ARDUINO PAMETNO AUTO

Drnasin, Luigi

Undergraduate thesis / Završni rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split / Sveučilište u Splitu**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:228:933576

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-24**

Repository / Repozitorij:

Repository of University Department of Professional Studies

zir.nsk.hr

UNIVERSITY OF SPLIT

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U SPLITU

SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Preddiplomski stručni studij, elektronika

LUIGI DRNASIN

ZAVRŠNI RAD

ARDUINO PAMETNO AUTO

Split, rujan 2021.

SVEUČILIŠTE U SPLITU
SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Preddiplomski stručni studij, elektronika

Predmet: Automatski regulacijski sustavi

Z A V R Š N I R A D

Kandidat: Luigi Drnasin

Naslov rada: Pametno auto

Mentor: Dr.sc.Barbara Džaja

Split, rujan 2021

SADRŽAJ

Sažetak 1

1. UVOD 2

2. ARHITEKTURA ARDUINO UNO MIKROKONTROLERA 3

2.1. Analogni pinovi 4

2.2. Digitalni pinovi 4

2.3. Pinovi napajanja 5

2.4. ICSP pinovi 5

2.5. Preostale komponente Arduino una 6

2.5.1. Restart tipaklo 7

2.5.2. USB tip B 8

2.5.3. USB (UART) most 8

2.5.4. Kristalni oscillator 8

2.5.5. Naponski regulator 9

2.5.6. Vanjsko napajanje 9

2.5.7. LED indikatori 9

2.5.8. ATmega328 9

3. PROGRAMIRANJE U ARDUINO PROGRAMSKOM JEZIKU 11

3.1. Logika izvršavanja Arduino koda 11

3.2. Definirane funkcije 11

3.3. Arduino biblioteke funkcija 12

3.4. Povezivanje Arduina sa računalom 13

4. POPIS I SPAJANJE POTREBNIH DIJELOVA 14

4.1.1. DC motori....14
4.1.2. L293D mikrokontroler....15
4.1.3. HC-05 Bluetooth modul....16
4.1.4. Infracrveni senzori....17
4.2. Ožićenje....19
4.2.1. Ožičenje auta upravljano Bluetooth mobilnom aplikacijom....19
4.2.2. Ožićenje auta sa infracrvenim senzorima....20
5. KODIRANJE....22
5.1. Bluetooth modul kod....22
5.2. Auto sa infracrvenim senzorima....26
6. ZAKLJUČAK....30
7. LITERATURA....31
8. POPIS SLIKA....33

Sažetak

Arduino pametno auto

U završnom radu napravljen je pametni auto upravljan Arduino mikrokontrolerom. Auto ima funkciju upravljanja preko mobilne aplikacije te su ugrađeni infracrveni senzori koji imaju funkciju praćenja crne linije na tlu. Upravljanje mobitelom vrši se preko bežićne mreže. U radu su opisane sve komponente, senzori i motori koji su korišteni pri izradi projekta. Objašnjeni su svi algoritmi i naredbe koje su nužne za funkcioniranje Arduino platforme. Opisani su svi strujni krugovi i njihova logika.

Ključne riječi: Arduino, pametno auto, mikrokontroler, IR senzori

Summary

Arduino smart car

In the thesis a smart car controlled by Arduino microcontroller was made. The car has a function of control via mobile application and built-in infrared sensors that have the function of monitoring the black line on the ground. The mobile phone is controlled via a wireless network. The paper describes all the components, sensors and motors used in the design. All algorithms and commands necessary for the functioning of the Arduino platform are explained. Circuits and their logic are described.

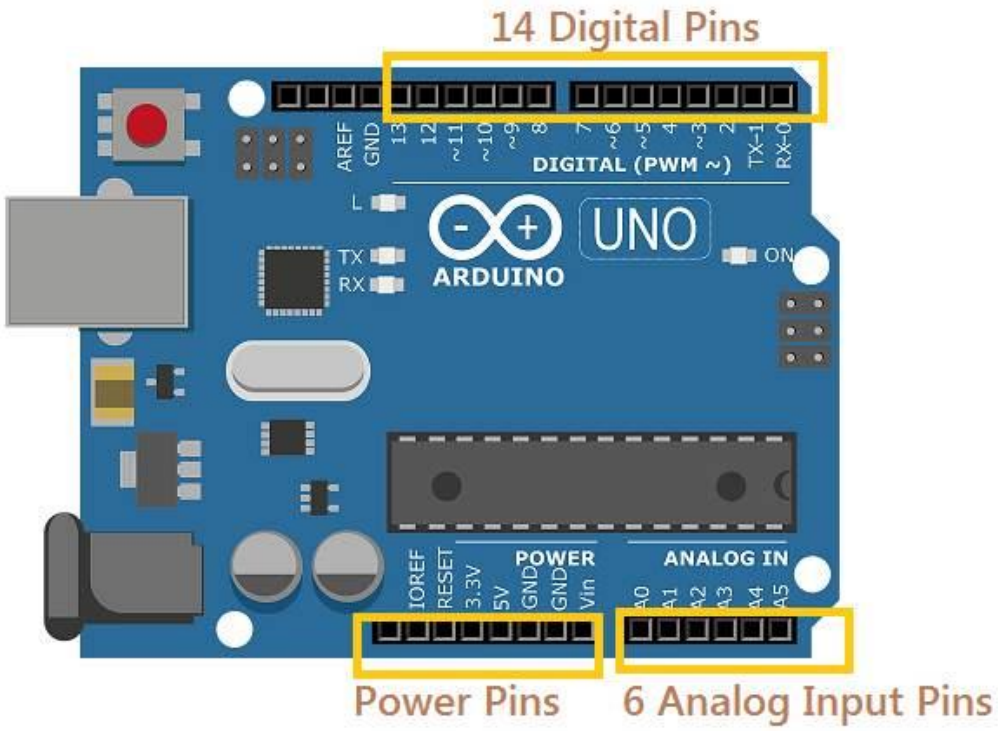
Keywords: Arduino, smart car, microcontroller, IR sensors

1. UVOD

Arduino je softverska i hardverska firma koja je osnovana 2005. godine u Italiji. Ideja osnivača firme je bila osmisliti jeftino rješenje upravljanja senzorima i robotima u rekreativne svrhe. Kroz godine upotreba Arduina brzo je rasla te je firma proširila proizvodnju pločica kao što su: Arduino uno, Arduino nano, Arduino mega, Arduino mikro i mnoge druge. Arduino očitava ulaze kao što su senzori, tipkala, poruke te ih pretvara u izlaze koji mogu pokreniti motor, upaliti ledicu i mnoge druge stvari. Kominakcija sa pločicom je ostvarena preko Arduino programskog jezika. Arduino programski jezik je malo modificirani C++ jezik. Slanjem parametara i naredbi Arduinu se nalaže što da radi. Jezik je jednostavan, svima dostupan i brzo se uči. U slučaju da korisnik nije siguran je li sve spojeno ispravno, potrebno je samo ispisati kod u Arduino simulaciji i ako ne javlja nikakvu grešku kod je ispravan. Velika prednost Arduina je što je lako dostupan i jako jeftin. Još jedna bitna stavka je što Arduino softver podržava Windows, Linux i MAC operativne sustave. Odličan je za početno učenje robotike i elektronike.

2. ARHITEKTURA ARDUINO UNO MIKROKONTROLERA

U prethodnom poglavlju iznesene su stavke o povijesti Arduina i nekim njegovim osnovnim značajkama. Ovo poglavlje obrađuje arhitekturu Arduino uno mikrokontrolera te pojašnjava njegovu svrhu i funkciju. Arduino se sastoji od dvije skupine: pinova i ostalih komponenti. Pinovi se dijele na: digitalne, analogne, napajanje i ICSP.



Slika 2.1. Arduino pinovi [1]

Pinovi prvenstveno služe za povezivanje Arduina s nekim vanjskim elementom. Svrha tog povezivanja može biti zbog napajanja ili pak komunikacije s nekim drugim uređajem. Arduino uno sadrži četiti skupine pinova, a to su:

- Analogni pinovi
- Digitalni pinovi
- Pinovi napajanja
- ICSP pinovi

Svaki od pinova ima različitu funkciju te su razlike objašnjene u sljedećim podpoglavljima.

2.1. Analogni pinovi

Analogni pinovi služe za očitavanje i pisanje analognih vrijednosti. Analogni pinovi imaju 1024 mogućih vrijednosti, odnosno 2^{10}. Analogni pinovi služe za postavljanje točno određene vrijednosti, npr. ako se želi postaviti ledica na neku specifičnu vrijednost onda se koriste analogni pinovi. Međutim, ako je potrebno postaviti neku vrijednost, npr. zvuk glasnoće na telefonu, to se ne može mjeriti sa 0 ili 1 već je potrebno ugoditi više vrijednosti koje se mogu mijenjati.
Na slici 2.1. u desnom kutu se nalaze analogni pinovi, označavaju se sa A0, A1, A2, A3, A4, A5.

2.2. Digitalni pinovi

Digitalni pinovi za razliku od analognih pinova imaju samo dvije vrijednosti, to su 0 i 1. Arduino sadrži 14 analognih pinova, sa oznakama od 0 do 13. Digitalni signal je brži i precizniji od analognog signala. Digitalni pinovi na Arduino uno pločici se mogu postaviti da budu ulaz ili izlaz. To se lako definira sa Arduino komandama: pinMode(), digitalRead() i digitalWrite(). Svaki pin u sebi ima otpornik koji se naredbom digitalWrite(HIGH/LOW) može paliti i gasiti. Pinovi 0(RX) i 1(TX) služe za primanje i prenošenje signala. Digitalni signal je otporan na zvuk i podnosi veće udaljenosti.

2.3. Pinovi napajanja

Pinovi napajanja se nalaze lijevo od analognih pinova, što je prikazano na slici 2.1. Označeni su s lijeva na desno oznakama: 3.3V, 5V, GND, GND i Vin. U nastavku je objašnjena funkcija svakog pina.

- Pin 3.3V osigurava konstani istosmjerni napon za napajanje vanjske jedinice od 3.3V, deoj je maksimalna podržana struja 40mA.

- Pin 5V osigurava konstani istosmjerni napon za napajanje vanjske jedinice od 5V, maksimalna podržana struja je 40 mA.

- Pinovi GND ili uzemljenje je točka gdje je napon 0V. GND služi za zatvaranje strujnog kruga. Bez uzemljenja strujni krug ne bi bio ispravan.

- Vin pin je ulazni napon Arduina kad koristi vanjsko napajanje. Također mu je funkcija napajanje arduina sa naponom u rasponu od 7-12V.

2.4. ICSP pinovi

Uz analogne, digitalne i pinove napajanja postoje još ICSP (engl. *In-Circuit Serial Programing*) pinovi koji služe za ICSP komunikacijsko sučelje. ICSP se sastoji od dvije skupine od po 6 pinova. ICSP pinovi služe za programiranje Arduina bez njegovog isključivanja iz strujnog kruga. Međusobno su povezani sa pripadajućim digitalnim pinovima.

Slika 2.2. ICSP pinovi [2]

Prva skupina ICSP pinova je povezana sa ATmega16U2 čipom dok je druga skupina povezana sa ATmega328, što je vidljivo na slici 2.2. ICSP skupina se sastoji od šest pinova:

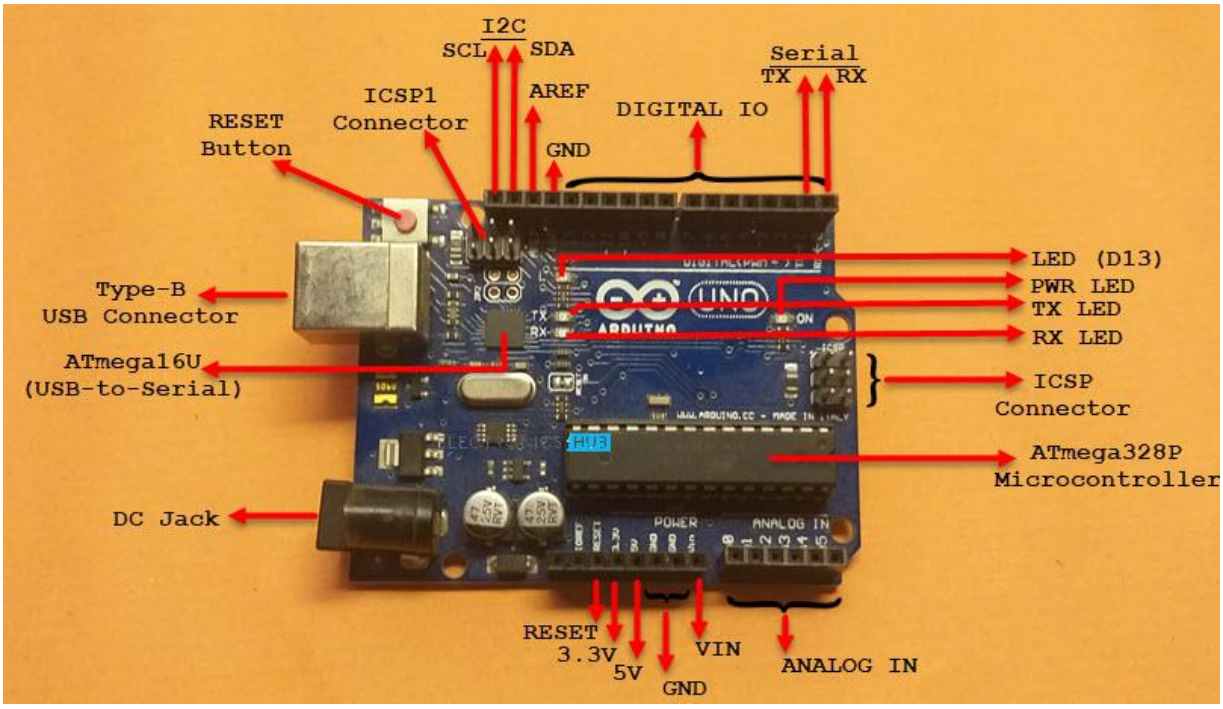
- MISO (engl. *Master In Slave Out*)
- MOSI (engl. *Master Out Slave In*)
- SCK (engl. *Serial Clock*)
- RESET (engl. *Reset*)
- VCC napajanje
- GND uzemljenje

2.5. Preostale komponente Arduino una

Osim već pojašnjenih pinova Arduino ima i drugih komponenata, slika 2.3, a one su:

- Restart tipkalo

- USB priključak
- USB (UART) most
- Kristalni oscillator
- Naponski regulator
- Priključak za vanjsko napajanje
- LED indikatori
- Mikrokontroler, ATmega328



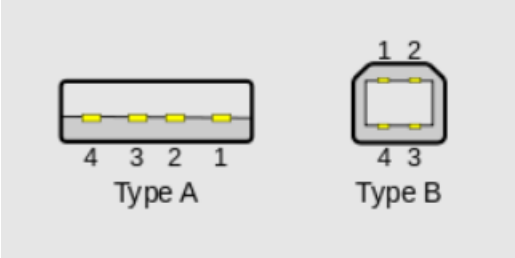
Slika 2.3. Arduino arhitektura [3]

2.5.1. Restart tipaklo

Tipkalo restart se nalazi u gornjem lijevom dijelu slike 2.3. Njegova svrha je da u kojoj god fazi izvršavanja programa Arduino bio da se ta radnja prekine i započne ispočetka, odnosno da se resetira.

2.5.2. USB tip B

Većina korisnika je upoznata sa USB (*engl. Universal serial bus*) tip A, odnosno klasičinim plosnatim priključkom. Arduino pak koristi manje poznat USB tip B. USB port je ulaz preko kojeg je ostvarena komunikacija Arduino mikrokontrolera i računala ili nekog drugog uređaja. Arhitektura USB tipa A i tipa B je vidljiva na slici 2.4. USB tip B se sastoji od četiri vodiča, dva su logično plus i minus dok su ostala dva podatkovna vodiča sa oznakama DATA+ i DATA-. DATA+ služi za slanje podataka od Arduina do računala, a DATA- za komunikaciju računalo-Arduino.



Slika 2.4. USB A i B tip [4]

2.5.3. USB (UART) most

UART (engl. *Univerasl asynchronous receiver-transmitter*) je harvderski komunikacijski protokol preko kojeg Arduino komunicira sa računalom. UART koristi samo dvije žice za odašiljanje i primanje signala.

2.5.4. Kristalni oscillator

Kristalni oscillator je izvor frekvencije od 16MHz koji se može prilagođavati shodno našoj potrebi. To je taktni signal koji konstantno titra i rad Arduina bez njega ne bi bio moguć. On daje Arduinu informaciju o vremenu.

2.5.5. Naponski regulator

Naponski regulator osigurava stabilan istosmjerni napon na Arduino pločici. Napon mora konstanto bit reguliran uz što manje oscilacija za što bolji rad mikrokontrolera.

2.5.6. Vanjsko napajanje

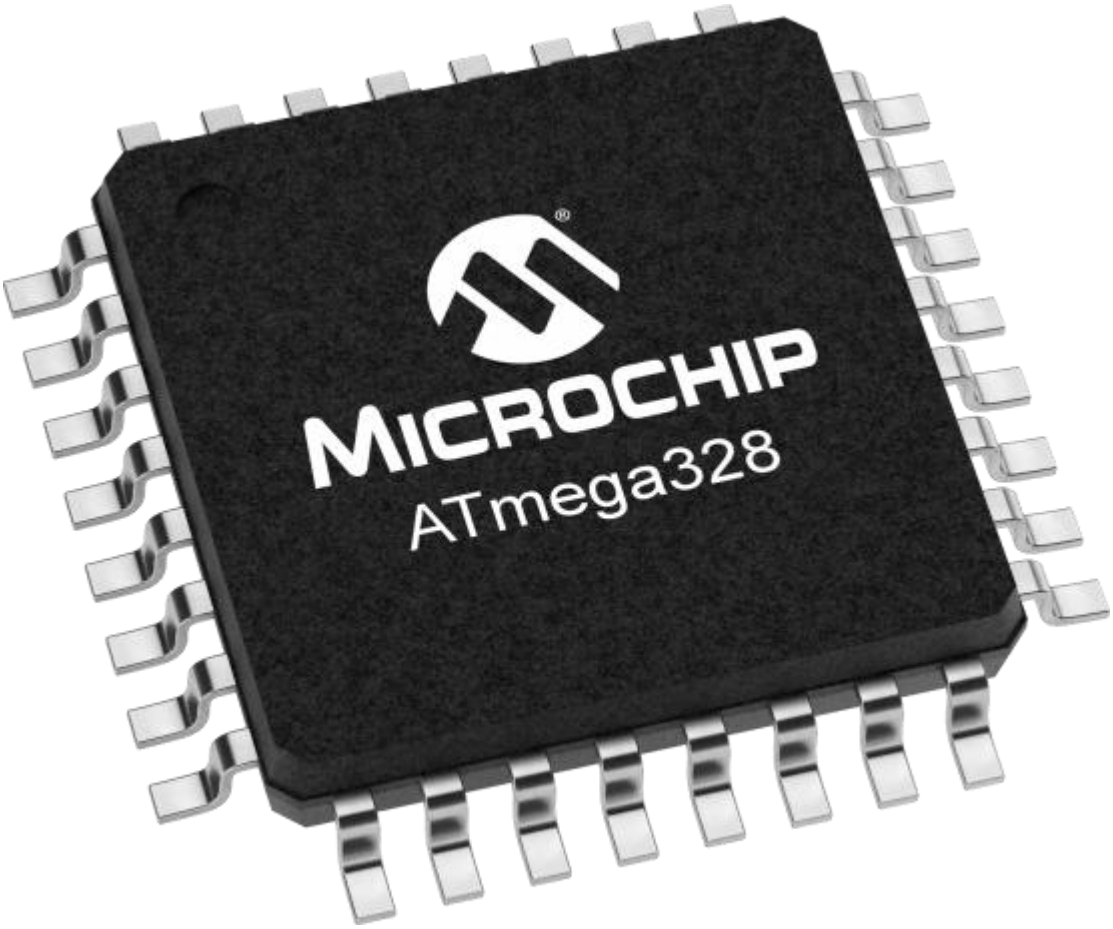
Vanjsko napajanje je ograničeno na 7-12V te je minimalna struja 250mA.

2.5.7. LED indikatori

Na Arduino pločici se nalaze četiri LED (engl. *Light Emitting Diode*) koje su označene: ON, TX, RX i L. ON ledica se pali kad je Arduino pod naponom. TX i RX ledice se aktiviraju kada se dešava obostrana razmjena podataka između Arduina i nekog uređaja. Posljednja ledica s oznakom L je spojena direktno na pin broj 13. Ona se pali kada je na pin 13 doveden neki element.

2.5.8. ATmega328

Atmega328 na slici 2.5. je 8-bitni mikročip koji je samo srce Arduino uno mikrokontrolera. Sadrži 32 KB ROM (engl. *Read-Only Memory*) memorije, 2 KB SRAM (engl. *Static Random Access Memory*) memorije i 1 KB EEPROM (engl. *Electrically Eresable Programmble Read-Only Memory*). ATmega sadrži 32 KB memorijskog prostora što znači da korisnički kod mora biti ograničen na 32 KB ili se u protivnom neće moći izvršiti. SRAM memorija služi za unos podataka dok se kod izvršava, dok EEPROM ima svrhu da zadrži memoriju ako nestane napajanja tako da se podatci ne izgube.



Slika 2.5. ATmega328 [5]

3. PROGRAMIRANJE U ARDUINO PROGRAMSKOM JEZIKU

U ovom poglavlju objašnjena je osnovna logika Arduino programiranja. Arduino programski jezik je izgrađen na C++ programskom jeziku.

3.1. Logika izvršavanja Arduino koda

Logika Arduino programiranja je malo drugačija od klasičnog C++ programiranja. Osnovna razlika je što svaki Arduino kod mora imati dvije osnovne petlje, a to su:

- void setup() {}
- void loop() {}

Setup i *loop* su dvije osnovne funckije u svakom Arduino kodu. Riječ *void* znači da ove funkcije ne vraćaju ništa. U *setup* se definiraju stavke koje se očitavaju samo jednom u kodu, kao npr. ako se žele postaviti neki parametri, neke fiksne vrijednosti, varijable ili postaviti neki pin kao ulazni ili izlazni. U funkciji *loop* se izvršava beskonačna petlja. Petlja predstavlja glavni dio programskog koda, npr. izvršavanje funkcija kao što su *for, while, if* itd.

3.2. Definirane funkcije

Arduino programiranje već ima definirane funkcije koje nije potrebno kreirati te pomažu u jednostavnijem izvršavanju koda. Neke od funkcija su:

- digitalRead() – Očitava vrijednost digitalnog ulaza
- digitalWrite() – Postavlja digitalni izlaz na HIGH ili LOW
- pinMode() – Postavlja pin mod na INPUT ili OUTPUT
- delay() – Vremenska stanka
- analogRead() – Očitava analogni napon sa analognog ulaza
- analogWrite() – Upisuje analogni napon na analogni izlaz
- Serial.print() – Ispisuje čitljive podatke na namješteni port

Slijedeći primjer prikazuje osnovni programčić u Arduinu.

```
void setup()
{
pinMode( 13, OUTPUT);     // Postavljanje digitalnog pina 13 kao izlaz
}
void loop ()
{
digitalWrite (13, HIGH); // Postavi pin 13 na high
delay (1000);             // Pauza sekundu
digitalWrite (13, LOW);  // Postavi pin 13 na low
}
```

Prikazani kodu uz već izrađene funkcije od strane Arduina izvršava blinkanje led diode. Dioda je postaviljena na pin 13. Uz pomoć *digitalWrite()* funkcije opisano je njeno stanje "HIGH" ili "LOW" odnosno, je li se pali ili gasi.

3.3. Arduino biblioteke funkcija

Arduino također već ima preodređene biblioteke funkcija odnosno skup funckija koje se mogu koristiti. Neke od njih su:

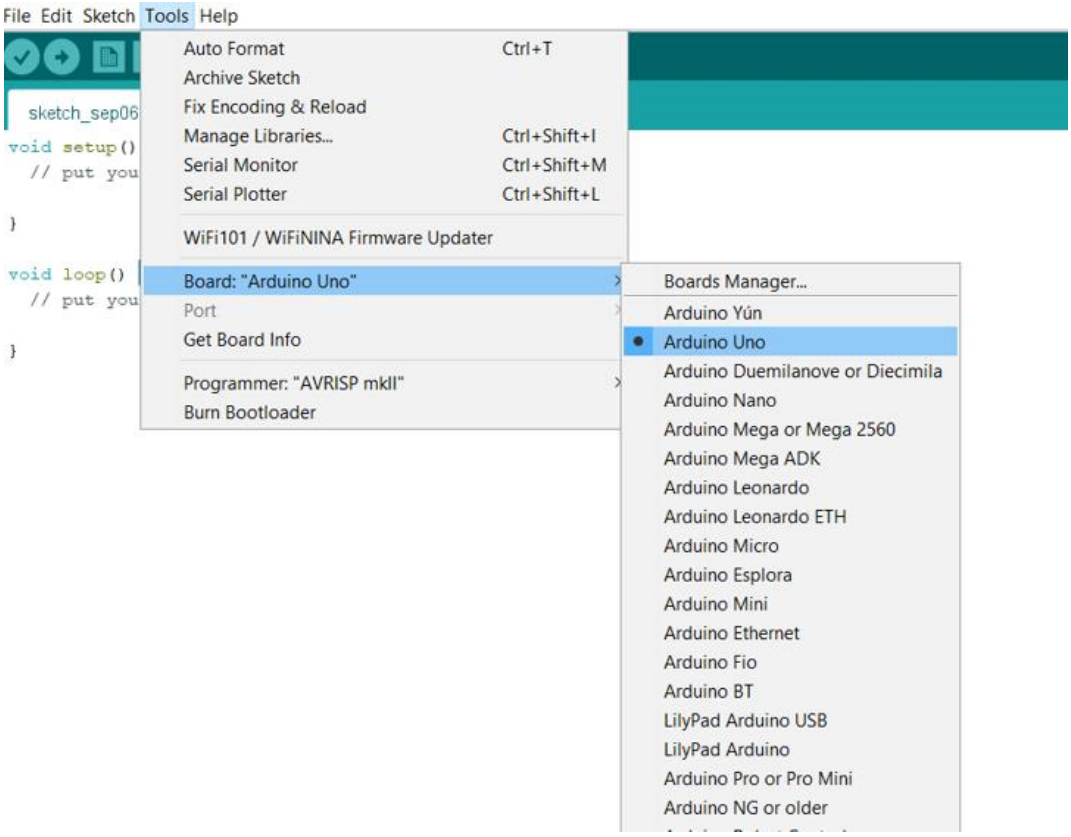
- Servo
- Motor
- Ethernet

Ako se želi pozvati servo i motor biblioteka to će se napraviti na sljedeći način. Sa "#include" pa se poziva biblioteka funkcija, njeno ime se stavlja u "< >" zagrade.

```
#include <AFMotor.h>          // Pozivanje biblioteke motor kontrolera
#include <Servo.h>           // Pozivanje biblioteke servo motora
```

3.4. Povezivanje Arduina sa računalom

Povezivanje Arduina sa računalom se vrši preko USB komunikacije. Nakon otvaranja Arduino aplikacije na računalu potrebno je spojiti se na pločicu, a to se radi odabirom: *Tools → Board: → Arduino Uno,* slika 3.1. U ovom slučaju je to Arduino Uno, ali može se izabrati bilo koji Arduino kontroler. Nakon toga bira se *Port,* što najčešće bude COM3 ili COM4. Nakon toga, pritiskom na tipkalo *Verify* provjeri se ispravnost koda te ga se nakon toga pomoću tipkala *Upload* unese na pločicu.



Slika 3.1. Povezivanje Arduina sa računalom

4. POPIS I SPAJANJE POTREBNIH DIJELOVA

Upotrijebljene komponente i detaljne tehničke karakteristike prikazane su u ovom poglavlju. U projektu su se koristile sljedeće komponente:

- DC motori
- Arduino uno
- L293D mikrokontroler
- HC-05 bluetooth modul
- Infracrveni senzori

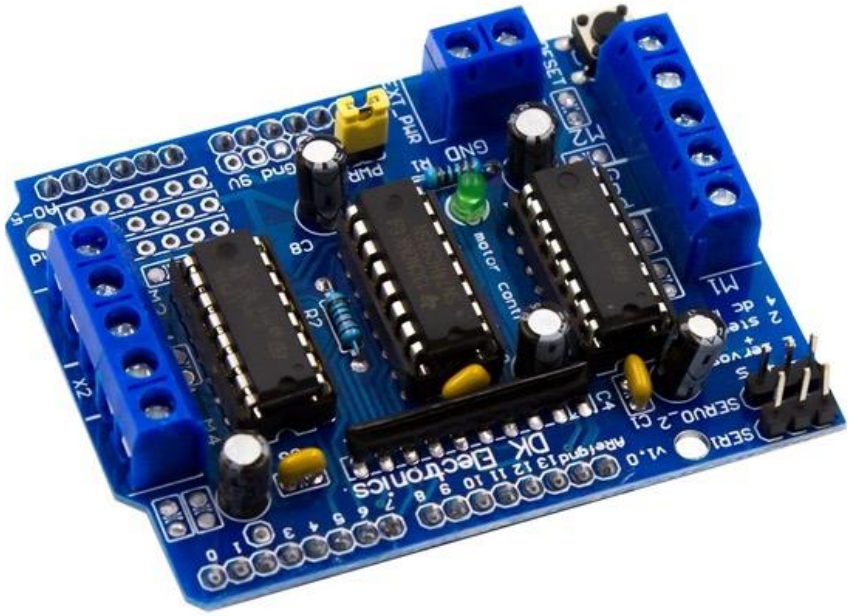
4.1.1. DC motori

Nalov projekta glasi “Pametno auto” stoga je logično da se moraju koristiti kotači, a kotače će pokretati DC motori, slika 4.1. Potrebna su četiri istosmjerna motora. DC motori ili istosmjerni motori rade na naponu od 12V, sastoje se od dvije žice: pozitivna i negativna. Ako je portebno da motor ide maksmalnom brzinom put naprijed spaja se pozitivna na 12V i negativna na GND. U slučaju da se pak zamjene žice dobit će se rotacija pri maksmilnoj brzini u suprotnom smjeru. U slučaju korištenja napona manjeg od 12V motori će se okretati nešto sporije što znači da je regulacijom napona moguće upravljati brzinom okretanja motora. Pošto upravljanje izvorom napajanja odnosno njegovo smanjenje u ovom slučaju nije praktično koristiti će se neke druge metode. Umjesto upravljanja napajanjem koristit će se PWM (engl. *Pulse Width Modulation*) signal koji će diktirati brzinu okretanja motora. PWM signal se ne može direktno spojiti na Arduino već preko L293D mikrokontrolera. Jedna funkcija mu je pojačavanje Arduino PWM signala koji radi u rasponu od 0-5V na signal od 0-12V koji je potreban za DC motore, a druga mu je funckija primanje informacija od Arduina da zamjeni polove od napajanja tako da se može dobiti okretanje motora i u suprotnom smjeru.

Slika 4.1. DC motori [6]

4.1.2. L293D mikrokontroler

Mikrokontroler L293D, slika 4.2, ili motor upravljač (engl. *motor driver*) je prijekopotreban kontroler bez kojeg rad auta ne bi bio moguć. L293D može istovremeno upravljati četiti DC motora i dva servo motora. To je zapravo H-most koji podržava 0.6A struju po mostu odnosno vršu vrijednost od 1.2A, svaki kanal podržava napon od 4.5V do 25V stoga treba pripaziti koji će se DC motor koristiti. L293D motor jednostavnom promjenom polariteta napona DC motora može promjeniti smjer njihove vrtnje. Savršeno se uklapa na Arduino te ima slobodna mjesta ako je potrebno koristiti analogne ili digitalne pinove sa Arduino uno pločice. Na pločicu će se lemiti dodatni pinovi za spajanje Bluetooth modula i infracrvenih senzora.

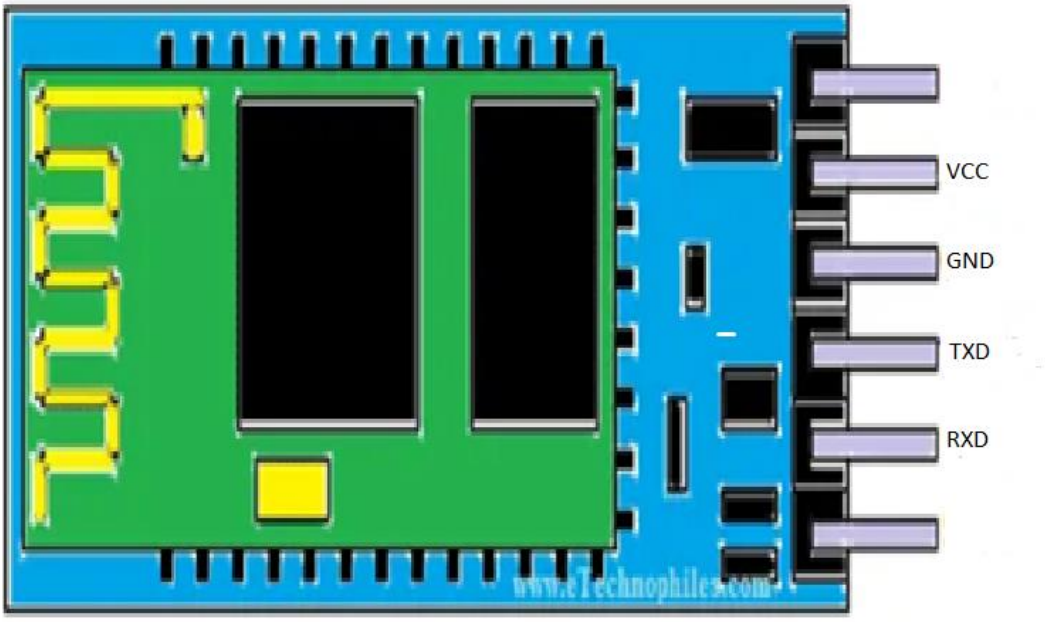


Slika 4.2. L293D mikrokontroler [7]

4.1.3. HC-05 Bluetooth modul

Upravljanje auta je ostvareno kroz bežićnu komunikaciju sa mobilnim telefonom stoga je potreban element koji to omogućuje, a to je HC-05 bluetooth modul. Preko njega Arduino prima informacije, odnosno on omogućava da signal sa mobilnog telefona stigne do Arduino pločice. HC-05 ima šest pinova slika 4.3, dok se u ovom radu koristite četiri, a to su: VCC, GND, TXD, RXD.

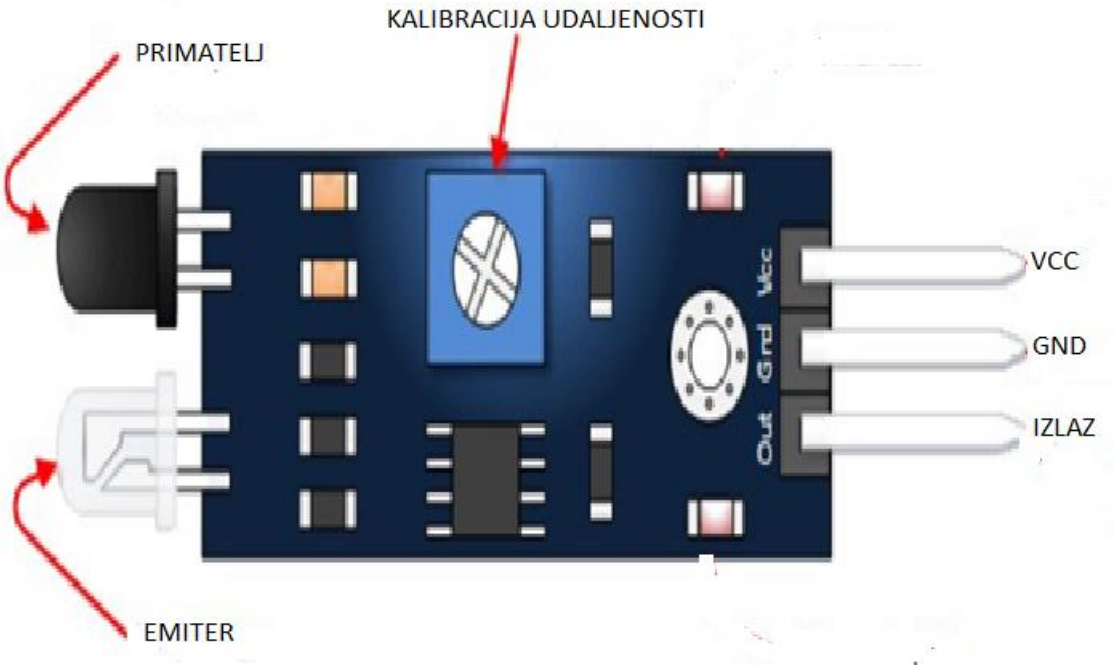
- VCC je napajanje modula, modul radi na naponu od 3.3V do 5V.
- GND je uzemljenje odnosno masa.
- TXD je pin koji prenosi signal od mobilnog telefona i prenosi ga RX pin na Arduino pločici.
- RXD je pin koji prima naredbu od TX pina sa Arduina i prenosi ga bežično.



Slika 4.3. HC-05

4.1.4. Infracrveni senzori

Infracrveni senzor radi na principu emitranja i detektiranja infracrvenih zraka. Radijacija infracrvenih zraka se nalazi u elektromagnetskom spektru koji nije vidljiv ljudskom oku. Arhitektura senzora je vidljiva na slici 4.4.

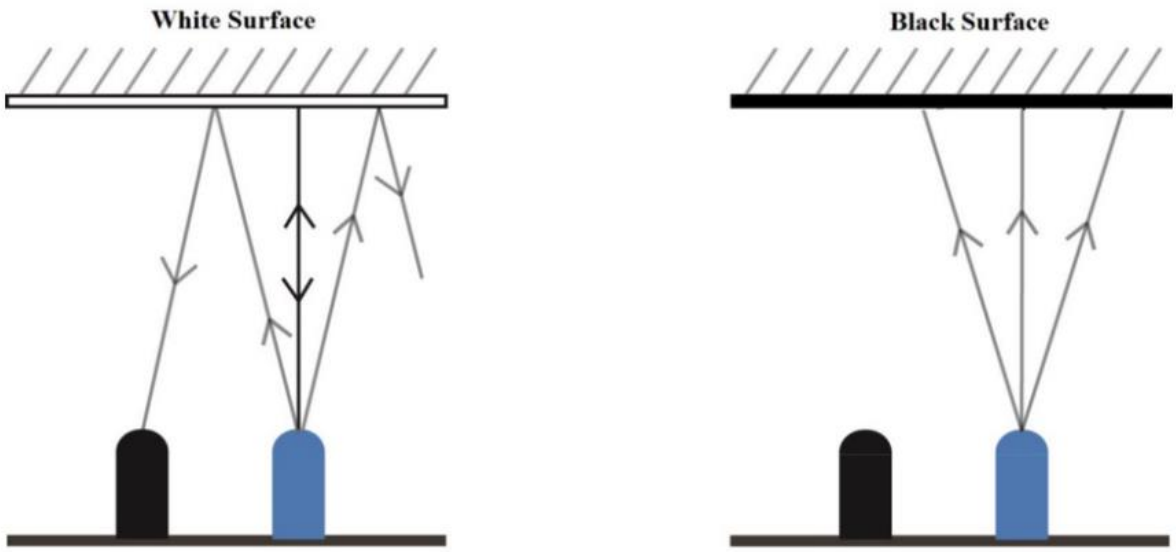


Slika 4.4. Infracrveni senzor

Osnovni djelovi senzora su:

- Emiter (engl. *Emitter*) koji emitira infrared zrake
- Primatelj (engl. *Receiver*) je foto dioda sa otpornikom koja prima odbijene zrake
- Prilagođivač udaljenosti (engl. *Distance adjuster*) je dio s kojim se kalibrira udaljenost na kojoj se želi da senzor radi
- VCC, 3-5V
- GND
- Izlaz, koji šalje dobiveni signal na Arduino pločicu
- Ledica koja se pali kada se signal pojavi na primatelju
- Ledica napajanja koja se pali kad je senzor pod naponom

Senzor radi na principu da kada primatelj dobije odbijeni signal šalje 0 ili LOW kao informaciju na Arduino, ako pak primatelj nije primio nikakav signal to jest nije bilo površine od kojeg se odbio ili je površina upila infracrvene valove tada šalje signal 1 ili HIGH. Što znači ako je površina crna odnosno upila je zrake nećemo dobit nikakvu informaciju na primatelju slika 4.5.



Slika 4.5. Senzor na bijeloj i crnoj površini [8]

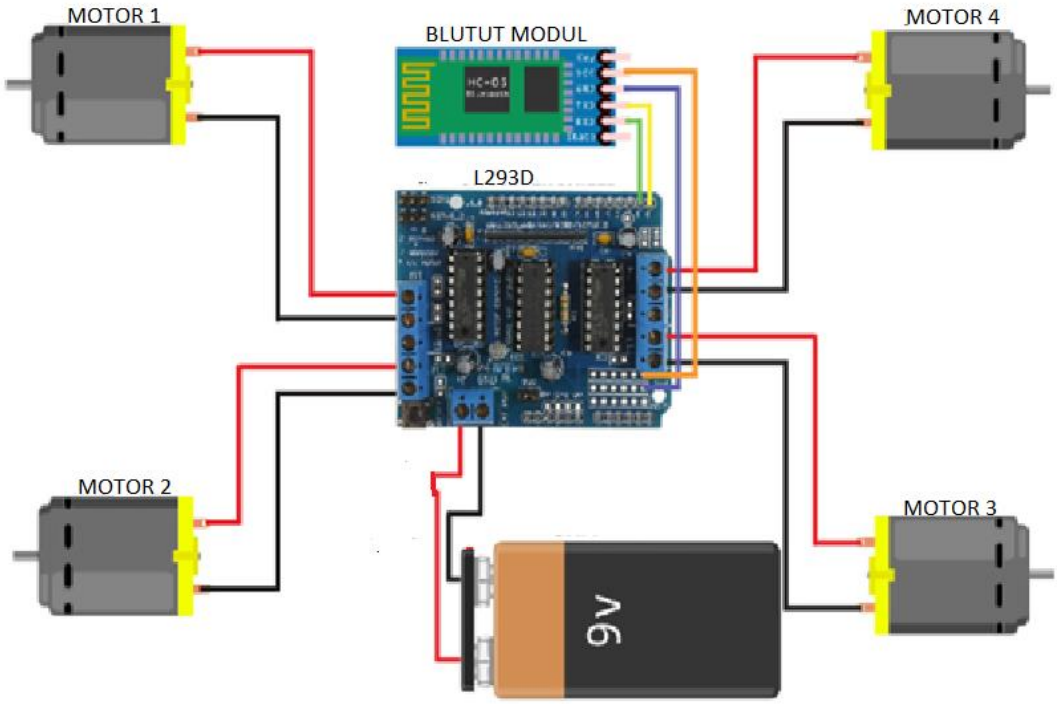
4.2. Ožićenje

Jako bitna stavka da bi projekt bio ispravan je da se točno napravi ožićenje. U ovom podpoglavlju prikazano je povezivanje svih elemenata koji su se koristili u sklapanju projekta.

4.2.1. Ožičenje auta upravljano Bluetooth mobilnom aplikacijom

Kao što je vidljivo na slici 5.1. motori su spojeni na L293D kontroler, svaki motor sa svojim plusom i minusom na za to predviđeno mjesto. Baterija napaja kontroler i ubačen je prekidač.

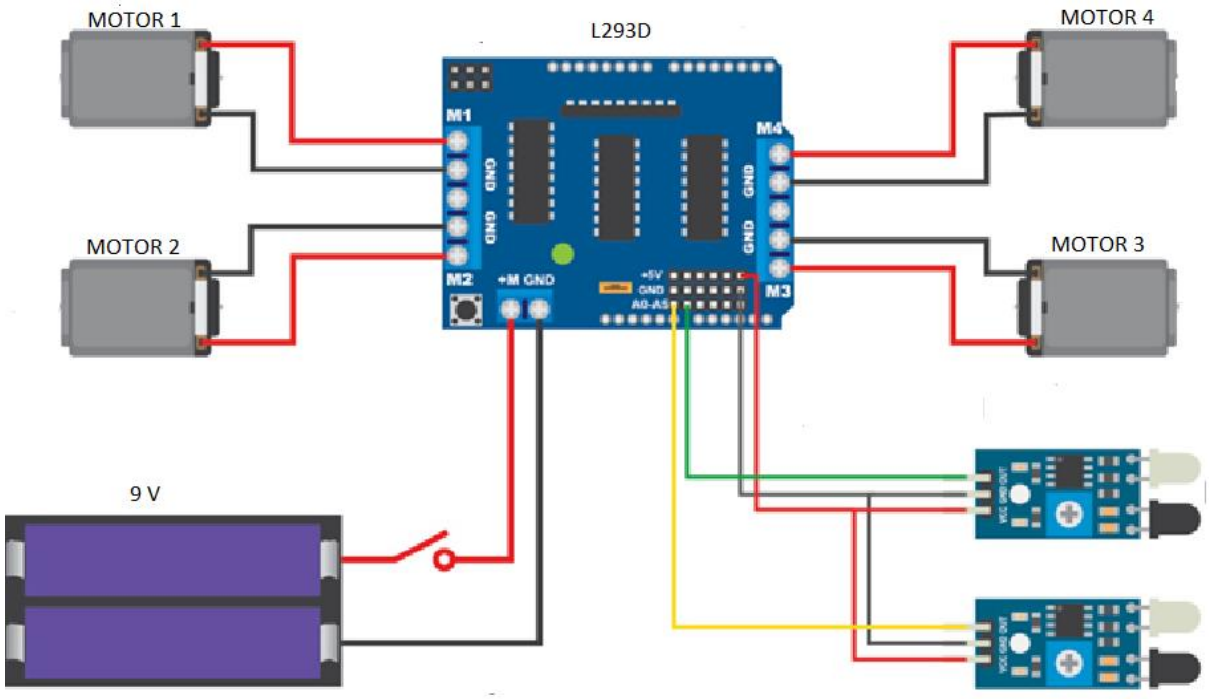
Bluetooth modul je spojen na napajanje gdje su prethodno zalemljeni pinovi. Dok su pinovi TXD I RXD spojeni na RX i TX.



Slika 4.6. Ožićenje bluetooth modula

4.2.2. Ožićenje auta sa infracrvenim senzorima

Razlika ožićenja je u tome da više se ne koristi Bluetooth modul već se koriste dva infrared senzora. Senzori se spajaju na analogne pinove koji su prethodno zalemljeni. Prvi senzor se spaja na A0, a drugi na A1. Napajanje senzora se uzima sa iste točke gdje se i Bluetooth napajao, što se vidi na slici 4.7.



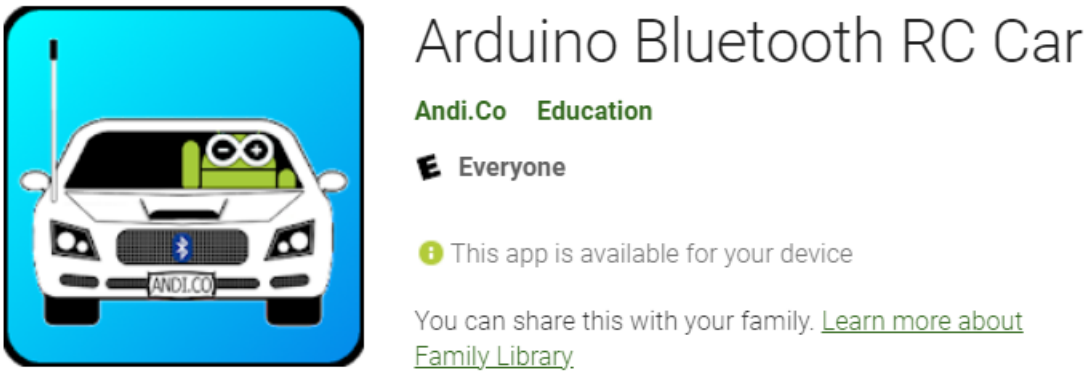
Slika 4.7. Ožićenje infra crvenog modula

5. KODIRANJE

Ovo poglavlje obrađuje kod koji je korišten za oba slučaja.

5.1. Bluetooth modul kod

U projektu auto preko Bluetootha komunicira sa mobilnom aplikacijom i prati njene upute. Koristila se aplikacija "Bluetooth RC Controller", slika 5.1. Aplikacija ima više definiranih funkcija, u projektu se koristilo pet osnovnih funkcija za kretanje auta, a to su: naprijed, nazad, lijevo, desno i stop.



Slika 5.1. Arduino aplikacija [9]

Komande auta koje će se koristit u kodu:

- Naprijed – F
- Nazad – B
- Lijevo – L
- Desno – R
- Zaustavljanje – S

```
#include <AFMotor.h>           // Uključivanje motor biblioteke

AF_DCMotor motor1(1);          // Definiranje motora po pinovima
AF_DCMotor motor2(2);
AF_DCMotor motor3(3);
AF_DCMotor motor4(4);

int tipkalo;                   // Kreiranje varijable "tipkalo"

void setup()
{
  Serial.begin(9600);
}

void loop(){
  if(Serial.available() > 0){ // Izvrši samo kad primi naredbu
    tipkalo = Serial.read();  // Očitaj naredbu
    Stop();                   // Poziv funkcije za zaustavljanje
    if(tipkalo == 'F')        // Ako je Serial.read = F, ajde naprijed
    {
      forward();              // Poziv funckije naprijed
    }
    else if(tipkalo == 'B')  // Ako je idi nazad
    {
      back();                 // Poziv funckije nazad
    }
    else if(tipkalo == 'R')  // Idi desno
    {
      right();                // Poziv funkcije desno
    }
    else if(tipkalo == 'L')  // Idi lijevo
    {
      left();                 // Poziv funkcije lijevo
```

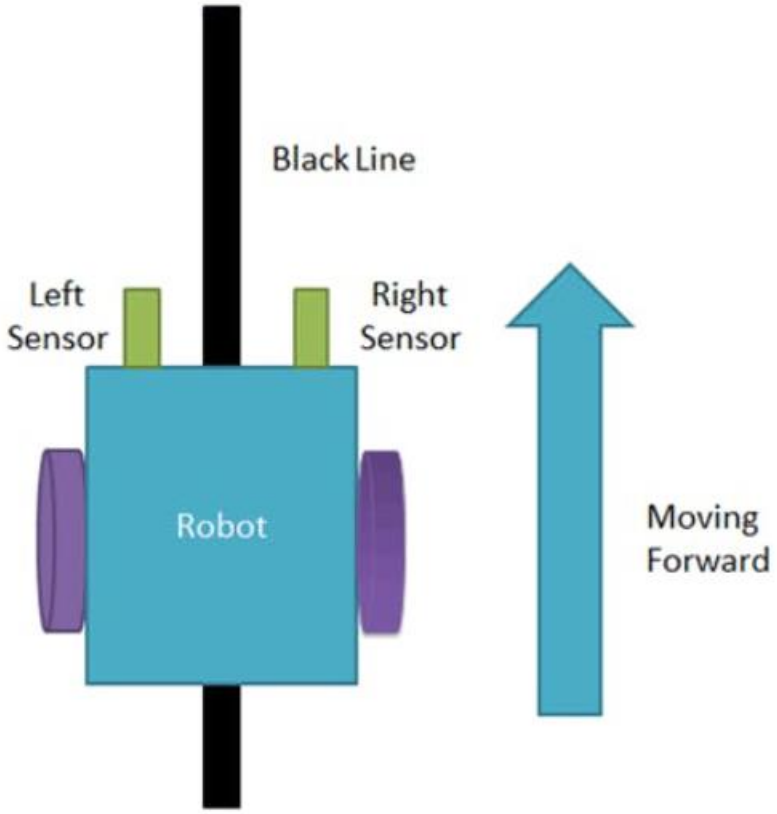
```
    }
  }
}
void forward()                // Funkcija za naprijed
{
  motor1.setSpeed(255);      // Maksimalna brzina svih motora naprijed
  motor1.run(FORWARD);
  motor2.setSpeed(255);
  motor2.run(FORWARD);
  motor3.setSpeed(255);
  motor3.run(FORWARD);
  motor4.setSpeed(255);
  motor4.run(FORWARD);
}
void back()                  // Funkcija za nazad
{
  motor1.setSpeed(255);     // Maksimalna brzina svih motora nazad
  motor1.run(BACKWARD);
  motor2.setSpeed(255);
  motor2.run(BACKWARD);
  motor3.setSpeed(255);
  motor3.run(BACKWARD);
  motor4.setSpeed(255);
  motor4.run(BACKWARD);
}
void left()                  // Funkcija lijevo
{
  motor1.setSpeed(255);    // Lijevi motori nazad, a desni naprijed
  motor1.run(BACKWARD);
  motor2.setSpeed(255);
  motor2.run(BACKWARD);
  motor3.setSpeed(255);
  motor3.run(FORWARD);
  motor4.setSpeed(255);
```

```
  motor4.run(FORWARD);
}

void right()              // Funkcija desno
{
  motor1.setSpeed(255);  // Lijevi motori naprijed, a desni nazad
  motor1.run(FORWARD);
  motor2.setSpeed(255);
  motor2.run(FORWARD);
  motor3.setSpeed(255);
  motor3.run(BACKWARD);
  motor4.setSpeed(255);
  motor4.run(BACKWARD);
}

void Stop()               // Funckija zaustavljanja
{
  motor1.setSpeed(0);    // Zaustavljanje svih motora
  motor1.run(RELEASE);
  motor2.setSpeed(0);
  motor2.run(RELEASE);
  motor3.setSpeed(0);
  motor3.run(RELEASE);
  motor4.setSpeed(0);
  motor4.run(RELEASE);
}
```

Na početku pisanja koda moralo se uključiti biblioteku za L293D kontroler. Nakon toga su se definirali pinovi na DC motorima. Glavni dio koda u *loop* petlji koji se stalno ponavlja je izvšen preko *if* petlje, gdje se svaka funkcija kretanja izjednačila sa pripadujićim tipkalom. Izradile su se funkcije za kretanje u svim smjerovima i za zaustavljanje koje su se pozivale u *if* petlji po potrebi.

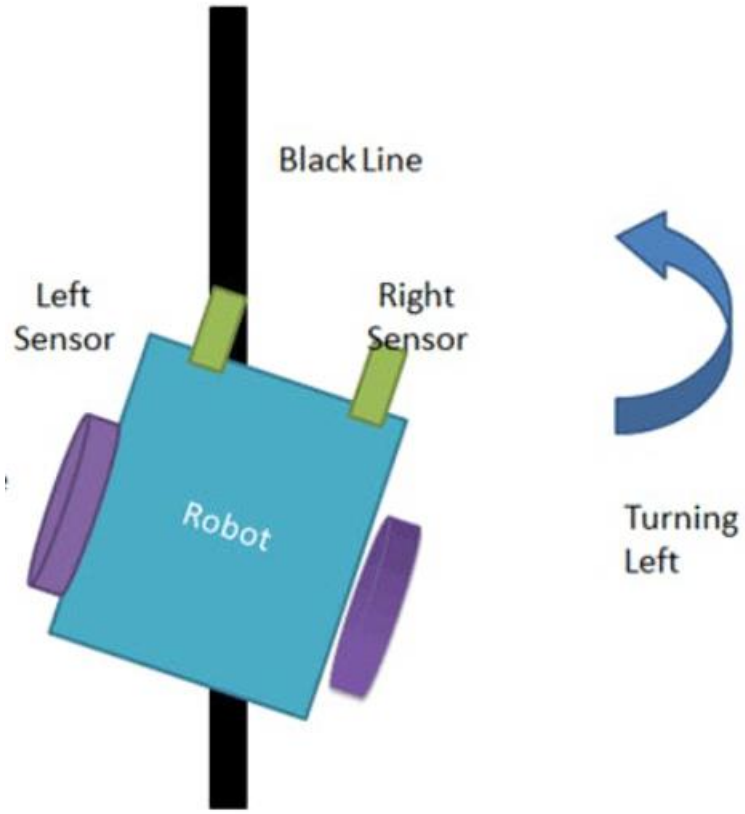
5.2. Auto sa infracrvenim senzorima

Osnovna logika auta je da prati crnu liniju po podu. Senzor radi na principu emitiranja i očitanja. Slijedom toga, ako se na primatelju pojavi signal senzor šalje informaciju 0 na Arduino, a ako se ne pojavi signal u našem slučaju ako ga crna linija absorbira onda šalje signal 1. Ako je crna linija točno po sredini auta to jest oba senzora su dobila povratnu informaciju auto će se kretati naprijed, ako crna linija promjeni smjer desno ili lijevo auto će se pokušati vratiti na idealnu poziciju gdje oba senzora dobivaju povratnu informaciju. Auto staje ako se crna linija postavi okomito na senzore. Na slici 5.2. prikazana je logika kretanja auta kad ide ravno.



Slika 5.2. Kretanje auta naprijed [10]

Na slici 5.3. je auto koje ide lijevo odnosno lijevi senzor je na crnoj liniji.



Slika 5.3. Kretanje auta lijevo [11]

Potrebno je još kalibrirati senzore preko već spomenutog prilagođivača udaljenosti.

```
#include<AFMotor.h>          // Uključivanje biblioteke
AF_DCMotor motor1(1);        //Motori po pinovima
AF_DCMotor motor2(2);
AF_DCMotor motor3(3);
AF_DCMotor motor4(4);

#define LEFT_SENSOR A0        // Lijevi senzor na A0 analogni pin
#define RIGHT_SENSOR A1       // Desni senzor na A1 analogni pin
int distance = 50;
void setup() {

pinMode(RIGHT_SENSOR, INPUT); // Postavljanje senzora na ulaze
pinMode(LEFT_SENSOR, INPUT);  //
```

```
}

void loop() {
if(analogRead(RIGHT_SENSOR)<=distance                                    &&
analogRead(LEFT_SENSOR)<=distance) /* Provjera da li oba senzora nisu
na liniji, if se izvršava samo ako su oba uvjeta zadovoljena*/
{
  motor1.run(FORWARD);   // Svi motori naprijed sa pola brzine
  motor1.setSpeed(120);
  motor2.run(FORWARD);
  motor2.setSpeed(120);
  motor3.run(FORWARD);
  motor3.setSpeed(120);
  motor4.run(FORWARD);
  motor4.setSpeed(120);
}
 else            if(!analogRead(RIGHT_SENSOR)<=distance            &&
analogRead(LEFT_SENSOR)<=distance) /* Ako je desni na  liniji, a lijevi
ne */
{
  motor1.run(FORWARD);  // Kretanje desno
  motor1.setSpeed(255);
  motor2.run(FORWARD);
  motor2.setSpeed(255);
  motor3.run(BACKWARD);
  motor3.setSpeed(255);
  motor4.run(BACKWARD);
  motor4.setSpeed(255);
}
 else             if(analogRead(RIGHT_SENSOR)<=distance             &&
!analogRead(LEFT_SENSOR)<=distance) /* Ako je lijevi na liniji, a desni
ne */
{
  motor1.run(BACKWARD); // Kretanje lijevo
  motor1.setSpeed(255);
  motor2.run(BACKWARD);
  motor2.setSpeed(255);
  motor3.run(FORWARD);
  motor3.setSpeed(255);
  motor4.run(FORWARD);
  motor4.setSpeed(255);
}
 else            if(!analogRead(RIGHT_SENSOR)<=distance            &&
!analogRead(LEFT_SENSOR)<=distance) // Ako su oba na liniji
{

  motor1.run(RELEASE); // Zaustavi motore
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
 }
```

```
}
```

Na početku koda uključuju se potrebne biblioteke i definira se motore po pinovima. Prije izvršavanja programa postavlja se konstantna vrijednost A0 analognog pina na lijevi senzor te A5 za desni senzor. Funkcijom *analogRead* očitava se vrijednost lijevog i desnog senzora te se pomoću *if* petlje testira uvjete gdje se nalazi auto. Dakle ako nam lijevi senzor očitava crnu liniju, a desni ne očitava auto će se okreniti lijevo, isto vrijedi u obrnutom slučaju. Zaustavljanje je u slučaju ako oba senzora očitaju crnu liniju, a kretanje u naprijed ako senzori nemaju očitanje.

6. ZAKLJUČAK

U ovom radu napravljen je pametni auto koji demonstrira osnovnu logiku Arduino mikrokontrolera. Infracrveni mod i Bluetooth mod su realizirani te rade neometano. Rad na ovom projektu je otkrio mnoge probleme koji su rješeni samostalnim istraživanjem i konstantnim učenjem. Iako je ovaj projekt moguće poboljšati u svim segmentima, predstavlja jedan razvojni proces koji je započeo idejom, razvijao se istraživanjem, završio primjenom stečenog znanja te rezultirao neprocjenjivim iskustvom. Kako u svakom projektu postoji mogućnost poboljšanja i unaprijeđenja, tako postoji u ovom.

7. LITERATURA

[1] Chaurasia, N. *Arduino Analog Input read from Pin with the help of analogRead function from Potentiometer*, 6, 2019, https://elextutorial.com/learn-arduino/arduino-analogread-analog-input-pin-read-potentiometer/ [6.9.2021]

[2] Apnorton, *Unlabeled Headers/Pins on Arduino Uno*, 4, 2013, https://electronics.stackexchange.com/questions/67108/unlabeled-headers-pins-on-arduino-uno [6.9.2021]

[3] Ravi, T. *Arduino UNO Pinout, Specifications, Board Layout, Pin Description*, 1, 2021, https://www.electronicshub.org/arduino-uno-pinout/ [3.9.2021]

[4] Theydiffer, *DIFFERENCE BETWEEN USB TYPE-A AND USB TYPE-B*, 1, 2016, https://theydiffer.com/difference-between-usb-type-a-and-usb-type-b/ [1.9.2021]

[5] Microchip ATmega48A/PA/88A/PA/168A/PA/328/P, megaAVR Data Sheet.

[6] HandsOn Tech, Dual Shaft 200RPM@6V Mini DC Gear Motor

[7] TEXAS INSTRUMENTS, L293x Quadruple Half-H Drivers

[8] Manzoor, M. *Arduino with Infrared Sensor*, 8, 2020, https://medium.com/illumination/arduino-with-infrared-sensor-48ad4415f320 [2.9.2021]

[9] Andi.Co, Education, Google Play, Arduino Bluetooth RC Car, https://play.google.com/store/apps/details?id=braulio.calle.bluetoothRCcontroller&hl=en [1.9.2021]

[10] RoboCircuits, PROJECT HUB, *Line Follower Robot Arduino © CC BY-NC-SA*, 5, 2018, https://create.arduino.cc/projecthub/robocircuits/line-follower-robot-arduino-299bae [3.9.2021]

[11] RoboCircuits, PROJECT HUB, *Line Follower Robot Arduino © CC BY-NC-SA*, 5, 2018, https://create.arduino.cc/projecthub/robocircuits/line-follower-robot-arduino-299bae [3.9.2021]

8. POPIS SLIKA

Slika 2.1. Arduino pinovi [1] 3
Slika 2.2. ICSP pinovi [2] 6
Slika 2.3. Arduino arhitektura [3] 7
Slika 2.4. USB A i B tip [4] 8
Slika 2.5. ATmega328 [5] 10
Slika 3.1. Povezivanje Arduina sa računalom 13
Slika 4.1. DC motori [6] 15
Slika 4.2. L293D mikrokontroler [7] 16
Slika 4.3. HC-05 17
Slika 4.4. Infracrveni senzor 18
Slika 4.5. Senzor na bijeloj i crnoj površini [8] 19
Slika 4.6. Ožićenje bluetooth modula 20
Slika 4.7. Ožićenje infra crvenog modula 21
Slika 5.1. Arduino aplikacija [9] 22
Slika 5.2. Kretanje auta naprijed [10] 26
Slika 5.3. Kretanje auta lijevo [11] 27