

ARDUINO AUTOMOBIL

Šmit, Ena

Master's thesis / Specijalistički diplomski stručni

2018

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

Permanent link / Trajna poveznica: <https://urn.nsk.hr/um:nbn:hr:125:540996>

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

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



Repository / Repozitorij:

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

VELEUČIŠTE U RIJECI

Ena Šmit

ARDUINO AUTOMOBIL

(specijalistički završni rad)

Rijeka, 2018.

VELEUČI LIŠTE U RIJECI

Poslovni odjel

Specijalistički diplomski stručni studij

Informacijske tehnologije u poslovni sustavi m

Ena Šmit

ARDUI NO AUTOMOBIL

(specijalistički završni rad)

MENTOR

STUDENT

M. sc. Vesna Krajčí, predavač

Ena Šmit

MBS: 2422000129/ 16

Rijeka, listopad 2018.

VELEUČILIŠTE U RIJECI

Prilog 1.

Poslovni odjel

Rijeka, 01.10.2018.

**ZADATAK
za specijalistički završni rad**

Pristupnik: Ena Šmit

MBS: 2422000129/16

Studentici specijalističkog diplomskog stručnog studija Informacijske tehnologije u poslovnim sustavima izdaje se zadatak specijalističkog završnog rada – tema specijalističkog završnog rada pod nazivom:

Arduino automobil

Sadržaj zadatka: Izraditi mobilni robot u obliku manjeg automobila na daljinsko upravljanje pomoću Arduino platforme i bluetooth tehnologije. Objasniti Arduino platformu, bluetooth tehnologiju, upotrijebljene motore i senzore te ostale komponente potrebne za izradu automobila. Također objasniti način sastavljanja navedenih dijelova kako bi se dobio gotov automobil. Povezati automobil preko Arduino pločice s mobitelom i objasniti programski dio za njihovu bluetooth komunikaciju. Izraditi i objasniti vlastitu Android aplikaciju za daljinsko upravljanje automobilom pomoću mobitela.

Preporuka: Prilikom izrade mobilnog robota u obliku automobila što je moguće više koristiti lako dostupne i jeftine komponente.

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

Zadano: 01.10.2018.

Predati do: 15.12.2018.

Mentorica:
mr. sc. Vesna Krajčić, predavač

Vesna Krajčić
(potpis mentorice)

v. d. pročelnika Poslovnog odjela:
mr. sc. Anita Stilić, viši predavač

Anita Stilić
(potpis pročelnice)

Zadatak primljen dana: 01.10.2018.

Ena Šmit

Ena Šmit
(potpis studentice)

Dostavlja se:
- mentorici
- pristupnici

IZJAVA

Izjavljujem da sam specijalistički završni rad pod naslovom _____
KREDUINO AUTOMOBIL izradio samostalno pod
nadzorom i uz stručnu pomoć mentora mr. sc. VESNA KRAJCIĆ.

Ime i prezime



(potpis studenta)

Sažetak

Ovaj rad opisuje postupak izrade automobila na daljinsko upravljanje koristeći Arduinovo platfotmu i *bluetooth* tehnologiju. Rad definira sve pojmove potrebne za detaljno razumijevanje postupka izrade odnosno objasnjava tehnologiju i dijelove korištene prilikom izrade projekta. Također su objašnjeni postupci fizičkog sastavljanja auta, funkcije mobilne aplikacije te ostvarenje veze između mobilne aplikacije i Arduinovo platfotme. Rad prikazuje mogućnosti praktičnog projekata te završava opisom njegove implementacije.

Ključne riječi: Arduino, bluetooth, senzor, Android

Sadržaj:

1.	Uvod	1
2.	Upotrijebljena tehnologija i dijelovi	2
2.1.	Postupak sastavljanja automobil-a	8
2.2.	Troškovnik	10
3.	Arduinovo IDE	11
3.1.	Set up() funkcija	12
3.2.	Loop() funkcija	14
3.3.	Autonomna vožnja	18
4.	Mobilna aplikacija	21
4.1.	Designer dio	23
4.2.	Blocks dio	23
4.3.	Dizajniranje aplikacije	25
4.4.	Konfiguracija funkcionalnosti aplikacije	30
4.5.	Provjera i pokretanje aplikacije	34
5.	Zaključak	40
	Literatura	41
	Popis kratica	42
	Popis slika	43
	Popis tablica	44

1. Uvod

Tehnologija daljinskog upravljanja postoji već značajan niz godina i nije nepoznana čak i u svim uzrastima. Djeca se od malena susreću s navedenom tehnologijom korištenjem igračaka poput auta na daljinsko upravljanje dok se odrasli, između ostalog, sa tehnologijom daljinskog upravljanja mogu susresti korištenjem dronova i robota. Svrha ovog projekta je približiti tehnologiju daljinskog upravljanja korištenjem tehnologije koja omogućuje visok stupanj fleksibilnosti i personalizacije. Korištenjem Arduina omogućuje se komunikacija auta sa mobilnim uređajem odnosno mobilnom aplikacijom korištenjem *Bluetooth* veze.

Cilj ovog završnog rada je izrada automobila putem Arduino razvojne platforme i kontrole poomoći *Bluetooth*-a i ultrazvučnog senzora, uz primenjući Android aplikaciju. Aлат за izradu aplikacije je **MIT App Inventor**.

Rad je podijeljen u 5 dijelova, od kojeg drugo poglavje rada pobliže opisuje tehnologiju i ključne dijelove potrebne za shvaćanje ovog rada. Utreće poglavljajuće objašnjene Arduino IDE programi kod u njemu napisan. Učetvrtom poglavljajuće opisana mobilna aplikacija.

2. Upotrijebljena tehnologija i dijelovi

Radi boljeg razumijevanja projekta, nužno je objasniti korištenu tehnologiju i sastavni elemente u sustavu. Za izradu projekta potrebno je razumjeti sljedeće:

- Arduin
- Arduino UNO R3
- Arduino IDE
- Bluetooth
- SG90 sljednički motor
- L298N upravljač motora
- Motor bez četvrtice
- V5 shield (senzor shield)
- Ultrazvučni sensor
- Modul zvučnog signala
- RGB LED modul

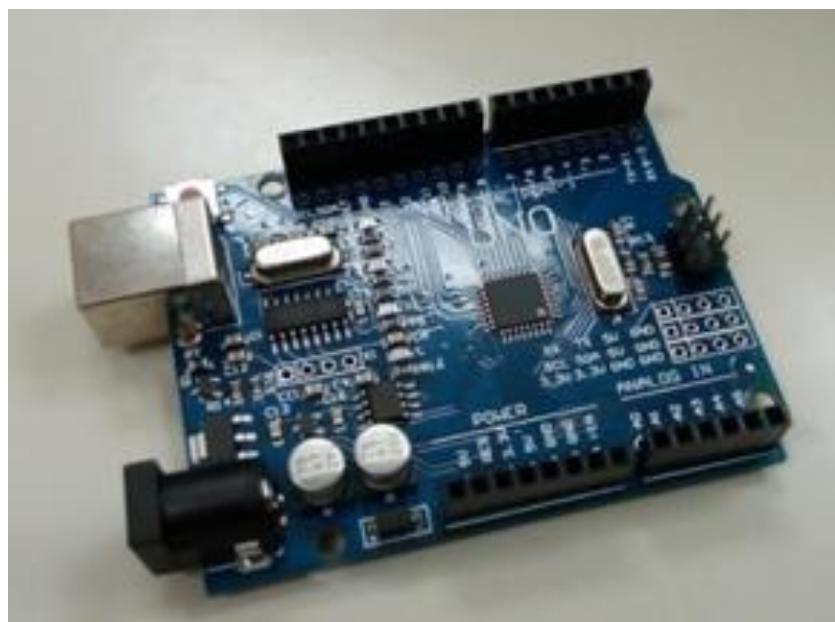
Arduino je mikrokontrolerska platforma bazirana na jednostavnom i opovljujućem programu te je samim time vrlo popularna i raširena platforma. Ovoren programski paket iz kojeg se Arduino programira baziran je na Processing programskom jeziku i vrlo je jednostavan za učenje. Jezik se može proširiti kroz C++ knjižnice. Arduino programski paket je jednostavan za korištenje za početnike, ali dovoljno fleksibilan za napredne korisnike. Radi na Mac OS X, Windows i Linux operacijskim sustavima.

Arduino Uno je ploča za mikrokontroler temeljena na ATmega328P. Imala je 14 digitalnih ulazno/izlaznih pinova (igitala) od kojih se 6 može koristiti kao PWM izlaz, 6 analognih ulaza, 16 MHz kvarcnog kristala, USB priključak, utičnicu za napajanje, I2C header i gumb za resetiranje. Sadrži sve potrebno da bi mogao podržavati mikrokontroler. Može se jednostavno spojiti s računalom preko USB kabla i puniti preko AC-DC adaptera ili preko baterija. Uno ploča je prva u nizu Arduino ploča i referentni model za Arduino platformu.

Arduino Uno R3 smatra se idealnom alternativom za one koji nisu stolični elektronikom i kodiranjem. Najčešće je korištena/dokumentirana ploča Arduino obitelji - što dovodi do iznimno aktivne zajednice koja može pomoci pri rješavanju potencijalnih problema.

Također, za potrebe završnog rada koristio se **Ardui no 1.8.7(IDE)**. To je program motorenog koda koji omogućava jednostavno pisanje koda i njegovo slanje na ploču. Pogodan je za Windows, Mac OS X i Linux operacijske sustava. Arduino(IDE) se može koristiti sa svakom Arduino pločom

Slika 1 - Prikaz Arduino Uno R3 ploče



Izvor: Autorica

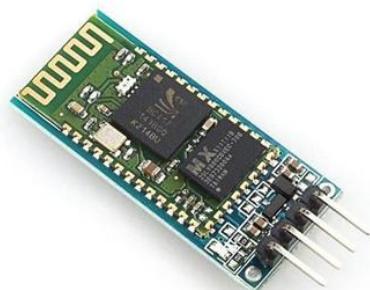
Slika 2 - Specifikacije Arduino Uno R3 ploče

Microcontroller	ATmega328P
Operating voltage	5V
USB	14
Digital I/O Pins	6
Analog Input Pins	6
Flash Memory	32 KB
SRAM	2 K
EEPROM	1 K
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Izvor: store-usa.arduino.cc (01.10.2018.)

Bluetooth je način bežične razmjene podataka između dva ili više uređaja. Veći na uređaji poput mobitela, računala i kamera i maj u mogućnost slanja podataka pomoću bluetooth-a. Veza se uspostavlja putem radio valova u frekvencijskom području od 2,4 do 2,48 GHz. Uređaji koji se povezuju vezu mogu ostvariti u promjeru od otprilike 10 metara oko uređaja, a zbog korištenja radio veze uređaji ne moraju biti u optičkoj vidljivosti niti međusobno usmjereni. Osnovna i načina bluetootha omogućava prijenos podataka do 1 Mbit/s. Bluetooth tehnologija je korištena kako bi se ostvarila komunikacija između automobilova i mobilnog uređaja/aplikacija.

Slika 3 - Prikaz bežičnog bluetootha pri naprednjima



Izvor: Autorica

Za potrebe ovog završnog rada korišteni su i **SG90 slijedni motori**. Masa jednog motora iznosi samo 9 gram. Može se rotirati za približno 180 stupnjeva - 90 stupnjeva u oba smjera.

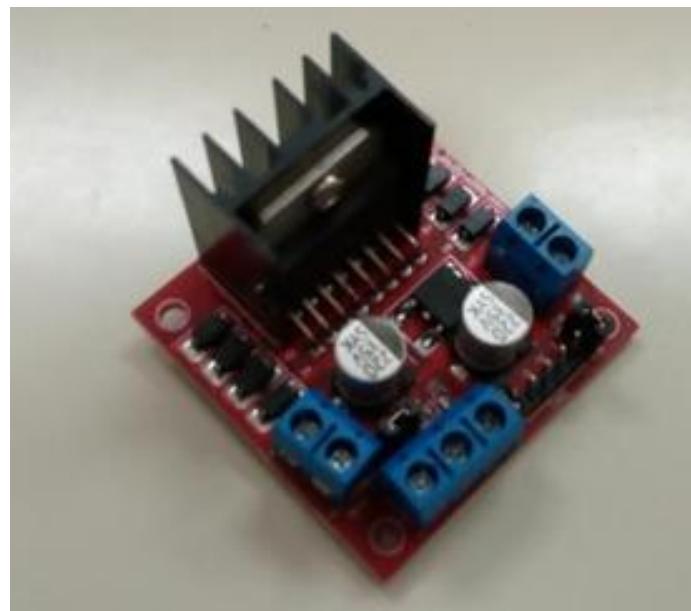
Slika 4 - Prikaz SG90 slijednog motora



Izvor: www.mikropik.com(16.08.2018.)

L298N Dual H Bridge Motor Controller je korišten za upravljanje brzim i s mjerom okretanja osovine motora.

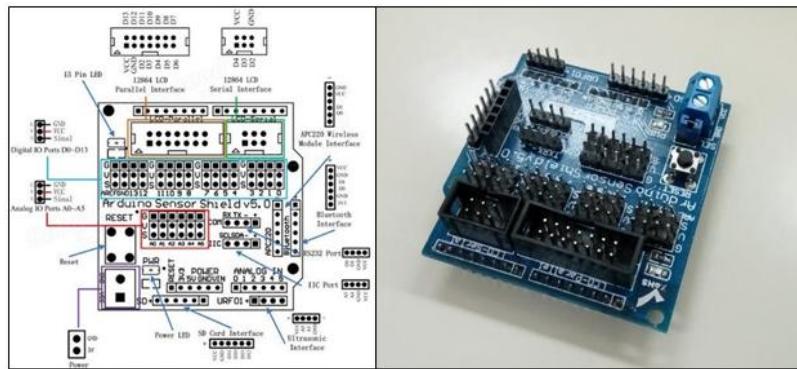
Slika 5 - Prikaz L298N Dual H Bridge Motor Controllera



Izvor: Autorica

Arduinovo Sensor Shield V 5.0 senzorsko proširenje ploče je nastavak na matičnu ploču Arduino Uno. Priključuje se na vrh Arduino ploče i omogućava povezivanje sa svim digitalnim ulazima i izlazima i analognim mulazima. Verzija 5.0 ima mogućnost odvojiti opskrbu od 5 V na sve 3-pinske konektore i napajati ih iz zasebnog vanjskog napajanja.

Slika 6 - Funkcijski dijagram te prikaz korištenog VS štita



Izvor: www.banggood.com(16.08.2018.) i autorka

Ultrazvučni senzor je uređaj koji mjeri udaljenost objekta pomoći zvučnih valova. Mjeri udaljenost pomoći slanja zvučnog vala određene frekvencije i prati vraćanje istog zvučnog vala. Na temelju vremena potrebnog da se emitirani zvuk vrati, moguće je izračunati udaljenost između senzora i objekta.

Slika 7 - Prikaz korištenog ultrazvučnog senzora



Izvor: Autorka

Motor bez četkica je sinhronizirani električni motor. Ovi motori su jeftini, mali, jednostavni za ugradnju i idealni su za upotrebu u mobilnom robotskom automobilu.

Modul zvučnog signala je pasivni jer ne ima sustav koji proizvodi sam zvuk. Nužno je imati mikrokontroler, poput Arduina, koji proizvodi zvuk.

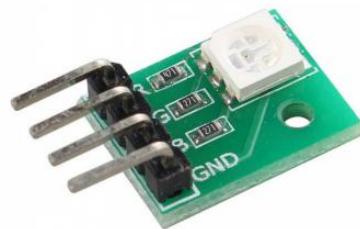
Slika 8 - Prikaz modula zvučnog signala



Izvor: www.arduino.cc (04.09.2018.)

RGB LED senzor je svjetlosni senzor koji ima otpornike koji ograničavaju struju kako bi spriječili izgaranje.

Slika 9 - Prikaz svjetlosnog senzora



Izvor: www.ebay.com (04.09.2018.)

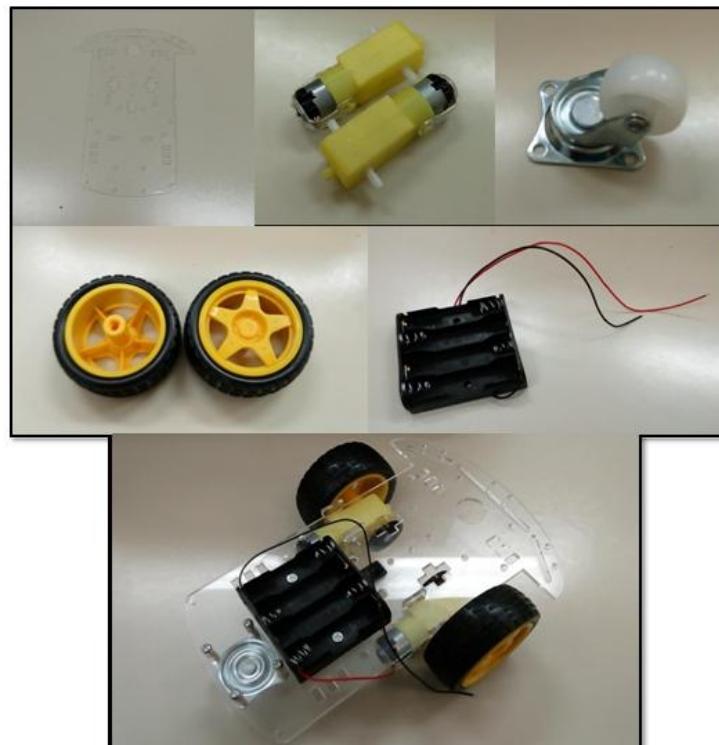
Što se **baterija** tiče, automobil za pokretanje i uporabu zahtjeva četiri AA baterije.

2.1 Postupak sastavljanja automobila

Prije korak je sastaviti primarni dio ove automobile, odnosno dio ove koji će automobilu dati prepoznatljiv oblik koji će omogućiti daljnje postavljanje željenih komponenata.

U primarni dio spada plastična ploča koja je nosilac svih ostalih dijelova, motor bez četkica, kotači, malen kotač za rotaciju te kućište za baterije.

Slika 10 - Priček pri marnih dijelova automobila

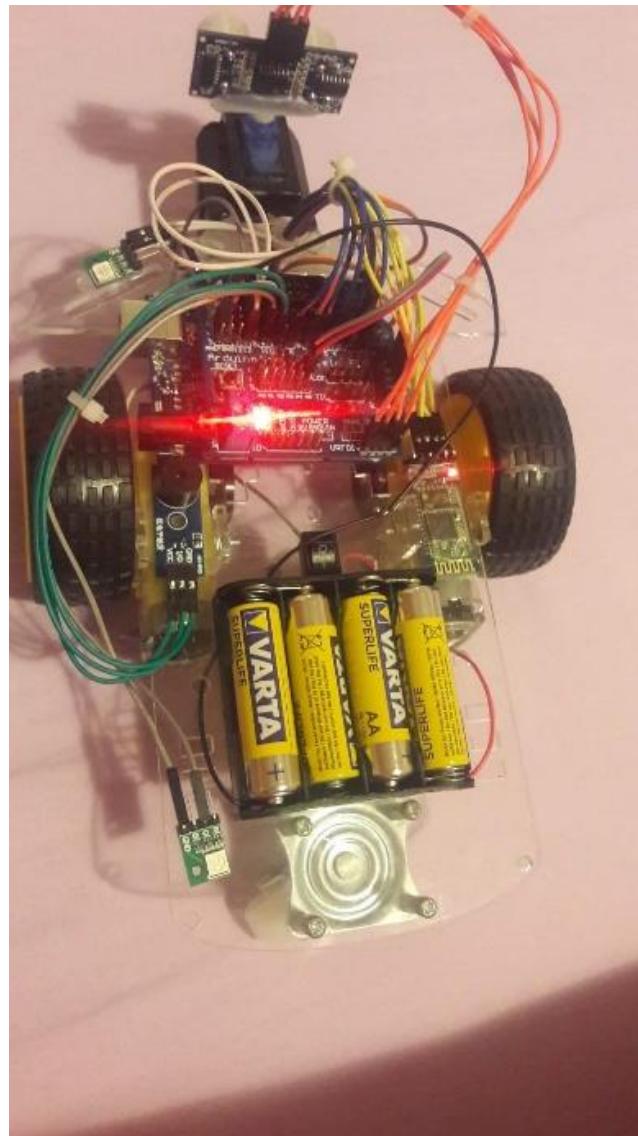


Izvor: Autorica

Pri marni dijelovi se sastavljaju pomoću vijaka i matice. Nakon što se pri marni dijelovi sastave, automobil je spreman za stavljanje ostalih dijelova poput ultrazvučnog senzora, V5 shielda, Arduino Uno ploče, L298N Dual H Bridge Motor Controller-a i SG90 slijednog motora. Ostatak elemenata se žičano poveza i lijepi pomoću pištolja s ljepilom.

Na i dućoj slici može se vidjeti izgled sastavljenog Arduino automobila.

Slika 11 - Prikaz sastavljenog Arduino automobila



Izvor: [Aut orica](#)

2.2 Troškovni k

Prilikom izrade završnog rada cilj je bio koristiti jeftine i lako dostupne komponente. Sve komponente su naručene putem Bay-a.

U tablici 1 je prikazan okvirni troškovnik za izradu Arduinovog automobila.

Tablica 1 - Troškovnik

Naziv	Gjena (kn)
Prijenični dijelovi (plexiglas pliča, dva kotaca, dva motora bez četkića, gumb za uključivanje/isključivanje, kotač za rotaciju, kućište za baterije, matice, spojnice)	45,00
Arduino Uno R3	25,00
Arduino Sensor Shield V5.0	25,00
L298N Dual H Bridge Motor Controller	12,00
SG90 slijedni motor	12,00
Ultrazvučni senzor	7,00
Bežični Bluetooth predajnik	22,00
Žlžjumper kabel set 40 kom veze	40,00
Modul zvučnog signala	7,00
RGB LED senzor (2 kom)	14,00
Ukupno	209,00

3. Arduino IDE

Arduinovo IDE je program otvorenog koda koji služi za pisanje koda i slanje na Arduinovo ploču. Basiran je u programskom jeziku Java, a bazira se na Processingu. Možuće ga je pokrenuti na Windows, Mac OS ili Linux operacijskim sustavima.

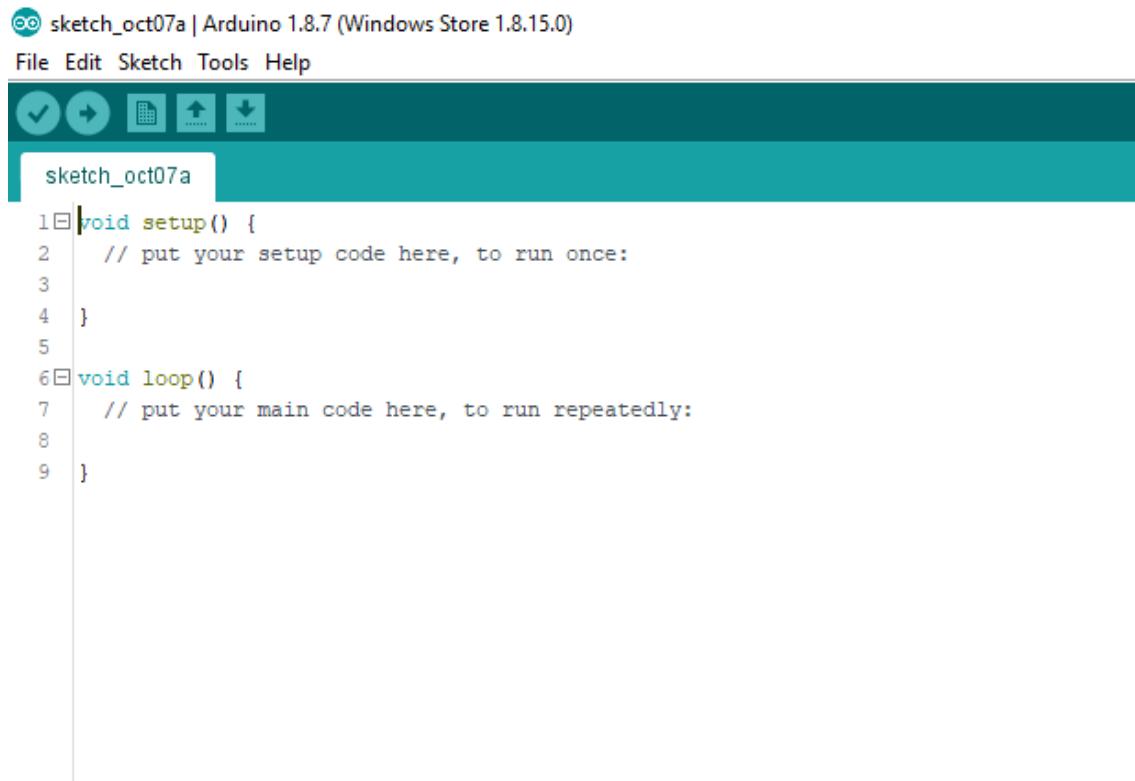
Mnogi programi za Arduinovo imaju sljedeću strukturu:

setup() - izvodi se na početku kada se Arduinovo upali ili kada se resetira

loop() - beskonačna petlja koja je u stvari glavni program Arduinova

Proces slanja koda na ploču je dosta jednostavan. Potrebno je povezati ploču sa računalom preko USB kabla, podešiti opcije serijskog porta u IDE-u te odaberati opciju *upload* u alatnoj traci. IDE će, prije nego se program pošalje na ploču, verificirati kod i prevesti ga za odgovarajuću arhitekturu mikrokontrolera. U slučaju da verifikacija ne prođe, program neće biti poslan na ploču.

Slika 12 - Arduinovo IDE



The screenshot shows the Arduino IDE interface. The title bar reads "sketch_oct07a | Arduino 1.8.7 (Windows Store 1.8.15.0)". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for upload, download, and other functions. The main area displays the following code:

```
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Izvor: Autorica

U daljnjem tekstu biti će objašnjen kod napisan u Arduino IDE programu koji nije definiran rad Arduino automobila.

Kako je ranije spomenuto, `setup()` i `loop()` su dvije glavne funkcije Arduino. Funkcija `setup()` se pozove kada se Arduino upali, a funkcija `loop()` je zapravo glavna funkcionalnost programa i poziva se dokle god je Arduino upaljen. Dakle, nakon što je funkcija pozvana i Arduino je izvrši, Arduino je ponovo pozove.

3.1 Set up() funkcija

U funkciji `setup()` inicijalizirane su varijable, pionovi i komunikacija sa *bluetoothom*.

Za početak, prvo je potrebno namjestiti varijable:

```
timePassed = 0;  
  
selfDrivingMode = false;  
  
hornDuration = 0;  
  
oldTime = 0;  
  
drivingPhase = GO_FORWARD;  
  
drivingTimer = 100;  
  
sensorTimer = 0;  
  
frontLights = false;  
  
backLights = false;
```

Zatim se omogućava veza sa *bluetoothom* način da dokle god se šalju podaci, ti podaci se i čitaju. Kada se *bluetooth* uključi, on šalje neki tekst (podatke) te je potrebno pročistiti međuspremnik (*buffer*) - dakle, što god se posalje, automatski se izbriše.

```
Serial.begin(9600);  
  
while( Serial.available() ) Serial.read();
```

Na kraju su postavljeni pinovi za motor, trubu, prednje i zadnje svjetlo, ultrazvučni senzor itd. Kod uključivanja automobila, upalit će se prednje i zadnje svjetlo na 100 milisekundi.

```
// Postavljanje izlaza motora
pinMode( motorLeft_Activate, OUTPUT );
pinMode( motorLeft_Forward, OUTPUT );
pinMode( motorLeft_Back, OUTPUT );
pinMode( motorRight_Activate, OUTPUT );
pinMode( motorRight_Forward, OUTPUT );
pinMode( motorRight_Back, OUTPUT );

// Postavljanje izlaza ultrazvučnog senzora
pinMode( trigger, OUTPUT );
pinMode( echo, INPUT );

// Postavljanje pina za servo motor ultrazvučnog senzora
servoMotor.attach(9);

// Kratki zvučni signal s trubom
analogWrite( light_front, 255 );
analogWrite( light_back, 255 );
delay(100);

// Gašenje trube
analogWrite( light_front, 0 );
analogWrite( light_back, 0 );
}
```

3.2 Loop() funkcija

U loop() funkciji se izračunava koliko je vremena prošlo od prethodnog poziva funkcije loop() što je potrebno zbog *timera*, zato mčitanje da li je korisnik posao naredbu preko *Bluetootha* (funkcija ReadRemoteInput()), uključi vanje i automatsko isključi vanje trube po potrebi te autonomna vožnja.

```
void loop() {
    unsigned long timeNow = millis();

    if( oldTime == 0 ) oldTime = timeNow;

    timePassed = timeNow - oldTime;

    if( timePassed > 250 ) timePassed = 250;

    oldTime = timeNow;

    ReadRemoteInput();
    // Uključi/isključi trubu, po potrebi
    if( hornDuration > 0 ) {
        analogWrite( horn, 0 );      // Truba uključena

        hornDuration = hornDuration - (int)timePassed;
    } else {
        analogWrite( horn, 255 );   // Truba isključena
    }
}
```

Kod izračuna koliko je vremena prošlo od prethodnog poziva funkcije, funkcija millis() vraća koliko je milisekundi prošlo od kada je Arduino uključen. To možda korisniku nije toliko bitno, ali trebal bi spriječati neke moguće greške u radu (*bugove*).

To će kasnije biti potrebno za *timere* kada se izračunava *timer* za trubu i za *self-driving* algoritam koliko je vremena prošlo u kojoj fazi.

Dakle, izračunat je *timer* i zatim se čita ReadRemoteInput() koji pročita ako i ma što je poslano preko *Bluetootha*, pročita što je poslano i onda prema naredbi to izvršava.

U slučaju da automobil radi u načinu autonomne vožnje, on će se prekinuti ukoliko korisnik pritisne neki gumb s mjerom (naprijed, natrag, lijevo ili desno).

```
if( Serial.available() <= 0 ) return;

char data;

data = Serial.read();

switch( data ) {

case 'F': // Naprijed
    StopSelfDriving();
    go_Forward();
    break;

case 'B': // Natrag
    StopSelfDriving();
    go_Back();
    break;

case 'L': // Lijevo
    StopSelfDriving();
    go_Left();
    break;

case 'R': // Desno
    StopSelfDriving();
    go_Right();
    break;

case '?': // Zaustavi auto
```

```

go_Stop();

break;

case 'V': // Zvučni signal 250 ms
hornDuration = HORN_DURATION;
Serial.write('H');

break;

case 'W': // Prednje svijetlo paljenje/gašenje
if( frontLights ) {
frontLights = false;
analogWrite( light_front, 0 );
Serial.write('w');
} else {
frontLights = true;
analogWrite( light_front, 255 );
Serial.write('W');
}

break;

case 'U': // zadnje svijetlo paljenje/gašenje
if( backLights ) {
backLights = false;
analogWrite( light_back, 0 );
Serial.write('u');
} else
backLights = true;
analogWrite( light_back, 255 );
Serial.write('U');

break;

```

```
case 'S': // Autonomna vožnja uključena/isključena
if( selfDrivingMode ) {
    StopSelfDriving();
} else {
    StartSelfDriving(); }
break; }
```

3.3 Autonomna vožnja

Autonomna vožnja je način rada automobila u kojem nije vožnje nije kontroliran od strane korisnika, nego ultrazvučnim senzorom koji šalje zvučni val određene frekvencije i prati vraćanje istog te na temelju potrebnog vremena da se emitirani zvuk vrati, može izračunati udaljenost između senzora i objekta (prepreke).

```
int UdaljenostDoPrepreke() {  
  
    int trajanje, udaljenost;  
  
    digitalWrite( trigger, HIGH );  
  
    delayMicroseconds(1000);  
  
    digitalWrite( trigger, LOW );  
  
    trajanje = pulseIn( echo, HIGH );  
  
    udaljenost = (trajanje/2) / 29.1;  
  
    return udaljenost; }
```

Algortam za autonomnu vožnju može biti u 3 stanja, odnosno način na ponašanja:

- GO_FORWARD - vozi na naprijed
- BACK_UP - vozi na natrag
- TURN_RIGHT - okreni se desno

U početnoj fazi se automobil kreće prema naprijed i prvo što će napraviti je izračunavanje udaljenost do prepreke. Ako je udaljenost manja od definirane, prebacuje se u BACK_UP fazu i kratko se pali prednje svjetlo kao signal da je automobil prepoznao prepreku. Ta faza traje 500 milisekundi, automobil se zaustavlja i prebacuje se u sljedeću fazu TURN_RIGHT koja traje 360 milisekundi. Kad ta faza završi, automobil se zaustavlja i ponovo kreće prema naprijed.

```

void SelfDriveFaza_GoForward() {
    // Provjeri da li je prepreka blizu nas
    int udaljenost = UdaljenostDoPrepreke();

    if( udaljenost < 15 ) {
        drivingPhase = BACK_UP;      // Pocni vracanje na nazad
        drivingTimer = 500;          // Idi nazad za 500ms
        go_Stop();
        go_Forward();

        // Upali prednja svjetla na kratko, prepoznata prepreku
        ispred
        analogWrite( light_front, 128 );
        delay(50);

        analogWrite( light_front, 0 );
    } else {
        go_Back();
    }
}

void SelfDriveFaza_BackUp() {
    // Smanji timer od trenutne faze automatske voznje
    drivingTimer = drivingTimer - timePassed;

    if( drivingTimer < 0 ) {    // Zavrsili smo ici na nazad,
        pocni okretati na desno
        go_Stop();

        drivingPhase = TURN_RIGHT; // Pocni okretanje na desno
        drivingTimer = 360;
        go_Left();
    }
}

```

```

void SelfDriveFaza_TurnRight() {
    drivingTimer = drivingTimer - timePassed;

    if( drivingTimer < 0 ) {
        go_Stop();
        drivingPhase = GO_FORWARD;
        go_Back(); // Reverse in self-driving mode
    }
}

void SelfDrivingAlgorithm() {
    sensorTimer = sensorTimer + timePassed;

    float percent = sensorTimer /
        (float)SENSOR_ROTATION_DURATION;

    float k = sin( percent * 2 * PI ) * 0.5;

    int sensorAngle = 90 + (int)(SENSOR_ROTATION_ANGLE * k);
    // Vrijednost izmedju 20 i 160

    // Okreni servo od senzora udaljenosti na novi kut
    servoMotor.write( sensorAngle );

    if( drivingPhase == GO_FORWARD ) {
        SelfDriveFaza_GoForward();
    } else if( drivingPhase == BACK_UP ) {
        SelfDriveFaza_BackUp();
    } else if( drivingPhase == TURN_RIGHT ) {
        SelfDriveFaza_TurnRight(); } }

```

4. Mobilna aplikacija

Osnovna funkcija završnog rada je upravljanje automobilom preko aplikacije, odnosno upravljanje po moću tipki ili prebacivanje na automobil vožnju u kojoj se koristi ultrazvučni senzor za izbjegavanje prepreka.

Za kreiranje mobilne aplikacije korišteni su MT App Inventor 2 i Photoshop kao po moć kod dizajniranja aplikacije.

MT App Inventor 2 je besplatna web aplikacija otvorenog koda koja korisnicima omogućuje stvaranje programskih aplikacija za Android operacijski sustav. App Inventor zbog svoje jednostavnosti i pristupačnosti pruža mogućnost stvaranja aplikacija osoba na koje nisu programeri jer u kratko vrijeme mogu izgraditi potpuno funkcionalnu aplikaciju za pametne telefone i tablete te mijenja način na koji djeca uče o računalstvu. Odličan je izbor za korištenje u obrazovnim ustanova m, osobito u podučavanju i uvođenju u programiranje, što je posebno primjenjivo u sustavu osnovnoškolskog i srednjoskolskog obrazovanja.

Korisnici koji ne maju Android telefon, mogu svejedno izraditi aplikacije uz pomoć Android emulatora koji se pokreće na računalu i ponaša se baš kao i telefon. Razvojno okruženje App Inventora podržano je za operacijske sisteme Mac OS X, GNU Linux i Windows te nekoliko popularnih modela telefona za Android. App Inventor ne radi s preglednikom Internet Explorer. Aplikacije stvorene pomoću App Inventora mogu se instalirati na bilo koji Android telefon. Koristi grafičko sučelje, vrlo slično Scratchu i StarLogo TNG korisničkom sučelju koje korisnici mogućujo povlačenje i ispuštanje (*drag' n' drop*) vizualnih objekata kako bi se stvorila aplikacija.

Kako bi se pristupilo App Inventor-u, potrebno je imati kreiran Google račun.

Osim vodiča (razni *tutoridi*), dostupna je i knjiga uz koju se korisnik može brzo i detaljno upoznati s App Inventor alatom uz pomoć raznih prijera konkrenih aplikacija.

Slika 13 - App Inventor - podrška korisnicima

The screenshot shows the App Inventor website at www.appinventor.org/content/teaching-app-inventor/introduction/course-intro. The left sidebar has a 'Modules' section with 'Introduction' selected. The main content area is titled 'Course In A Box' and contains a grid of course modules:

Module	Description
Introduction	Introduction
Course Introduction	Video Tutorials (AI2)
Module Introduction	App Inventor Classic Videos (AI1)
Lesson Plan 1.1: Introduction and 'I Have a Dream'	Java Bridge
Lesson Plan 1.2: I Have a Dream: Conceptualize, Customize, Create	AI2 Book
	How do you...?

To the right of the grid, there is descriptive text: "remix them (citations/links are welcomed!), and pick and choose modules as you like. Whether you are teaching on-line, in-person, or a hybrid "flipped" classroom, the Course-in-a-Box is for you!" Below this is a section titled "Useful links" with three items:

- Ideas for Teaching a Class
- Instructor Google Group
- A Map of App Inventor Teachers

Izvor: Autorka

App Inventor se sastoji od 2 dijela: *Designer i Blocks* dijela.

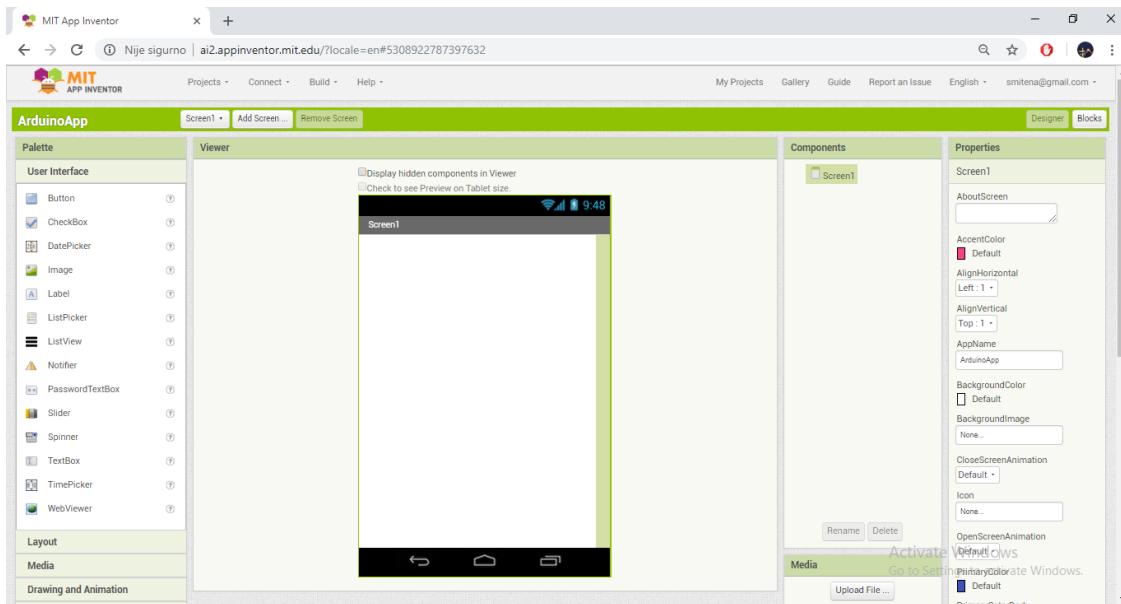
4.1 Designer dio

Designer dio sastoji se od: *Pallette*, *Viewer*, *Components* i *Properties*.

Pallette sadrži mogućnost dodavanja raznih funkcija aplikaciji povlačenjem i ispuštanjem (*drag'n'drop* metodom), koje se kasnije osposobljavaju za funkcioniranje. *Viewer* prikazuje zaslon mobilnog uređaja tj. daje prikaz kako će aplikacija izgledati na uređaju.

Components prikazuje komponente koje su dodane na zaslon. *Properties* daje mogućnost mijenjanja svojstava komponentama poput boje, visine, širine, dodavanja slike u pojedini komponentu itd.

Slika 14 - Sučelje MT App Inventor platforme



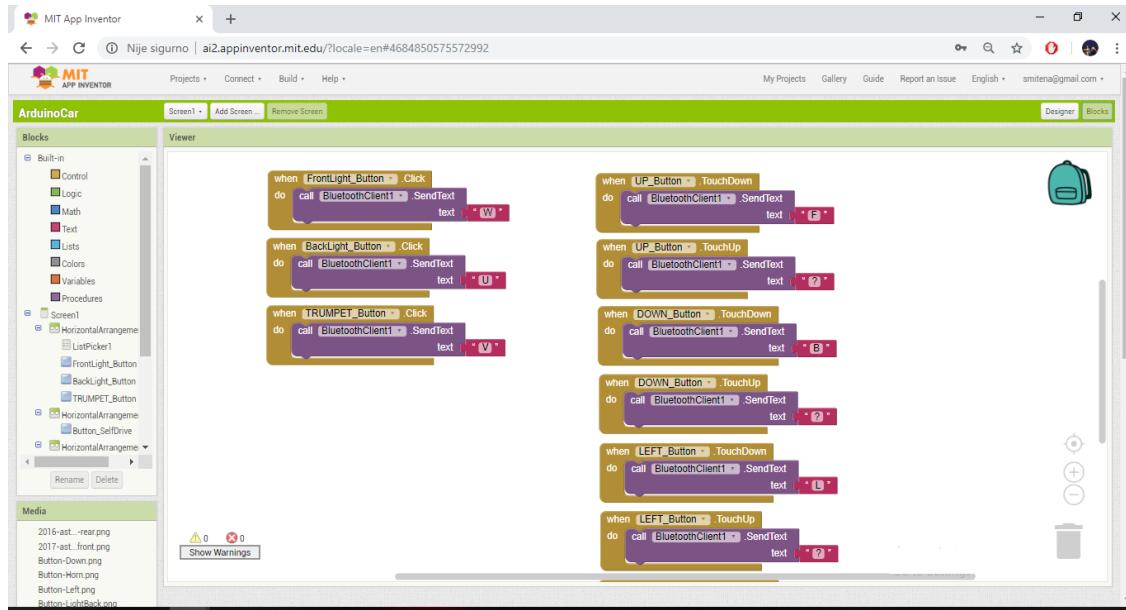
Izvor: Autorica

4.2 Blocks dio

Blocks dio sastoji se od *Blocks* i *Viewer* dijelat će se kako bi se komponentama koje su se odabrale u *Designer* dijelu dodijelile radnje. Blokovi određuju kako se komponente trebaju ponašati prilikom neke radnje te se kombinacijom blokova kreiraju funkcije potrebne za izvršavanje zadaće aplikacije.

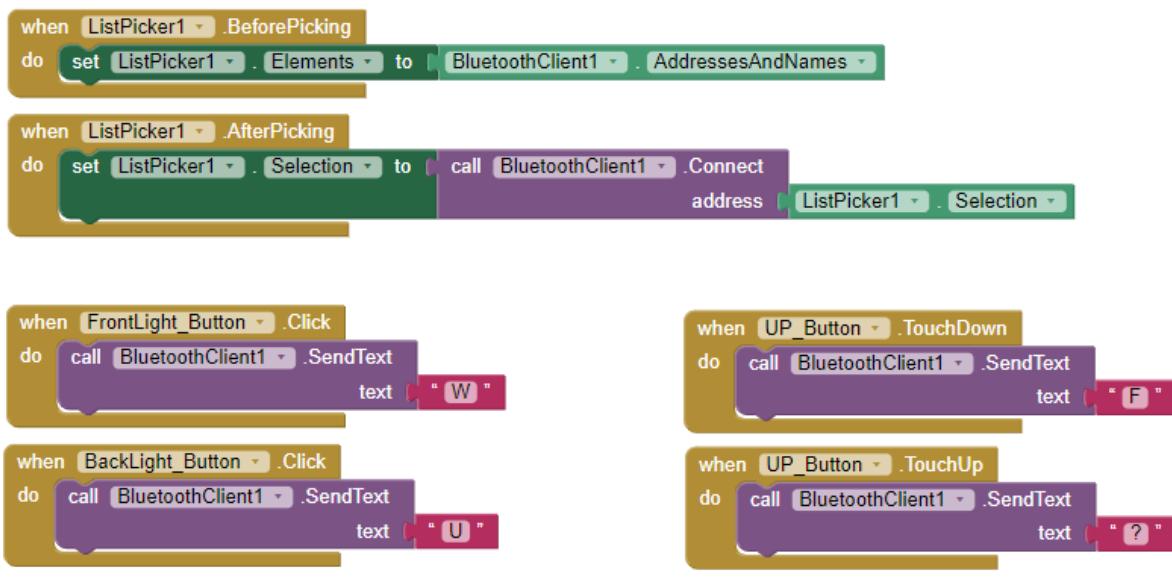
U donjem dijelu ekrana se nalaze i nadi katori su brojacim. U slučaju da postoji *Warning* aplikacija se neće moći pokrenuti.

Slika 15 - *Blocks* sučelje



Izvor: Autorica

Slika 16 - Prikaz programskog logika aplikacije (blokovi)



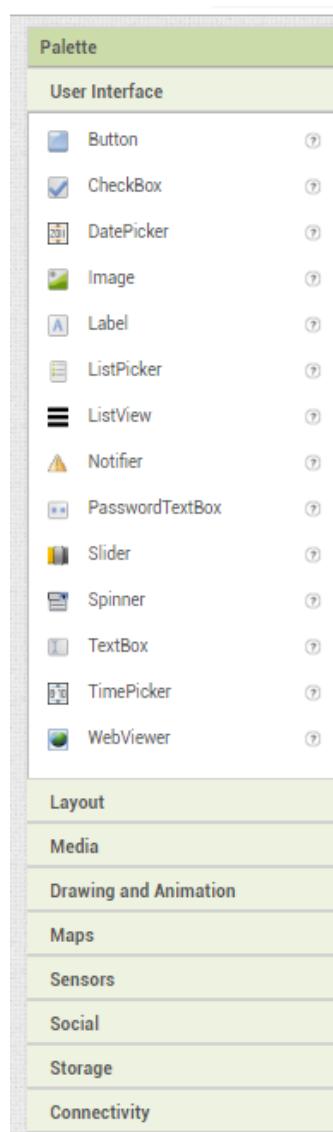
Izvor: Autorica

4.3 Dizajniranje aplikacije

Nakon što se u App Inventoru novom projektu dodijeli ime, na red dolazi dizajn. On započinje u *Designer* dijelu gdje unutar *Palette* izbornika korisnik traži željene funkcije i ih povlačenjem i ispuštanjem (*drag'n'drop* metoda) smješta u *Viewer* koji predstavlja zaslon mobilnog uređaja.

Komponente koje korisnik može odabrati u *Palette* izborniku podijeljene su u nekoliko kategorija radi lakšeg snalaženja korisnika.

Slika 17 - *Palette* izbornik



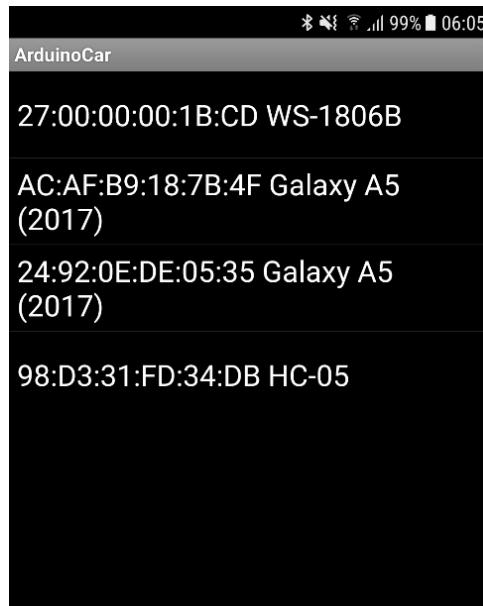
Izvor: Autorica

U izrađenoj aplikaciji korišteni su

- *Bluetooth Client* - komponenta *bluetooth* klijenta
- List Hcker
- Gu mb za prednje svjetlo
- Gu mb za zadnje svjetlo
- Gu mb za trubu,
- Gu mb za prebacivanje u automatsku vožnju
- Gu mb za snijer naprijed
- Gu mb za snijer natrag
- Gu mb za snijer lijevo
- Gu mb za snijer desno
- C ock

List Hcker je gu mb koji prilikom klikanja prikazuje popis tekstova koje korisnik može odabratiti, u ovom slučaju prikazuje popis dostupnih *bluetooth* uređaja

Slika 18 - Prikaz dostupnih *bluetooth* uređaja

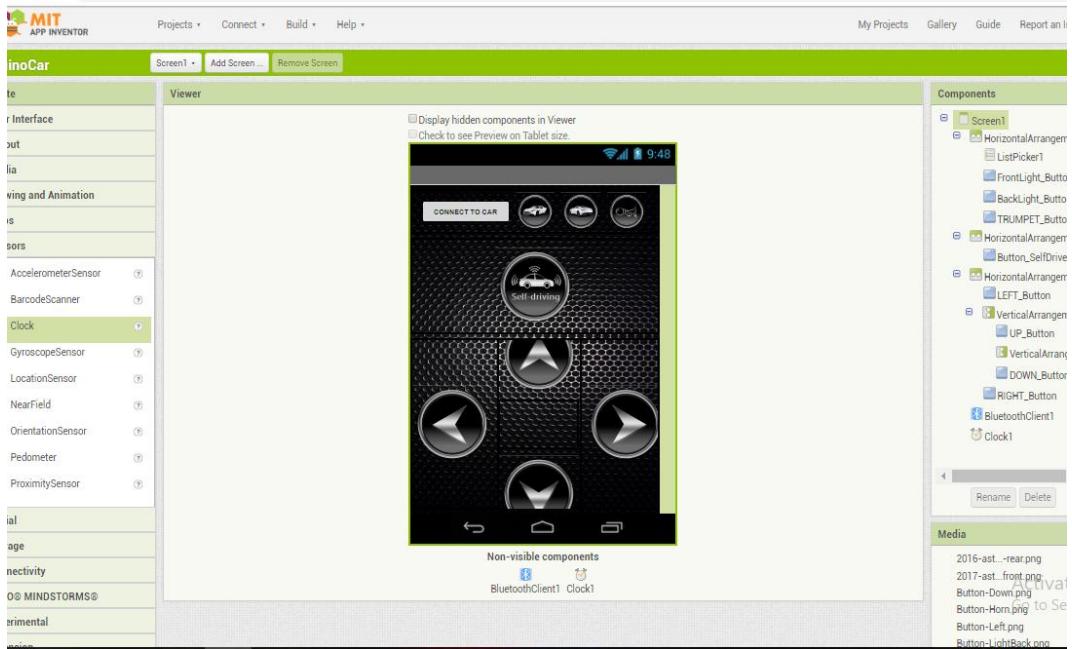


Izvor: Autorica

C ock služi za pretvaranje trenutka u tekst, odnosno, u trenutku kad korisnik klikne gu mb za prednje/zadnje svjetlo ili gu mb za prebacivanje u automatsku vožnju preko gumba će se

ispisati tekst ON kao znak da je određena naredba uključena. Nakon ponovog klika na isti gumb, tekst više neće biti prikazan.

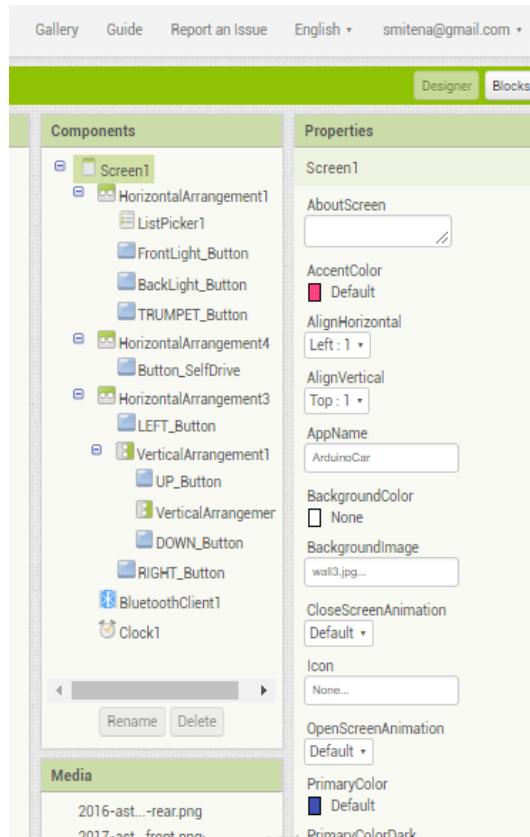
Slika 19 - Prikaz aplikacije u Designer sučelju



Izvor: Autorica

U izborniku *Components* prikazuju se komponente koje su dodane na zaslon. Klikom na komponentu unutar izbornika *Components*, u izborniku *Properties* može se mijenjati svojstvo odabране komponente poput postavljanja pozadine, boje teksta, fonta teksta, visine, širine, dodavanja slike u pojedinačnu komponentu itd.

Slika 20 - Izbornici Components i Properties



Izvor: Autorica

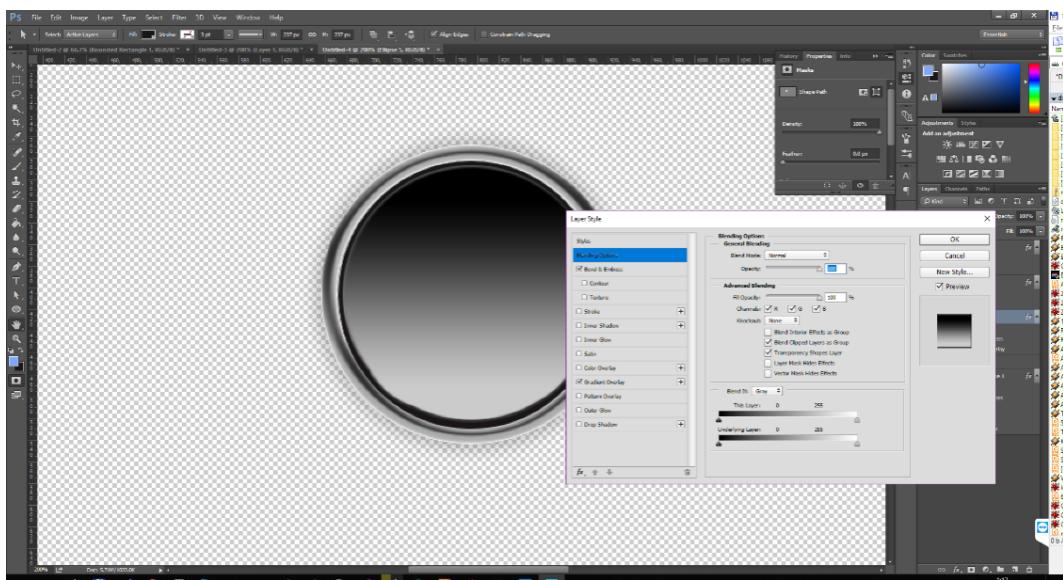
Prilikom kreiranja aplikacije je korišten Photoshop kako bi određene komponente dobile željeni izgled.

Adobe Photoshop je najpoznatiji računalni grafički alat za obradu slike.

Svi gumbi korišteni u aplikaciji izrađeni su u programu Photoshop.

Na sljedeći maksi ma je prikazan proces izrade gumba, rezultat tog procesa te prikaz same aplikacije na mobilnom uređaju.

Slika 21 - Proces izrade gumba u Photoshopu



Izvor: Aut orica

Slika 22 - Prikaz gumba Self-driving (autonomna vožnja) i gumba za trubu



Izvor: Aut orica

Slika 23 - Prikaz aplikacije na mobilnom uređaju



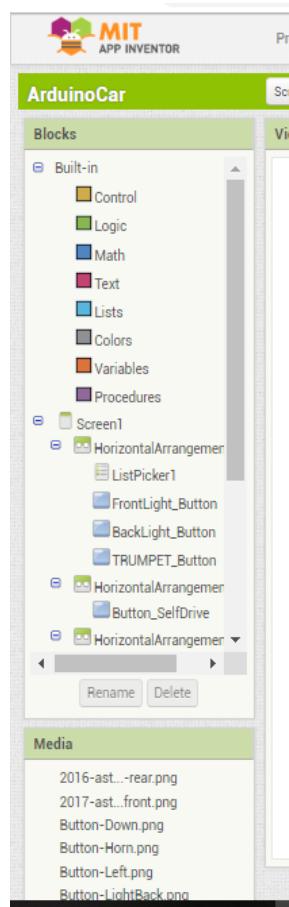
Izvor: Autorica

4.4 Konfiguracija funkcionalnosti aplikacije

Blocks di osastoji se od *Blocks* i *Viewer* izbornika. Služi kako bi se komponentama koje su se odabrale u *Designer* sučelju dodijeliće radnje.

Blocks izbornik sadrži ugrađene (*built-in*) blokove koji su dostupni bez obzira na komponente u aplikaciji koja se radi. Također, svaka komponenta ima svoj vlastiti skup blokova koji su specifični za vlastite događaje, metode i svojstva.

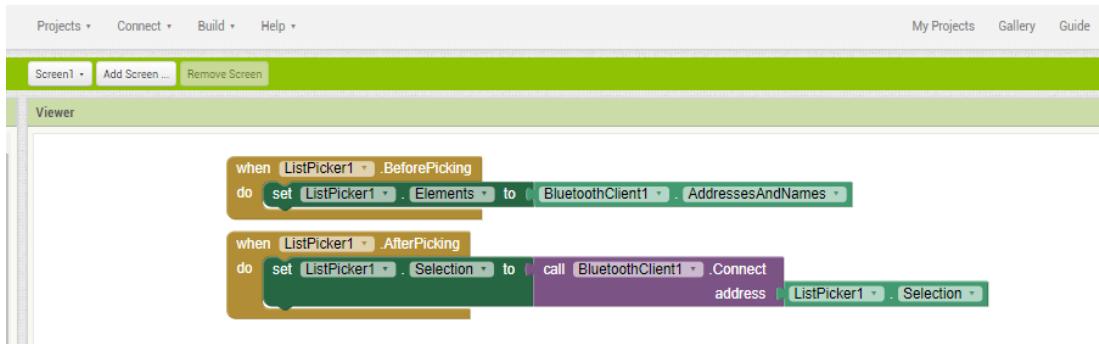
Slika 24 - *Hocks izbornik*



Izvor: Autorica

U *Viewer* dijelu se povlačenjem i ispuštanjem (*drag'n'drop* metoda) spajaju blokovi. Po završetku izrade aplikacije je potrebno provjeriti da li su blokovi ispravno posloženi (indikator u lijevom kutu ne pokazuje grešku).

Slika 25 - *App Inventor - Bluetooth klijent*



Izvor: Autorica

Odabirom gumba *Connect to car* u aplikaciji, List Hcker. BeforeHcking omogućava pri kaz popisa dostupnih *bluetooth* uređaja. Nakon što je odabran *bluetooth* klijent (List Hcker. AfterHcking), zove se *bluetooth* za konekciju i vezu i se spaja na adresu koju je korisnik izabrao.

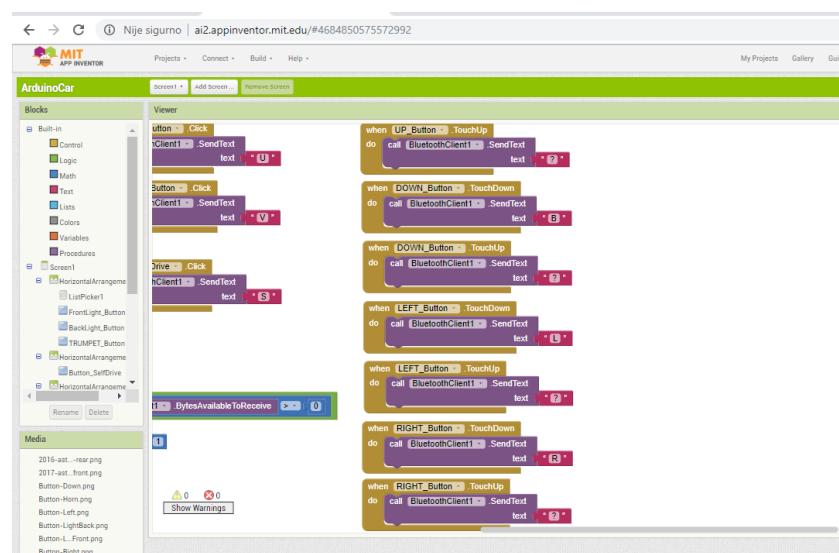
Pritisakom na gumb, šalje se tekst na povezani *bluetooth* uređaj, pročita se što je poslano i onda se prema naredbi to izvršava.

Slika 26 - Slika 25 - App Inventor - naredbe



Izvor: Autorica

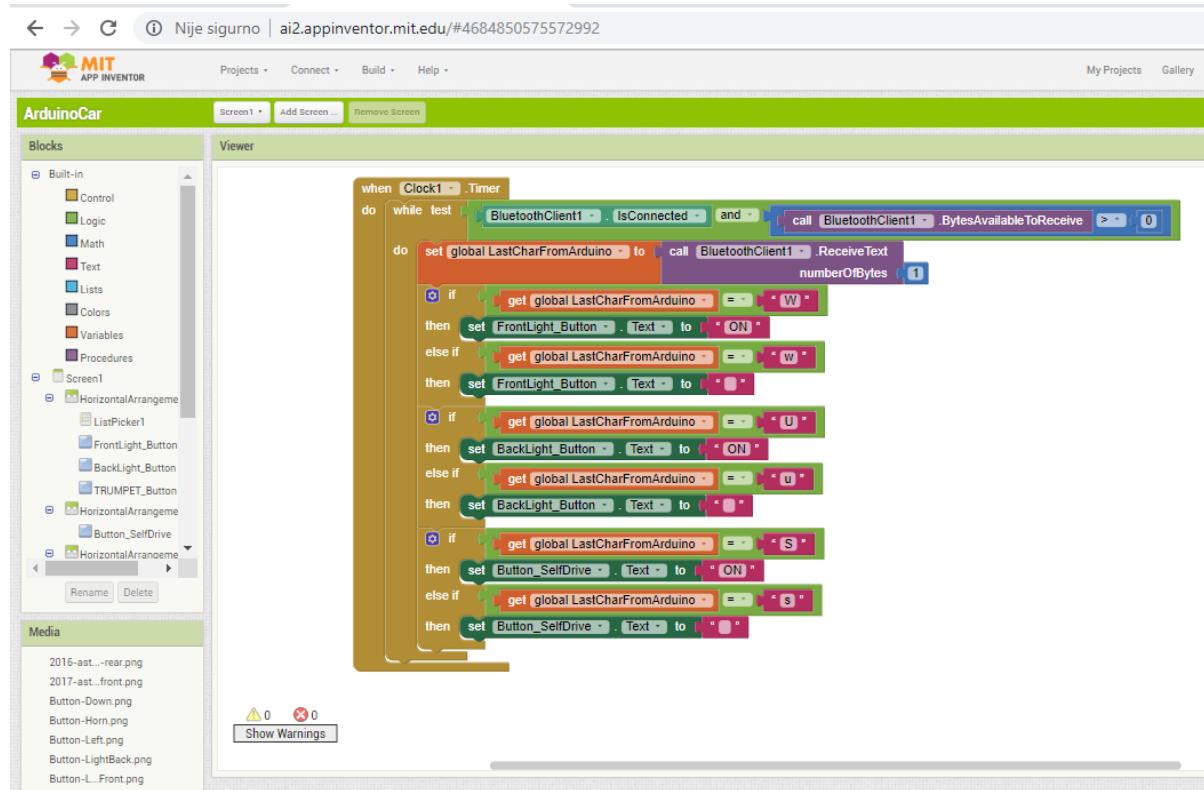
Slika 27 - App Inventor - naredbe



Izvor: Autorica

Na i dućoj slici se vidi blok Clock funkcije koja je ranije opisana.

Slika 28 - App Inventor - Clock

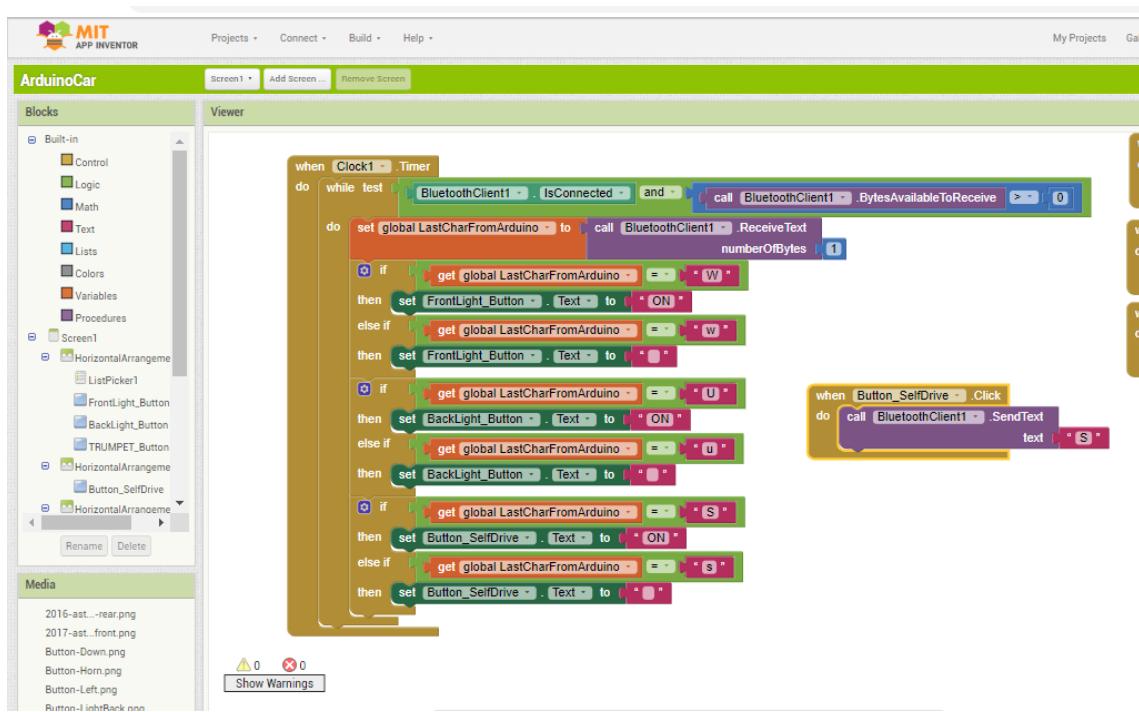


Izvor: Autorica

Prebacivanje na autonomnu vožnju odvija se na isti način kao i za ostale naredbe - šalje se tekst na povezani *bluetooth* uređaj, pročita se što je poslano i onda se prema naredbi to izvršava.

U slučaju da korisnik klikne na neki gumb s njera, automobil izbacuje iz autonome vožnje što je prethodno definirano u Arduino IDE

Slika 29 - Slika 28 - App Inventor - Autonomna vožnja



Izvor: Autorica

4.5 Provjera i pokretanje aplikacije

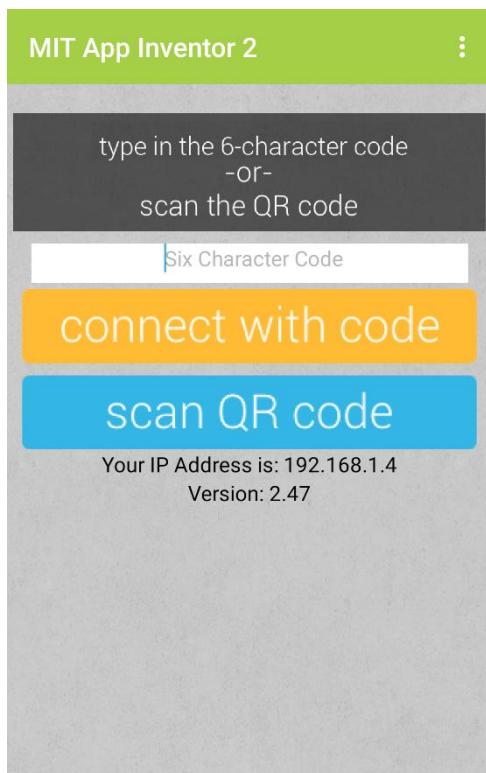
Pokretanje aplikacije može se odraditi na četiri načina:

1. M T App Inventor Companion
2. Android emulator A2
3. Povezivanje s telefonom ili tabletom pomoću USB kabela
4. Izvoz aplikacije na ne moriju telefona i instalacija

M T App Inventor Company je preporučena opcija u slučaju da korisnik koristi uređaj sa sustavom Android i ima bežičnu internetsku vezu.

Potrebno je u Google Play trgovini preuzeti aplikaciju **M T AI2 Company**. Jedini uvjet kod korištenja **M T AI2 Company** aplikacije je da mobilni uređaj na kojem se aplikacija nalazi i računalno na kojem je radena aplikacija budu na istoj Internet ili Wi-Fi mreži. Ukoliko nisu potrebno ih je unrežiti.

Slika 30 - **M T AI2 Company**

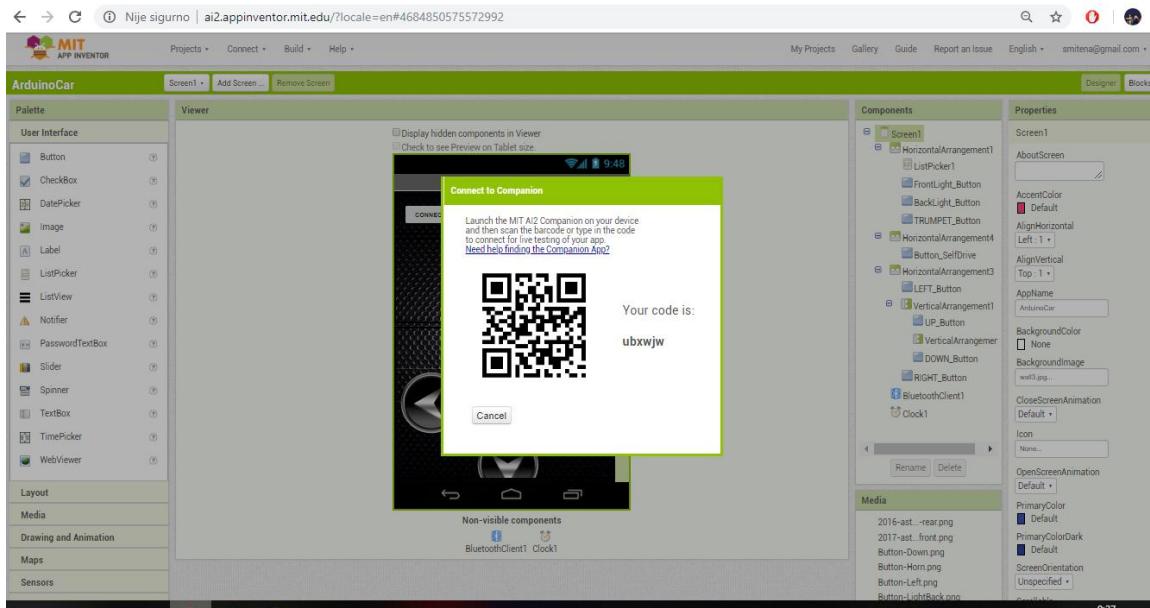


Izvor: Autorica

Da bi korisnik dobio generirani QR kod od **MT App Inventor**, na njihovoj stranici se odabire *Connect* u izborniku, a zatim *All Company* nakon čega se dobiva navedeni kod.

Nakon dobi venog koda, potrebno je očitati kod pomoću stražnje kamere nakon čega će se aplikacija otvoriti na zaslonu mobilnog uređaja.

Slika 31 - QR kod

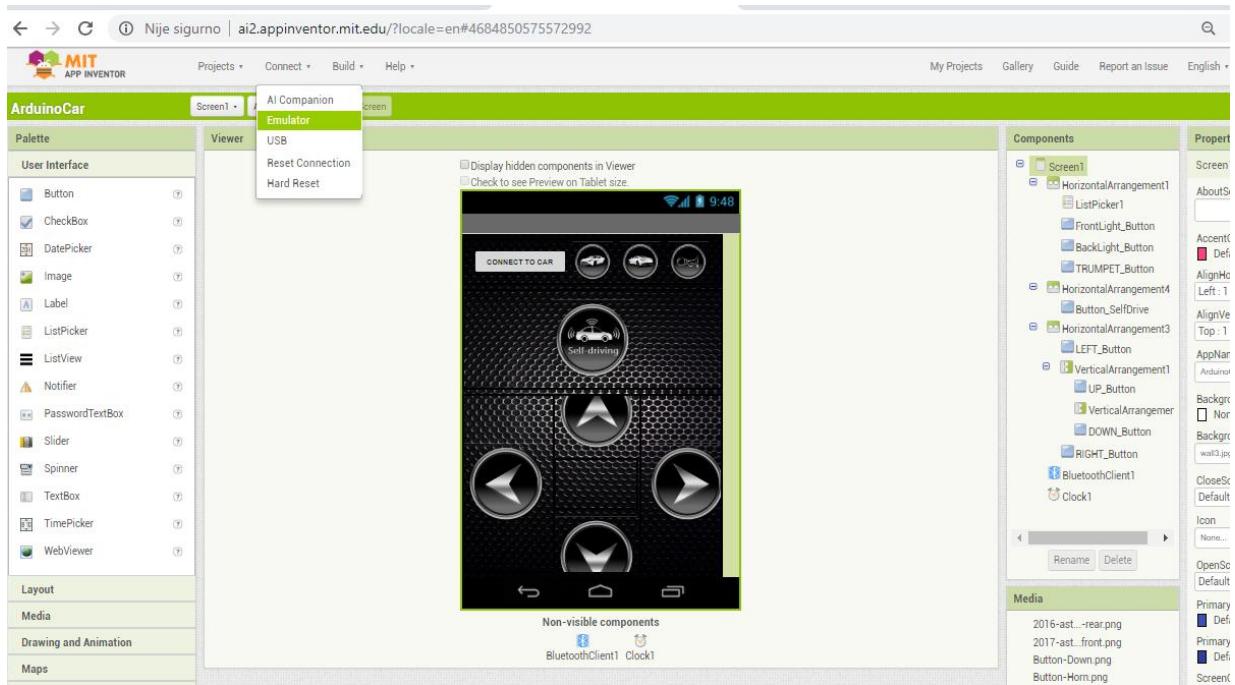


Izvor: Autorica

Drugi način je **Android emulator AI2** i najenjen je korisnicima koji ne maju Android telefona ili tablet. App Inventor pruža Android emulator, koji funkcioniра као Android, ali se prikazuje na zaslonu računala. Na taj način mogu se testirati aplikacije na emulatoru i distribuirati aplikacije drugim, čak i putem Google Play.

Emulator se može preuzeti na stranici <http://appinventor.mit.edu/explore/ai2/setup-emulator.html> zajedno sa uputama za instalaciju. Nakon toga na stranici App Inventora odabire *Connect* u izborniku, a zatim *Emulator*.

Slika 32 - Prikaz povezivanja na emulatator



Izvor: Autorica

Treći način je povezivanje s telefonom ili tabletom po mrežu USB kabelom

Postavljanje USB veze može biti neugodno, osobito na Windows računalima, koji na je potreban poseban upravljački program za povezivanje s Android uređajima, što nije slučaj s Mac računalima ili Linux operativnim sistemima koji ne zahtijevaju posebne upravljačke programe. App Inventor pruža testni program koji provjerava može li USB uređaj povezati s računalom.

Ovaj način se preporuča u okruženjima u kojima nema bežične veze ne rade. To uključuje neke hoteline, konferencijske centre i škole koje konfiguriraju svoje bežične mreže kako bi zabranile da dva uređaja na mreži međusobno komuniciraju.

Ovaj način povezivanja se provodi u 6 koraka:

1. Instalacija programa App Inventor - instalacija mora biti izvršena s računa koji i nema administratorske ovlaštosti
2. Preuzimanje i instalacija MIT AI2 Companion aplikacije na telefon
3. Pokretanje ai Starter (samo za Windows i GNU/ Linux) - Korištenje emulatora ili USB kabala zahtijeva korištenje programa ai Starter.

4. Postavljanje uređaja za USB - na Android uređaju mora biti dopušteno "USB ispravljanje pogrešaka"
5. Povezivanje računala i uređaja - uređaj mora biti povezan kao "masovni uređaj za pohranu"
6. Ispitivanje veze preko stranice za ispitivanje veze - <http://appinventor.mit.edu/test/>

Slika 33 - Preuzimanje aplikacije

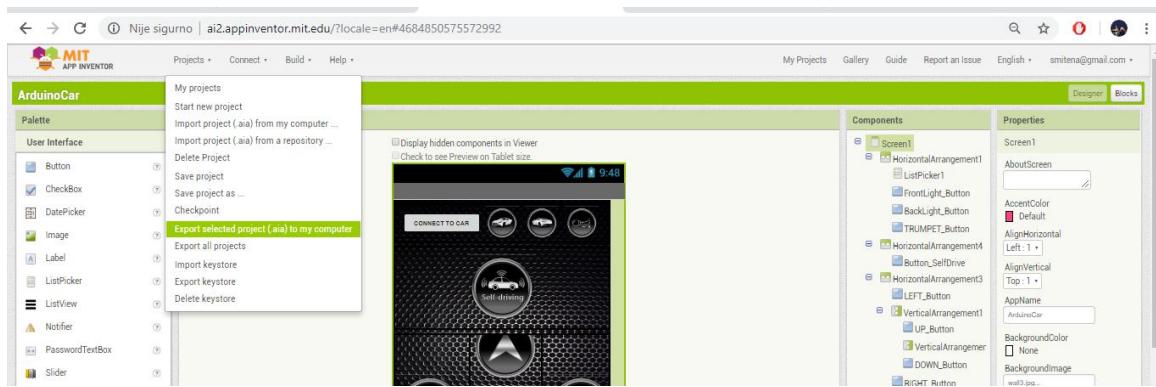


Izvor: Autorica

I posljednji način je izvoz aplikacije na memoriju telefona i instalacija

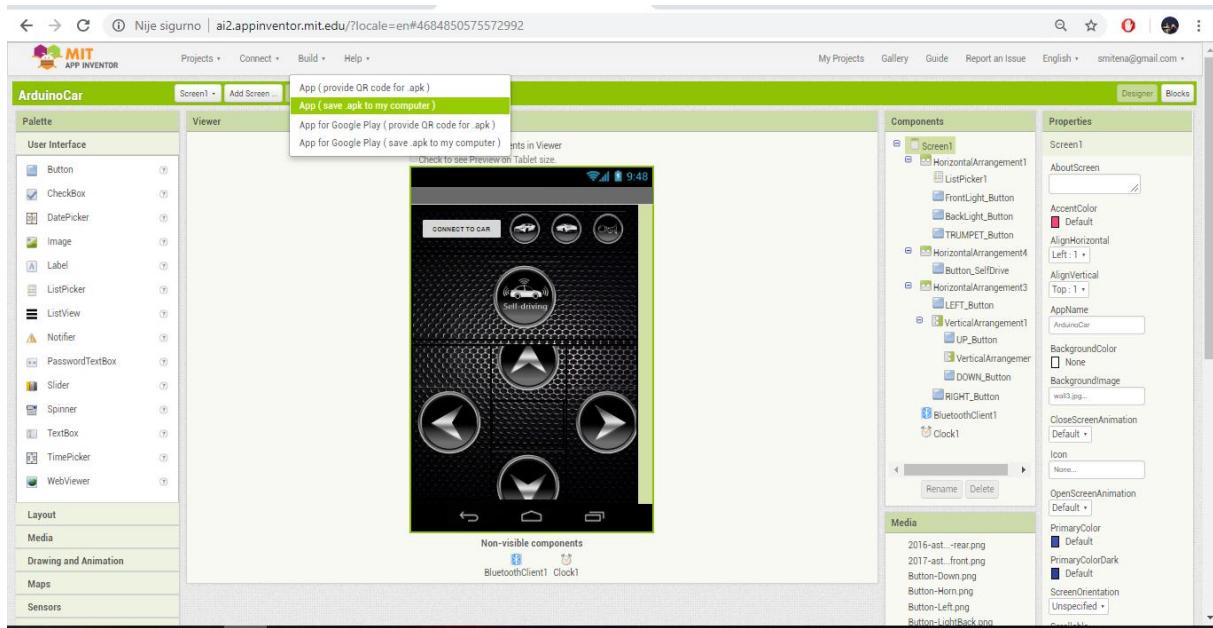
Aplikacija se može dijeliti u izvršnom obliku (.apk) koji se može instalirati na uređaj ili u obliku izvornog koda (.aia) koji se može učitati u App Inventor i ponovo koristiti. Aplikacija se može distribuirati i na trgovini Google Play.

Slika 34 - Prikaz izvoza .aia forme



Izvor: Autorica

Slika 35 - Pri kaz spre manj a .apk for me



Izvor: Autorica

5. Zaključak

Ranije je rečeno da je svrha ovog projekta približiti ljudi na tehnologiju daljinskog upravljanja korištenjem tehnologije koja omogućava visok stupanj fleksibilnosti i personalizacije. Korištenjem dostupnih ijeftinih komponenti te korištenjem App Inventora kao besplatnog i pristupačnog alata za početnike, ovaj Arduino automobil lako može naći svrhu u podučavanju i uvođenju u programiranje u sustavu osnovnoškolskog i srednjoskolskog obrazovanja.

Sa mogaćnost Arduino tehnologije omogućava da se automobil konstantno mijenja i nadograđuje što omogućava širi spektar mogućih korisnika te zadovoljenje potencijalnih želja korisnika.

Što se danjih poboljšanja tiče, moguće je nabaviti dodatne elemente poput kamere. Tada bi automobil mogao biti iskorišten od strane policije ili vatrogasaca npr. za slanje u potragu za ozlijeđeni ma.

Literatura

1. Abduzeedo, Stylish Metallic Button in Photoshop, www.abduzeedo.com/node/56065 (01.10.2018.)
2. Arduino, www.arduino.cc/en/Guide/Introduction (10.09.2018.)
3. BangGood - specifikacije UNO R3 Sensor Shield V5 Expansion Board za Arduino - www.banggood.com/UNO-R3-Sensor-Shield-V5-Expansion-Board-For-Arduino-p-954753.html? (15.09.2018.)
4. MicroPik - pdf o SG90 slijedni-motorima, www.micropik.com/PDF/SG90Slijedni.pdf (12.09.2018.)
5. Mebea - Brushless DC Motors, www.nmbt.com/brushless-dc-motors/whybrushless-motors/ (12.09.2018.)
6. MIT App Inventor2, www.appinventor.mit.edu (17.09.2018.)
7. Wikipedija, članak o Bluetooth-h-u, hr.wikipedija.org/wiki/Bluetooth (10.09.2018.)

Popis kratica

I DE - Integrated Development Environment

LED - Light Emitting Diode

QR - Quick Response

USB - Universal Serial Bus

Popis slika

Slika 1 - Prikaz Arduinovo Uno R3 ploče	3
Slika 2 - Specifikacije Arduinovo Uno R3 ploče	4
Slika 3 - Prikaz bežičnog Bluetooth prijenosnika	4
Slika 4 - Prikaz SG90 sljednog motora	5
Slika 5 - Prikaz L298N Dual H Bridge Motor Controllera	5
Slika 6 - Funkcijski dijagram te povezivanja korištenog V5 štitna	6
Slika 7 - Prikaz korištenog ultrazvučnog senzora	6
Slika 8 - Prikaz módula zvučnog signala	7
Slika 9 - Prikaz svjetlosnog senzora	7
Slika 10 - Prikaz primarnih dijelova automobila	8
Slika 11 - Prikaz sastavljenog Arduinovo automobila	9
Slika 12 - Arduinovo IDE	11
Slika 13 - App Inventor - podrška korisnicima	22
Slika 14 - Sučelje MIT App Inventor platforme	23
Slika 15 - Blocks sučelje	24
Slika 16 - Prikaz programske logike aplikacije (blokovi)	24
Slika 17 - Palette izbornik	25
Slika 18 - Prikaz dostupnih Bluetooth uređaja	26
Slika 19 - Prikaz aplikacije u Designeru sučelju	27
Slika 20 - Izbornici Components i Properties	28
Slika 21 - Proces izrade gumbu u Photoshopu	29
Slika 22 - Prikaz gumba Self-driving (automatske vožnje) i gumba za trubu	29
Slika 23 - Prikaz aplikacije na mobilnom uređaju	30
Slika 24 - Blocks izbornik	31
Slika 25 - App Inventor - Bluetooth klijent	31
Slika 26 - Slika 25 - App Inventor - naredbe	32
Slika 27 - App Inventor - naredbe	32
Slika 28 - App Inventor - Clock	33
Slika 29 - Slika 28 - App Inventor - Automatska vožnja	34
Slika 30 - MIT AI2 Company	35
Slika 31 - QR kod	36
Slika 32 - Prikaz povezivanja e-maila	37

Slika 33 - Preuzi manje aplikacije.	38
Slika 34 - Prikaz izvoza .ai a for me.	38
Slika 35 - Prikaz spremanja .apk for me.	39

Popis tablica

Tablica 1 - Troškovnik	10
----------------------------------	----