izklopi, zato lahko baterije napajamo z navadnim adapterjem. Pri dolgotrajni neuporabi baterij, napetost na baterijah začne padati in ko pride razlike med napetostjo prve in druge baterije več kot 0,5V. BMS baterije več ne zazna in polni ter prazni le eno baterijo, pri tem pa pride da naprava ne deluje več pravilno. Maksimalna obremenitev BMS - a je 3A kar pa je pri mojem projektu več kot dovolj.



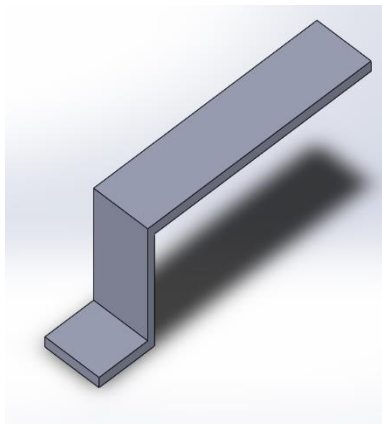
Slika 22 Bms in piklop baterij

4.5 Izdelava plastičnih elementov

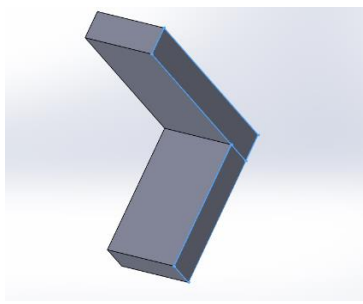
Pri izdelavi sem za priključitev in pravo pozicijo senzorjev moral izdelati plastične nosilce za senzorje. Za izdelavo plastičnih delov sem se odločil za 3d tiskalnik saj je zelo enostaven in hiter za uporabo. Najprej sem vsak del dobro načrtoval nato začel z modeliranjem v programu SolidWorks. V programu sem že na začetku nastavil pravo mero, ker sem si tako prikrajšal čas pri končnem definiranju 3d tiskalnika. Ko sem risbo končal sem program shranil v STL obliki, ker je tako bil program pripravljen na 3d tisk v programu Repetier Host. Program Repetier Host ima možnost nastavljanja hitrosti tiskanja in prilagajanje pozicije ter obračanje izdelka za lažje tiskanje. Ko je bil izdelek končan sem ga dobro pritrdil na robota in nanj lahko dobro pričvrstil senzor, saj je bil narejen v točno pravi meri pod pravim kotom.

4.5.1 Risanje v programu SolidWorks

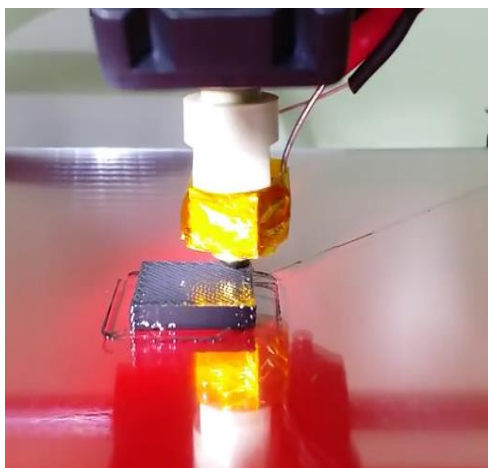
Program SolidWorks je profesionalen program namenjen risanju in načrtovanju predmetov v 3d obliki. Pri risanju predmetov moramo najprej narediti skico in nastaviti prave mere da lahko potem vse skupaj pretvorimo v 3d obliko.



Slika 23 Primer narisane delca 1. v programu SolidWorks



Slika 24 Primer narisane delca 2. v programu SolidWorks



Slika 25 Prikaz izdelave plastičnih delcev

5. Opis programa mikrokrmilnika

Programska koda je napisana v okolju Arduino. Programski jezik je enostaven in lahek za uporabo. Programsko okolje ima možnost izpisovanja in pregledovanja vrednosti v serijskem vmesniku kar je pri delu zelo uporabno saj lahko hitro in enostavno popravljamo program.

5.1 Izbira in inicializacija spremenljivk

Spremenljivke so vrednosti, ki jih potrebujemo za lažjo organizacijo izvajanja programa. Vhodno izhodne komponente priključimo na nožice, ki jih predhodno deklariramo kot konstante.

```
int hitrost =170;
int hitrost2 =110; //Definiranje spremenljivk za določanje hitrosti

const int stu0 = 30;
const int stu1 = 31;
const int stu2 = 32;
const int stu3 = 33; // DEFINIRANJE BARVNEGA SENZORJA 1
const int outtu = 34;

const int s0 = 35;
const int s1 = 36;
const int s2 = 37; // DEFINIRANJE BARVNEGA SENZORJA 2
const int s3 = 38;
const int out = 39;

int motorB2 = 11;
int motorB1 = 13;
int redLed = 49; // DEFINIRANJE IZHODOV ZA MOTORJE IN LED PINE
int motorA2 = 3;
int motorA1 = 12;

int redtu = 0;
int greentu = 0; //SPREMENLJIVKE ZA DOLOČANJE VALUTE SENZORJEV
int bluetu = 0;

int red = 0;
int green = 0;
int blue = 0;
```

Slika 26 Določanje spremenljivk

5.2 Glavni program – inicializacija vhodno-izhodnih komponent

Program začnemo z zanko »void setup()«, ki se izvede le enkrat. V njej določimo kateri pini so vhodi in kateri izhodi in začetek serijskega vmesnika.

```
void setup()
{
  Serial.begin(250000);
  pinMode(s0, OUTPUT);
  pinMode(s1, OUTPUT);
  pinMode(s2, OUTPUT);
  pinMode(s3, OUTPUT);
  pinMode(out, INPUT);

  pinMode(motorA1, OUTPUT);
  pinMode(motorA2, OUTPUT);
  pinMode(motorB1, OUTPUT); // DEFINIRANJE KATERI PINI SO VHODI IN KATERI IZHODI
  pinMode(motorB2, OUTPUT);

  pinMode(redLed, OUTPUT);

  digitalWrite(s0, HIGH);
  digitalWrite(s1, HIGH);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
```

Slika 27 Izvajanje pred programa

5.3 Glavni program – glavna zanka

Ko se »void setup()« zaključi se začne izvajati glavna zanka »void loop()«. V njej se začne izvajati glavna programska koda. V glavni zanki se izvajajo tudi funkcije ali metode za posamezne funkcionalnosti (color(), colortu(), itd.).

```
void loop()
{
  color();
  colortu();                                // ZAČETEK ZANK ZA DOLOČANJE BARV

  Serial.print("R Intensity:");
  Serial.print(red, DEC);
  Serial.print(" G Intensity: ");
  Serial.print(green, DEC);
  Serial.print(" B Intensity : ");          // IZPISOVANJE BARV NA SERIJSKEM VMESNIKU
  Serial.print(blue, DEC);

  Serial.print(".....R valuta:");
  Serial.print(redtu, DEC);
  Serial.print(" G valuta: ");
  Serial.print(greentu, DEC);
  Serial.print(" B valuta : ");
  Serial.print(bluetu, DEC);

  voznja();                                // ZAČETEK ZANKE ZA VOŽNJO

}
```

Slika 28 Izvajanje glavne zanke

5.4 Metodi za določanje barv – color() in colortu()

Metoda za določanje barv se začne tako, da najprej prebere rdečo, modro in nato še zeleno barv. Zanka rdeče barve se izvaja tako, da najprej postavi izhoda na senzorju s2 in s3 na logično 0, nato da pulz na izhodno nožico in nato prebere hitrost odbite frekvence za rdečo barvo. Modro barvo prebere tako, da postavi izhod s3 na logično 1, nato odda pulz na izhodno nožico in prebere hitrost odbite frekvence. Za določanje zelene barve pa postavi nožico s2 na logično 1, nato odda pulz na izhodno nožico, prebere hitrost odbite frekvence in ga zapiše v spremenljivko za zeleno barvo.

```
void color() //ZANKI V KATERIH DOLOČA BARVO SENZORJEV Barve
{
  digitalWrite(s2, LOW);
  digitalWrite(s3, LOW);
  //count OUT, pRed, RED
  red = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
  digitalWrite(s3, HIGH);
  //count OUT, pBLUE, BLUE
  blue = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
  digitalWrite(s2, HIGH);
  //count OUT, pGreen, GREEN
  green = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
}

void colortu()
{
  digitalWrite(stu2, LOW);
  digitalWrite(stu3, LOW);
  //count OUT, pRed, RED
  redtu = pulseIn(outtu, digitalRead(outtu) == HIGH ? LOW : HIGH);
  digitalWrite(stu3, HIGH);
  //count OUT, pBLUE, BLUE
  bluetu = pulseIn(outtu, digitalRead(outtu) == HIGH ? LOW : HIGH);
  digitalWrite(stu2, HIGH);
  //count OUT, pGreen, GREEN
  greentu = pulseIn(outtu, digitalRead(outtu) == HIGH ? LOW : HIGH);
}
```

Slika 29 Izvajanje metode za merjenje barv

6. Razprava

Raziskovalno nalogo sem uspešno končal. Pri raziskovanju sem naletel na kar nekaj težav, vendar pa sem jih uspel uspešno odpraviti. Pri raziskovanju sem moral zamenjati motorje, zamenjati pozicijo barvnih senzorjev in vseskozi glede na spremembe konstrukcije spreminjati še program. Pri komunikaciji z Bluetooth sistemom, sem imel rahlo težavo pri povezavi ker ni bila enaka vezava kot pri navadnem Bluetooth sistemom na programator.

Pred raziskovalno nalogo sem si postavil hipotezo:

- robot sledi črti črne barve in se ustavi pred oviro:

Robot brez težav sledi črti vendar še ima težavo pri debelini črte saj za debelejšo črto mora biti senzor postavljen višje .

- Robot uspe prevoziti klanec ki je lahko večji od 40°:

Hipotezo lahko potrdim, vendar je odvisna od trenja podlage.

- robota je mogoče brezžično voditi:

Hipotezo lahko potrdim, saj lahko robota upravljam brezžično.

- cena robota ne sme znašati več kot 100€:

Hipotezo lahko potrdim, saj cena komponent robota skupno znaša okoli 90€, vendar sem zaradi nakupa napačnih komponent vse skupaj porabil okoli 130€, kar pa še vedno ne presega mojega cilja pri postavitvi hipoteze.

7. Zaključek

Moj končni izdelek deluje pravilno, saj robot sledi črti in brezžična komunikacija z pametnim telefonom deluje brez težav.

Izdelava robota je bila bolj zapletena kot sem si sprva predstavljal. Pri izdelavi sem naletel na veliko težav, ki sem jih na koncu uspel rešiti. Da sem lahko nalogo dokončal sem moral skozi celotno raziskovanje menjavati komponente in spreminjati program.

Všeč mi je, saj lahko mobilnega robota uporabljam ne samo za sledenje črni črti, temveč ga lahko upravljam brezžično. Robot ki sem ga naredil je praktično zelo uporaben in lahek za uporabo saj ga je preizkusil moj mlajši brat. Znanje ki sem ga dobil pri izdelavi izdelka mi bo zelo koristilo pri mojih naslednjih projektih. Dobil sem dodatna znanja pri pisanju programov, uporabo enosmerih motorjev, Bluetooth komunikaciji...

8. Zahvala

Najprej bi se rad zahvalil mentorju gospodu prof. Gregorju Kramerju za pomoč in potrpežljivost pri delu.

Zahvalil bi se vsem, ki so mi pomagali pri kakršnem koli delu ali mi podali dobro mnenje ali vspodbudo.

Nato pa bi se rad zahvalil mojemu prijatelju Tilenu Podlesniku, saj mi je pomagal pri izdelavi plastičnih izdelkov in mi posodil 3d tiskalnik.

9. Viri in literatura

Domača stran arduino:

<https://www.arduino.cc/>

Stran za pregled specifikacij čipov in senzorjev:

www.alldatasheet.com

Izobraževalni posnetki iz youtuba:

https://www.youtube.com/results?search_query=arduino+tutorial



ŠOLSKI
CENTER
CELJE

računalništvo

Šolski center celje
Srednja šola za kemijo, elektrotehniko in

Izjava



računalništvo

Šolski center celje
Srednja šola za kemijo, elektrotehniko in

Pojasnilo