



UNIVERZITET U BIHAĆU
TEHNIČKI FAKULTET
ODSJEK: ELEKTROTEHNIČKI

PROJEKTNI ZADATAK IZ PREDMETA:
SISTEMI U REALNOM VREMENU

Mini automobil (robot) kontrolisan putem Android aplikacije i sa mogućnošću direktnog video prenosa

Eldan Hubić, 875
Ejub Piralić, 966
Hana Muslić, 987
Rejhana Nesimović, 903

Predmetni profesor: van. prof. dr. Edin Mujčić, dipl. ing. el.
Predmetni asistent: Drakulić Una, MA

Bihać, oktobar 2021.

Sažetak

U danjašnje vrijeme ljudi se susreću sa mnogim nedaćama u prometu. Nerijetko se dešavaju saobraćajne nezgode na parkinzima, prilikom parkiranja automobila. Ono što predstavlja najveći problem, jeste loša preglednost prilikom parkiranja automobila. Ukoliko preglednost nije dobra, vozač automobila ne može dovoljno dobro uočiti prepreke koje se nalaze ispred ili iza automobila, što može dovesti do nezgode. Iz tog razloga, danas, sve više automobila posjeduje senzore koji zvučnim ili vizualnim prikazom obavještavaju vozača o blizini određene prepreke. Osim senzora, u automobile se ugrađuju i kamere, koje vozaču olakšavaju parkiranje, i prikazuju mu ono što se nalazi iza ili ispred automobila. Po uzoru na jedan takav stvarni sistem, konstruisan je i ovaj projekat. Mali automobil, sa 2 infracrvena IR-08H senzora i kamerom, upravljan pomoću Raspberry Pi mikroprocesora uz kolaboraciju sa Android aplikacijom. Njegove osnovne funkcije su te, da uz pomoć senzora, automobil detektuje prepreku (bilo kakvog oblika) i da je u zavisnosti od odabrane opcije na aplikaciji (automatski ili manuelno) izbjegava. Korisnik Android aplikacije dobija direktan video prenos automobila kako se kreće, te putem aplikacije odlučuje da li će automobilom upravljati on (korisnik) ili će automobil to činiti sam. Ukoliko odabere automatsko upravljanje, automobil će se kretati sam i na osnovu senzorske detekcije, izbjegavati potencijalne prepreke. Međutim, ako korisnik odabere manuelno upravljanje, kontrola se dodjeljuje njemu i on određuje kuda će se automobil kretati. Ovaj projekat, sa svim potrebnim opcijama, bi trebao predstavljati princip kretanja automobila sa senzorima i kamerom, odnosno, stvarni sistem.

Ključne riječi: Mikroprocesor, Raspberry Pi, IR-08H senzor, kamera

Popis slika

1	Blok šema sistema	6
2	Raspberry Pi 3 Model B [9]	8
3	IR-08H senzor [11]	9
4	H-most i motor [5]	9
5	H-most L298N driver [12]	10
6	DC motor [13]	11
7	USB Kamera [15]	12
8	Šematski prikaz sistema u Fritzing softverskom alatu	13
9	Sistem sa svim hardverskim komponentama (maketa)	14
10	Pokretanje servera na Raspberry Pi-u	21
11	Izgled android aplikacije - Izbor upravljanja	22
12	Izgled android aplikacije - Automatsko upravljanje	23
13	Izgled android aplikacije - Manuelno upravljanje	24
14	Programska logika za odabir upravljanja	25
15	Programska logika za manuelno upravljanje	26
16	Programska logika za promjenu brzine i screenshot	26
17	Pokretanje aplikacije sa dugmetom za paljenje	27
18	Manuelno upravljanje mini automobilom	28
19	Automatsko upravljanje mini automobilom	29
20	Folder u koji se spremaju slike	30

Sadržaj

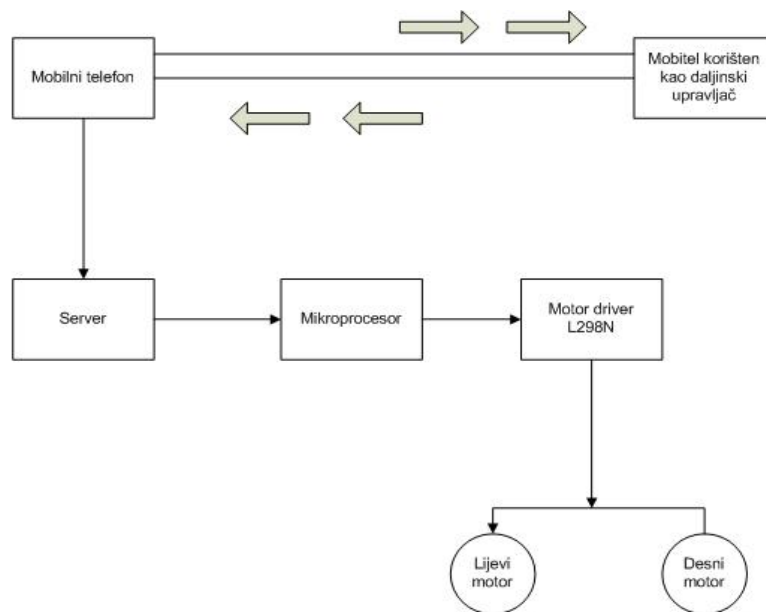
1. Uvod	5
2. Dizajn i implementacija sistema	6
2.1. Korištene komponente	7
2.1.1. Raspberry Pi 3 Model B	7
2.1.2. IR-08H senzor	8
2.1.3. H-most (L298N driver)	9
2.1.4. DC motor	10
2.1.5. USB kamera	11
2.2. Šematski prikaz sistema	13
2.3. Prikaz sistema sa hardverskim komponentama	14
2.4. Software-ska implementacija	15
2.4.1. Programski kôd	15
2.4.2. Povezivanje USB kamere sa Raspberry Pi-em	20
3. Android Aplikacija	22
3.1. Izgled android aplikacije	22
3.2. Programska logika	25
4. Eksperimentalna analiza rada sistema	27
5. Zaključak	31

1. Uvod

Cilj ovog projektnog zadatka jeste prikazati kako se jedan "mini automobil" (robot) kreće u prostoru izbjegavajući prepreke. Osnovna funkcija robota, kao i kontrola kretanja, vrši se pomoću mikrokoprocera Raspberry Pi 3 Model B. Da bi Raspberry Pi obavljao svoj posao, korišten je programski jezik Python, u kojem se nalaze sve potrebne naredbe, kao i Android aplikacija koja uz pomoć korisnika šalje potrebne komande mikroprocesoru Raspberry Pi, putem kojih se omogućava kretanje "mini automobila" u svim smjerovima, kao i direktan video prenos kretanja "mini automobila". Osim Raspberry Pi, koji predstavlja najvažniju ulogu u samom projektu, elementi bez kojih ovaj robot ne bi izvršavao svoj zadatak, su još i dva IR-08H senzora, zbog kojih se robot može kretati izbjegavajući prepreke koje se nalaze ispred njega, te dva DC motora na kojima se nalaze točkovi za kretanje.

2. Dizajn i implementacija sistema

U ovom poglavlju predstavljena je blok šema sistema (Slika 1), opisane su i prikazane korištene komponente, prikazana je šema sistema rađena u programskom paketu frizting, kao i stvarni izgled sistema. Velika pažnja se pridaje dijelu sa softverskom implementacijom sistema.



Slika 1: Blok šema sistema

Ovom blok šemom (Slika 1) prikazani su glavni dijelovi sistema predstavljeni blokovima, te povezani linijama koji prikazuju odnose blokova. U ovom slučaju su korišteni u dizajnu hardvera, kao i dizajnu softvera.

2.1. Korištene komponente

Prilikom realizacije ovog projekta, korištene su sljedeće komponente:

- Raspberry Pi 3 Model B
- 2 IR-08H senzora (barijerni senzor/senzor za prepreke)
- H most L298N
- 2 DC motora
- USB Web Kamera
- 4xAA battery pack
- Eksterni punjač za baterije (Power Bank)
- Žice

2.1.1. Raspberry Pi 3 Model B

Raspberry Pi je jeftini mikroračunar veličine bankovne kartice. Kreiran je od strane Raspberry Pi fondacije s ciljem da olakša i poboljša početnicima učenje u računarstvu. Dugim vremenskim Periodom, hardver se razvijao i poboljšavao sa svakom novom verzijom. On predstavlja uređaj koji posjeduje gotovo sve komponente koje posjeduje i obični računar, ali su smještene na maloj površini. S obzirom na njegovu pristupačnu cijenu, malu veličinu i veće mogućnosti od onih koje se dobiju na običnom mikrokontroleru, kao što je Arduino, Raspberry Pi je veoma brzo stvorio popularnost i počeo se primjenjivati u mnogim područjima. Uspoređujući ga sa stolnim ili prenosnim računarom, dosta je sporiji od njih, ali može odgovoriti na sve zahtjeve na koje odgovara i obični računar. Prednost u odnosu na računare je mala potrošnja energije, zbog čega se danas mnogo koristi u procesima stvaranja napredne robotike [3].

Na Slici 2 prikazan je Raspberry Pi kakav je korišten i u projektnom zadatku.



Slika 2: Raspberry Pi 3 Model B [9]

Raspberry Pi 3 predstavlja treću generaciju mikroračunara iz skupine Raspberry Pi. Ova generacija je 10 puta moćnija od prve generacije, ali je u potpunosti kompatibilna s njom. To znači, da je učinjeno to, da i prva i druga generacija mogu raditi sa trećom generacijom, ali mnogo učinkovitije. Jedna od najbitnijih karakteristika ove treće generacije su dodatne Wi-Fi i Bluetooth mogućnosti. To oslobađa povezivanje na računar ili internet mrežu, nego se prenos i dijeljenje podataka vrši putem Wi-Fi ili Bluetooth-a [4].

2.1.2. IR-08H senzor

IR-08H senzor je infracrveni senzor za izbjegavanje prepreka. Ovo je jeftini senzor, malih dimenzija, koji se često koristi u robotima, kao i u autoindustriji. Oni rade na način da šalju infracrvene zrake određenom frekvencijom, a objekat odbija tu svjetlost. Kada se propušteni svjetlosni valovi odbiju od objekat, prijemna cijev će primiti odbijene infracrvene valove. Ovi senzori najčešće imaju digitalni izlaz koji pokazuje da li je objekat otkriven. Mogu biti omogućeni ili onemogućeni, i imaju veoma visoku osjetljivost koja se može prilagođavati.

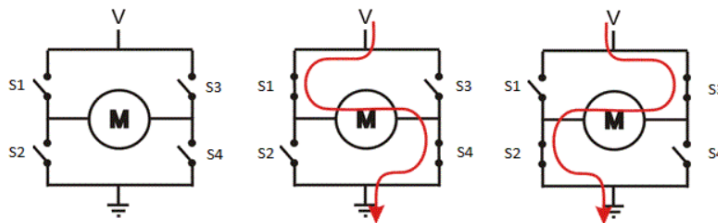


Slika 3: IR-08H senzor [11]

U ovom projektnom zadatku, korištena su dva IR-08H senzora kao sa Slike 3. Podešavanjem ovih senzora, mijenja se njihova osjetljivost, te na taj način mogu detektovati predmet u blizini. Takvo podešavanje senzora će omogućiti kretanje robota izbjegavajući prepreke, odnosno ne dozvoljavajući robotu da udari u njih [10].

2.1.3. H-most (L298N driver)

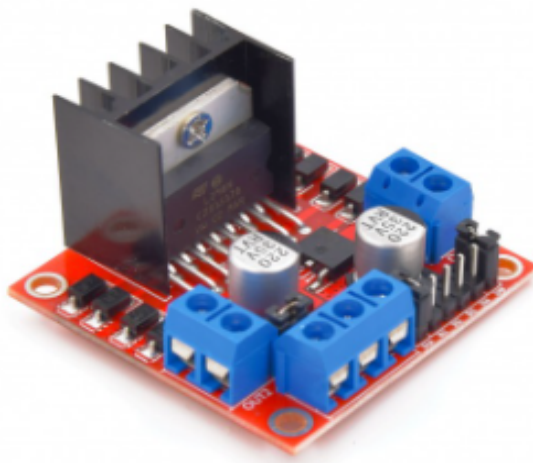
L298N driver za motor koristi H-most (engl. H-bridge) integrirani čip. H-most predstavlja strujni krug koji se sastoji od 4 prekidača spojenih sa motorom kao na Slici 4.



Slika 4: H-most i motor [5]

Ako se S2 i S3 prekidači zatvore, a S1 i S4 prekidači otvore, struja će teći u

određenom smjeru. Ako se napravi obratna situacija i prekidači S2 i S3 budu otvoreni, a prekidači S1 i S4 zatvoreni, struja će teći u suprotnom smjeru. Ono što može izazvati kratak spoj, jeste, da istovremeno budu zatvorena sva 4 prekidača ili oba prekidača na jednoj strani H-mosta.



Slika 5: H-most L298N driver [12]

Uz pomoć H-mosta se može jednostavno zamijeniti polaritet na potrošaču. Najčešće se koristi za promjenu smjera vrtnje DC motora, što je slučaj i u ovom projektnom zadatku. L298N driver u sebi ima dva izlaza kojim se može kontrolirati rad 2 DC motora, koliko je i bilo potrebno za ovaj projektni zadatak. Na Slici 5 predstavljan je izgled L298N drivera sa H-mostom [5].

2.1.4. DC motor

U projektnom zadatku korištena su dva DC motora, koja služe za kretanje autića (robota), na koje su stavljeni točkovi. Istosmjerni motor pretvara istosmjernu električnu struju u kružno kretanje. Zbog njegove mogućnosti kontinuirane promjene brzine okretanja, DC motor se koristi u industriji, te za pogon tračnih i nekih posebnih vozila. Brzina mu se mijenja na mnoge načine, a savremenim pogonima se upravlja uz pomoć računara. Zbog njegove mogućnosti napajanja iz akumulatorskih baterija, ovaj motor se koristi kao pokretač motora s unutarnjim izgaranjem. Klasični istosmjerni motor, kao

sa Slike 6 se sastoji od rotirajuće armature koja je napravljena u obliku elektromagneta s dva pola i od statora kojeg čine dva permanentna magneta. Krajevi namotaja armature, spojeni su na rotacijski prekidač, komutator, koji prilikom svakog okretaja rotora, dva puta mijenja smjer toka struje kroz armaturni namot, stvarajući tako moment koji zakreće motor [6].



Slika 6: DC motor [13]

Specifikacije DC motora:

- Dvostrana osočina 3.6mm sa otvorom 1.9mm
- Napajanje: 3-9V DC (preporučeno 4.5V)
- Dimenzije točka: prečnik 65mm, širina=27mm
- Dimenzije motora: 37.6 x 64.2 x 22.5mm
- Težina: 58g

2.1.5. USB kamera

Web kamera predstavlja malu TV kameru koja pod kontrolom računara (u ovom slučaju Raspberry Pi-a) može snimati slike i videoe u određenim formatima. Računar slike proslijeđuje na web stranicu, android aplikaciju i sl., što u pravom smislu predstavlja Web kameru. Ona se također može povezati i direktno na računar u svrhu video poziva putem interneta ili direktnog video

prenosa. Svaka Web kamera se sastoji od kućišta u kojem se nalazi detektor slike zajedno s pripadajućom elektronikom i objektivom, te priključnog kabla kojim se kamera povezuje s računarom. Ono što ove kamere čini pristupačnim jeste njihova pristupačna cijena i prateći programi koji dolaze uz kameru, potrebni za njeno korištenje. Glavni nedostatak su iznimno male dimenzije detektora (chip-a) i relativno loša kvaliteta slika [14].



Slika 7: USB Kamera [15]

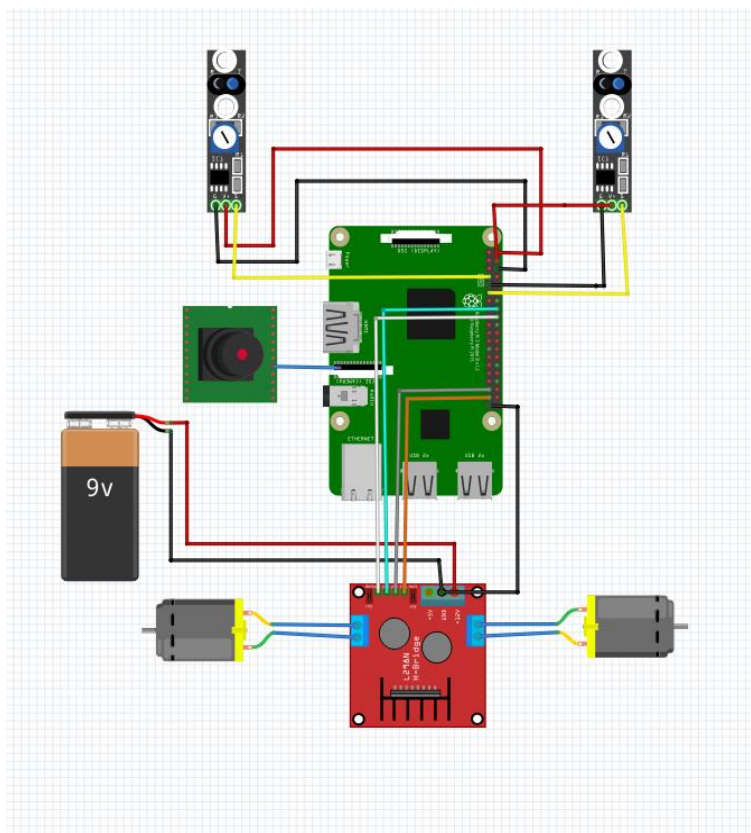
Najraširenije i najzastupljenije su kamere koje se spajaju na USB port. Takva kamera, te kamera sa Slike 7 je zastupljena i u ovom projektu.

Specifikacije USB kamere:

- Model: WB-200
- Rezolucija: 5.0 Mpx
- Dimenzije: 54 x 54 x 22mm
- USB 2.0 Interface
- Manuelni fokus

2.2. Šematski prikaz sistema

Na Slici 8 predstavljen je šematski prikaz sistema, izrađen je putem Fritzing open-source softverskog alata. To je softver za automatizaciju elektroničkog dizajna s dinamičkim interfejsom. Razvoj ovog alata služi korisnicima za dokumentaciju prototipova, izrade PCB pločica, te kreiranje završnog hardverskog proizvoda.

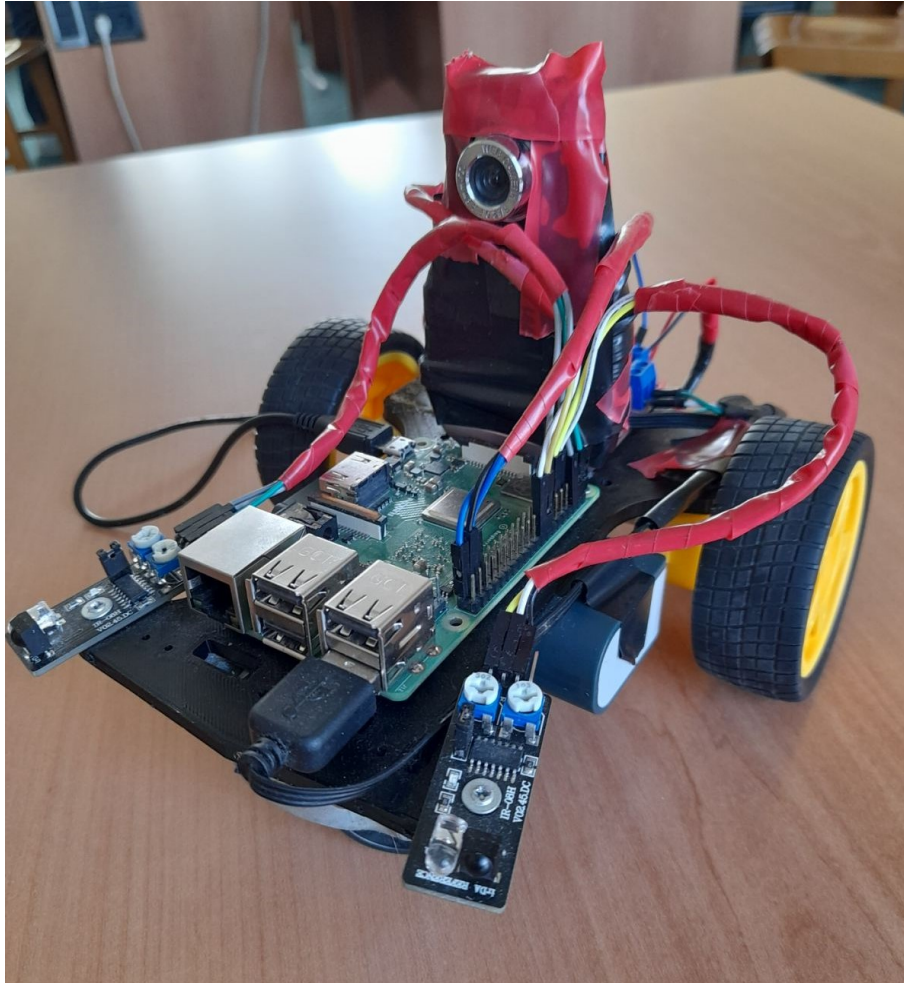


Slika 8: Šematski prikaz sistema u Fritzing softverskom alatu

Na Raspberry Pi je povezana Web Kamera, preko koje se vrši direktan video prenos. Na L298N driver (sa H-mostom) su povezana dva DC motora, i baterija od 9V. Direktno na Raspberry Pi, povezana su dva IR-08H senzora.

2.3. Prikaz sistema sa hardverskim komponentama

Na Slici 9 prikazan je stvarni izgled sistema.



Slika 9: Sistem sa svim hardverskim komponentama (maketa)

Nakon nabrojanih, opisanih i prikazanih hardverskih komponenti koje su korištene u sistemu, blok šeme sistema, kao i hardverske šeme sistema u programskom paketu Fritzing, slika iznad prikazuje jasan izgled funkcionalnog sistema, sa svim hardverskim komponentama. U sljedećem dijelu ovog poglavlja, detaljno je opisana softverska implementacija sistema sa slike.

2.4. Software-ska implementacija

Kôd za ovaj sistem pisan je u programskom jeziku Pythonu. Python predstavlja u isto vrijeme i jednostavan i moćan programski jezik. On je programski jezik opšte namjene, sa elegantnom sintaksom i dinamikom, zajedno sa svojom prirodom interpretera, ali i mogućnošću objektno orijentisanog načina programiranja [7].

Python nije jedini programski jezik koji se koristi za Raspberry Pi. Postoji mnogo jezika koji se mogu koristiti, od onih grafičkih na razini blokova, preko tradicionalnog C programskog jezika, pa sve do Basic-a i Machine Code Assembler-a. Ono što Python razlikuje od drugih programskih jezika, prilikom programiranja na Raspberry Pi-u, jeste njegova jednostavnost i modularnost, jer sadrži mnogo dodanih biblioteka koje daju funkcionalnost. Njegove mane su "uvlake", koje su veoma važne. Dodavanje uvlake na početak kôda u većini slučajeva mogu izazvati greške u kôdu, pa programer mora biti jako pažljiv i temeljit [8].

2.4.1. Programski kôd

```
1 #importovanje biblioteka
2 import RPi.GPIO as GPIO
3 from http.server import BaseHTTPRequestHandler, HTTPServer
4 from gpiozero import Robot, LineSensor
5 from time import sleep
6 from signal import pause
7 import json
8 import requests
9 import os
10
11 os.system('sudo motion') #pokretanje motiona preko terminala
12 #deklaracija google drive API tokena
13 headers = {"Authorization": "Bearer ya29.
14             a0ARrdaM_1tgnserHuCTw3aPX5MCNrV_8T9iH6MYI_txiCSz-4
15             WxmasQA3dktUtgZ2G4KkBP36bkhPx4wIjAA6l
16             cdxaaDDe7nT3RS1bRe_0d4fF4mdgpYAIyIWMcJM7_-QNyipUJk-
17             Ewf6dfApi9C8n9pqJzL"}
18 robby = Robot(left=(19,26), right=(24,23)) #inicijalizacija
19 GPIO pinova za motore
20 left_sensor = LineSensor(4) #inicijalizacija pina za lijevi
21 senzor
```



```
17 right_sensor= LineSensor(17) #inicijalizacija pina za desni
    senzor
18
19 speed = 0.65 #postavljanje trenutne brzine
20
21 GPIO.setmode(GPIO.BCM)
22 zahtjev = None
23 flag = 0
```

Kôd 1: Importovanje biblioteka i dodavanje pinova

U ovom dijelu kôda može se uočiti uvođenje biblioteka i dodavanje pinova. Biblioteke predstavljaju skupove određenih klasa, uz pomoć kojih se obavljaju potrebne operacije. Uvedena je "RPi.GPIO" biblioteka, koja u sebi sadrži klasu kojom kontroliše GPIO pinove. Uvedene su još i "BaseHTTPRequestHandler", "HTTPServer" biblioteke pomoću kojih se kontroliše komunikacija sa serverom, te "Robot" i "LineSensor" koji kontrolišu operacije za kretanje robota i aktivaciju senzora.

```
1 #mjesto spremanja slike i naziv pod kojim se sprema
2 para = {
3     "name": 'Screenshot'+str(flag)+' .jpg',
4     "parents": ["10a8BRWRwd_HI73G0yrZBIZi_W3Ua_6Py"]
5 }
```

Kôd 2: Mjesto za spremanje slika i naziv pod kojim se slika sprema

Dio kôda koji definiše mjesto gdje će se slika (screenshot) sa kamere spremiti, kao i naziv koji će se dodijeliti spremljenoj slici.

```
1 #kod za automatski nacin
2 def motor_speed():
3     while True:
4         left_detect = int(left_sensor.value)
5         right_detect = int(right_sensor.value)
6         left_mot = 0
7         right_mot = 0
8         ## Slučaj 1, oba senzora ne detektuju nista, oba
        motora idu naprijed
9         if left_detect == 1 and right_detect == 1:
10             left_mot = 1
11             right_mot = 1
```



```

12     ## Slučaj 2, lijevi senzor ne detektuje nista, desni
    detektuje, lijevi motor naprijed, desni nazad
13     if left_detect == 1 and right_detect == 0:
14         left_mot = 1
15         right_mot = -1
16     ## Slučaj 3, lijevi senzor detektuje, desni ne
    detektuje, lijevi motor nazad, desni naprijed
17     if left_detect == 0 and right_detect == 1:
18         left_mot = -1
19         right_mot = 1
20     ## Slučaj 4 oba senzora detektuju prepreku, oba
    motora idu unazad
21     if left_detect == 0 and right_detect == 0:
22         left_mot = -1
23         right_mot = -1
24     #print(r, l)
25     yield (right_mot * speed, left_mot * speed) #
    podesavanje brzine motora

```

Kôd 3: Funkcije za automatski način kretanja

Ovaj dio kôda predstavlja funkciju za kretanje robota uz detekciju oba senzora. U ovom slučaju, vrijednost "1" označava da senzor ne detektuje prepreku, dok vrijednost "0" označava detektovanje prepreke. Sukladno s tim, kada oba senzora imaju vrijednost "1" (ne detektuju prepreku), desni i lijevi motori će se kretati unaprijed (vrijednost "1"). Ukoliko desni senzor ima vrijednost "0", a lijevi senzor "1", desni motor će se kretati naprijed (vrijednost "1"), a lijevi nazad (vrijednost "-1"), i obratno. Ako oba senzora imaju vrijednost "0", odnosno detektuju predmet, oba motora će se kretati unazad i imaće vrijednost "-1".

```

1  #backend, kod za server
2  class RequestHandler_httpd(BaseHTTPRequestHandler):
3      def do_GET(self):
4          global zahtjev, robby, flag, para, files, headers, speed
          #deklaracija globalnih varijabli
5          messagetosend = bytes('Server Start!',"utf") #poruka kad
    se server pokrene
6          self.send_response(200) #pozitivan odgovor servera
7          self.send_header('Content-Type', 'text/plain')
8          self.send_header('Content-Length', len(messagetosend))
9          self.end_headers()
10         self.wfile.write(messagetosend)

```

```
11     zahtjev = self.requestline
12     zahtjev = zahtjev[5 : int(len(zahtjev)-9)] #pretvorba
    zahtjeva u string
13     print(zahtjev) #ispis zahtjeva u konzolu
14     if zahtjev == 'Manual': #ako je string zahtjev 'Manual',
        inicijalizacija robby varijable
15         robby = Robot(left=(19,26), right=(24,23))
16     elif zahtjev == 'Naprijed': #pokretanje motora prema
    naprijed
17         robby.forward(speed)
18     elif zahtjev == 'Desno': #pokretanje motora za skretanje
    u desno (lijevi naprijed, desni nazad)
19         robby.right(speed)
20     elif zahtjev == 'Lijevo': #pokretanje motora za skretanje
    u lijevo (lijevi nazad, desni naprijed)
21         robby.left(speed)
22     elif zahtjev == 'Rikverc': #oba motora idu unazad
23         robby.backward(speed)
24     elif zahtjev == 'Automatski': #pokretanje automatskog
    nacina
25         robby.source = motor_speed()
26     elif zahtjev == 'Stop': #stopiranje motora pri manualnom
    nacinu
27         robby.stop()
28     elif zahtjev == 'Stop1': #stopiranje motora pri
    automatskom nacinu
29         robby.stop()
30         robby.close()
31     elif zahtjev == 'Screenshot': #dio koda za funkciju
    slikanja
32         os.system('curl -s -o /dev/null http
        ://192.168.1.239:8080/0/action/snapshot') #naredba za
    slikanje (Screenshot)
33         para = {
34             "name": "Screenshot"+str(flag)+"jpg",
35             "parents": ["10a8BRWRwd_HI73G0yrZBIZi_W3Ua_6Py"]
36         }
37         #fajl koji se salje na google drive
38         files = {
39             'data': ('metadata', json.dumps(para), '
    application/json; charset=UTF-8'),
40             'file': open("./lastsnap.jpg", "rb")
41         }
42         #kod za slanje fajla na google drive
43         requests.post(
```

```
44     "https://www.googleapis.com/upload/drive/v3/files?
uploadType=multipart",
45     headers=headers,
46     files=files
47     )
48     flag += 1
49 else:
50     #podesavanje brzine motora
51     speed=float(zahhtjev)/100
52
53
54 return
```

Kôd 4: Funkcije u komunikaciji sa serverom

Ovaj isječak prikazuje funkciju koja surađuje sa serverom. Jasno se može uočiti da se radi o naredbama koje korisnik šalje sa određenog mjesta (u slučaju ovog projekta, korisnik preko Android aplikacije upravlja mini automobilom). Ukoliko korisnik koristi manuelni način rada, klikom na dugme on šalje zahtjev za manuelan rad i počinje upravljati mini automobilom. U zavisnosti od opcije koju odabere (naprijed, nazad, lijevo, desno, stop) određiće se kretanje mini automobila. Ukoliko korisnik želi prekinuti takav način rada, klikom na "Stop", šalje se zahtjev za zaustavljanje. Ono što je korisniku još omogućeno, je to da prati direktan video prenos preko kamere, te u bilo kojem trenutku može zatražiti slikanje (screenshot). Klikom na dugme za screenshot, izvršava se slikanje i slanje slike na Google Drive. Također, korisnik u bilo kojem trenutku može promijeniti brzinu kretanja mini automobila (raspon od 0 do 100), bilo to u automatskom ili manuelnom načinu kretanja.

```
1 server_address_httpd = ('192.168.1.239',8050) #adresa servera
2 httpd = HTTPServer(server_address_httpd, RequestHandler_httpd
3 )
4 print('Server Start')
5 httpd.serve_forever()
6 GPIO.cleanup()
```

Kôd 5: Server

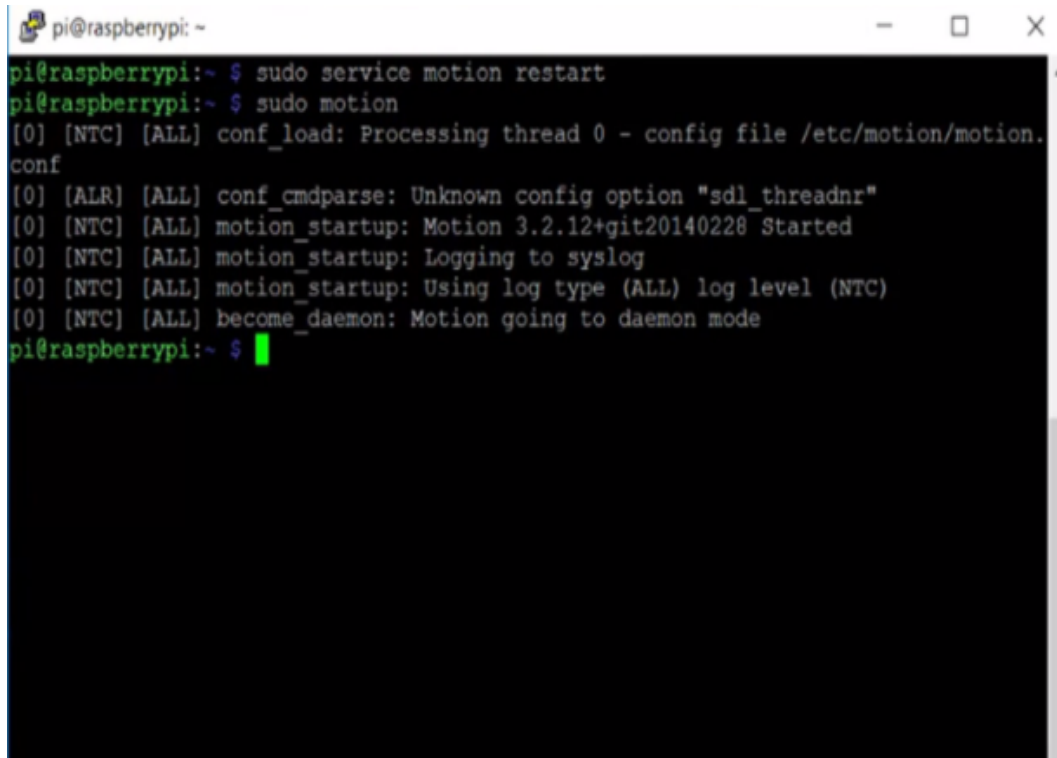
U ovom dijelu se vrši formiranje naredbi koje se šalju serveru. Dodavanje IP adrese "192.168.1.239", kao i porta "8050".

2.4.2. Povezivanje USB kamere sa Raspberry Pi-em

Nakon svih prvobitnih postavki, koje se podrazumijevaju u cmd se upisuje naredba 'sudo apt-get install motion' za početak instalacije. Nakon toga, provjerava se da li je kamera "otkrivena", ukoliko jeste, u cmd se upisuje naredba 'lsusb'. Nakon toga bi trebalo biti vidljiv naziv kamere. Ukoliko on ne postoji, vjerovatno postoji određeni problem ili kamera ne podržava "pokret". Nakon odrađene instalacije, potrebno je upisati u cmd 'sudo nano/etc/motion/motion.conf' i pritisnuti enter. Nakon toga, moraju se napraviti određene izmjene u postavkama u .conf datoteci. Nakon toga potrebno je pratiti korake:

- Provjera da li je 'daemon' uključen (ON)
- Postaviti 'framerate' bilo gdje između 1000 i 15000
- Zadržati 'Stream_port' na 8081
- 'Stream_quality' mora biti 100
- Promijeniti 'Stream_localhost' na OFF
- Promijeniti 'webcontrol_localhost' na OFF
- Postaviti 'quality' na 100
- Postaviti 'width' i 'height' na 650 i 480
- Postaviti 'post_capture' na 5
- Pritisnuti ctrl + x za izlaz. Upisati y za spremanje i enter.

Nakon ovih izmjena, potrebno je opet u cmd upisati naredbu 'sudo nano/etc/default/motion' i pritisnuti enter, zatim postaviti 'start_motion_daemon' na yes (DA), spremi i zatvoriti.



```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ sudo service motion restart  
pi@raspberrypi:~ $ sudo motion  
[0] [NTC] [ALL] conf_load: Processing thread 0 - config file /etc/motion/motion.  
conf  
[0] [ALR] [ALL] conf_cmdparse: Unknown config option "sdl_threadnr"  
[0] [NTC] [ALL] motion_startup: Motion 3.2.12+git20140228 Started  
[0] [NTC] [ALL] motion_startup: Logging to syslog  
[0] [NTC] [ALL] motion_startup: Using log type (ALL) log level (NTC)  
[0] [NTC] [ALL] become_daemon: Motion going to daemon mode  
pi@raspberrypi:~ $
```

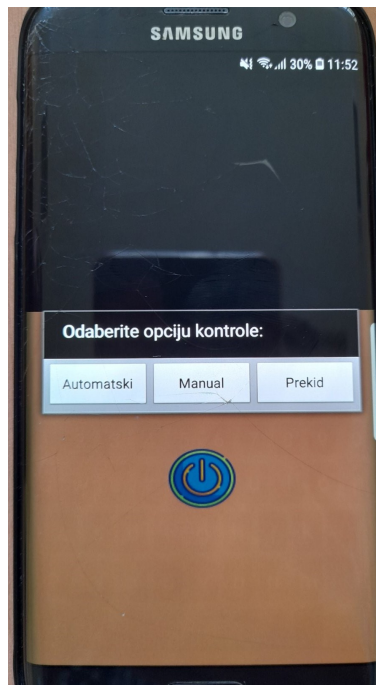
Slika 10: Pokretanje servera na Raspberry Pi-u

Prije svega, potrebno je u cmd ukucati naredbu 'sudo service motion restart' kako bi se restartovao "motion software". Nakon toga, za pokretanje servera, u cmd se unosi naredba 'sudo motion' i klikom na enter, server je spreman, što pokazuje i slika 10.

3. Android Aplikacija

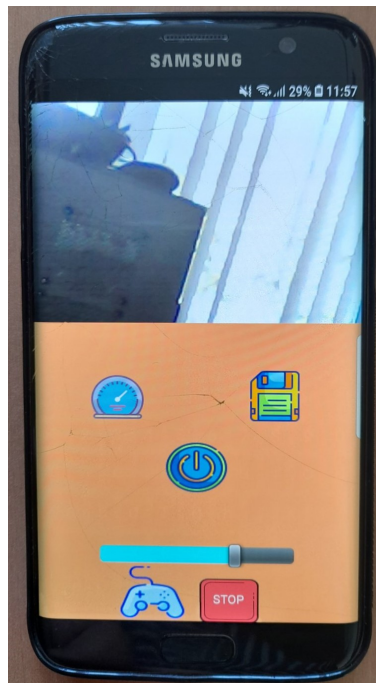
Za dizajn i programiranje aplikacije korištena je MIT App Inventor platforma. MIT App Inventor je platforma osmišljena za konceptualno podučavanje razvijanja računarskog mišljenja kroz razvoj i izradu mobilnih aplikacija. Izrada aplikacija je mnogo olakšana povlačenjem i ispuštanjem komponenti u svrhu kreiranja dizajna, te korištenjem blokova za programiranje (slaganje vizualnih blokova). Ona predstavlja razvojnu platformu koju svako može koristiti za rješavanje određenih zadataka u stvarnom svijetu. Projekt MIT App Inventor nastoji demokratizirati razvoj softvera osnažujući sve ljude, posebno mlade ljude, da pređu sa potrošnje tehnologije na stvaranje tehnologije. Unutar MIT App Inventora postoje unaprijed definisani elementi za izradu aplikacije. Koristeći različite elemente, definira se vanjski izgled aplikacije.

3.1. Izgled android aplikacije



Slika 11: Izgled android aplikacije - Izbor upravljanja

Na Slici 11 se može vidjeti izgled mobilne aplikacije na kojoj su ponuđene opcije upravljanja automobilom. Ukoliko krajnji korisnik odabere opciju "Automatski", kretanje mini-automobila će se vršiti automatski i korisnik neće dobiti mogućnost upravljanja robotom. Neće mu biti prikazane opcije za ručno upravljanje mini automobilom, nego samo "press button" za pokretanje automatskog upravljanja, što je prikazano na Slici 12.



Slika 12: Izgled android aplikacije - Automatsko upravljanje

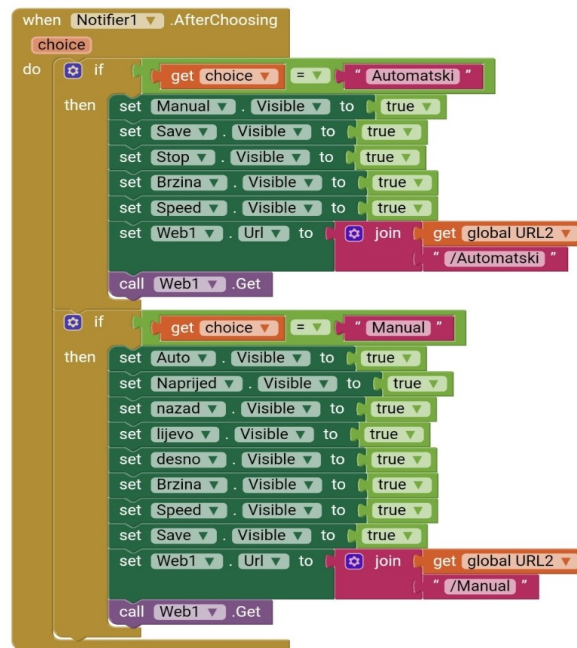
Kao što je prikazano na Slici 12, automatsko kretanje je omogućeno samo uz "press button" koji služi za pokretanje/zaustavljanje. Omogućuje se automatsko kretanje robota, koji uz pomoć svojih senzora izbjegava prepreke. U gornjem dijelu aplikacije može se vidjeti direktan video prenos. Ono što je korisniku također omogućeno jeste, klizač (eng. slider) za promjenu brzine, kao i dugme za pritiskanje (eng. push button) za slikanje. Pritiskom na dugme, korisnik uslika sliku, koja se zatim direktno šalje na korisnikov Google Drive račun.



Slika 13: Izgled android aplikacije - Manuelno upravljanje

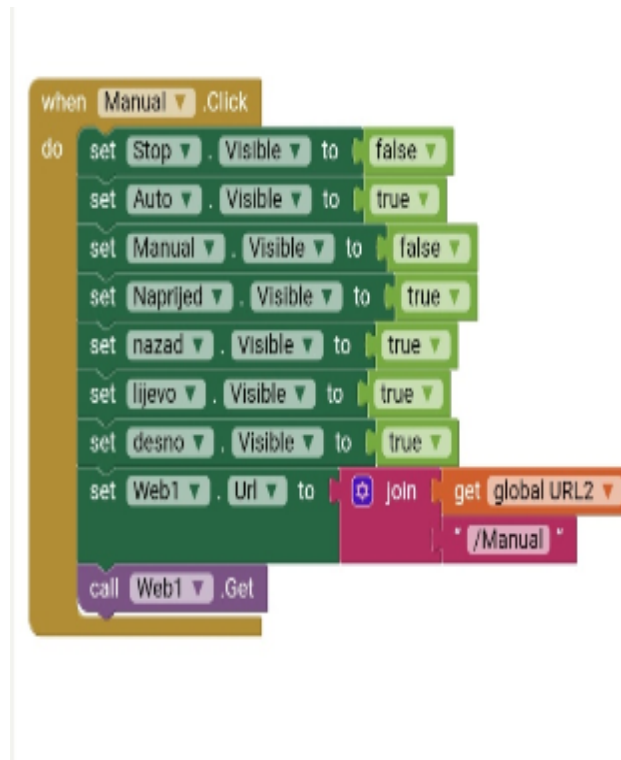
Na Slici 13 prikazan je izgled aplikacije za manuelno upravljanje mini automobilom. Korisnik upravlja robotom pritiskom na strelice. U zavisnosti od toga koju strelicu pritisne, šalje se naredba za kretanje robota u zadanom smjeru (naprijed, nazad, lijevo i desno). Također, kao i kod automatskog upravljanja mini automobilom, prikazanog na Slici 12, korisniku je omogućeno mijenjanje brzine i slikanje slike, te njeno spremanje na korisnikov Google Drive račun.

3.2. Programska logika



Slika 14: Programska logika za odabir upravljanja

Kao što je na početku samog poglavalja naglašeno, programska logika prilikom kreiranja mobilne aplikacije putem MIT App Inventor platforme se odvija pomoću blokova. Na Slici 14 prikazana je programska logika za odabir upravljanja kretanja robota. Nakon što korisnik dobije mogućnost na aplikaciji da odabere način upravljanja, njegov odabir (zahtjev) se šalje na server. Ukoliko odabere automatsko upravljanje, na zaslonu telefona se pojavljuje sučelje za automatsko upravljanje i korisnik pritiskom na "Start" pokreće robot. Ukoliko korisnik odabere manuelno upravljanje, sučelje mu omogućuje dodatne opcije, odnosno smjerove kretanja robota. Klikom na neki od smjerova, šalje se zahtjev i omogućuje se kretanje.



Slika 15: Programska logika za manuelno upravljanje

Na Slici 15 prikazana je programska logika manualnog upravljanja robotom, gdje se vidi da se poziva server i naredbe se šalju na server. Nakon što korisnik odabere manuelno upravljanje, može da bira u kojem smjeru želi da se kreće robot.

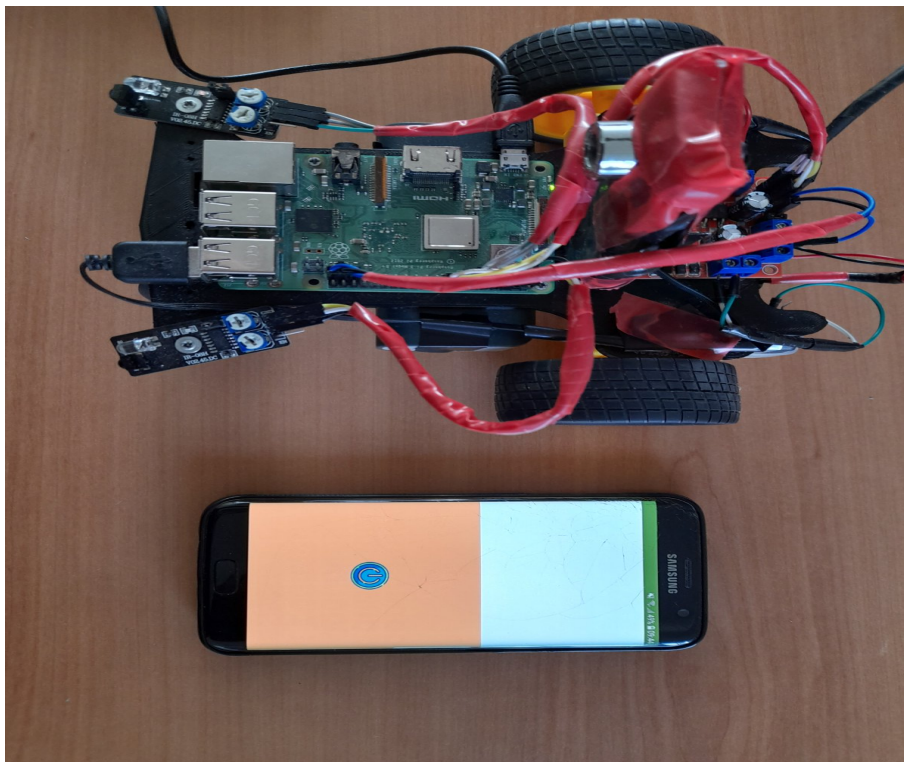


Slika 16: Programska logika za promjenu brzine i screenshot

Na Slici 16 prikazana je programska logika za promjenu brzine i slikanje slike (screenshot). Prvi blok predstavlja programsku logiku za brzinu, koja korisniku aplikacije u bilo kojem trenutku omogućuje promjenu brzine, bilo to prilikom automatskog ili manualnog kretanja mini automobila. Drugi blok je programska logika za spremanje slike (screenshot). Slika se sprema u folder na Raspberry Pi, kao i na korisnikov Google Drive.

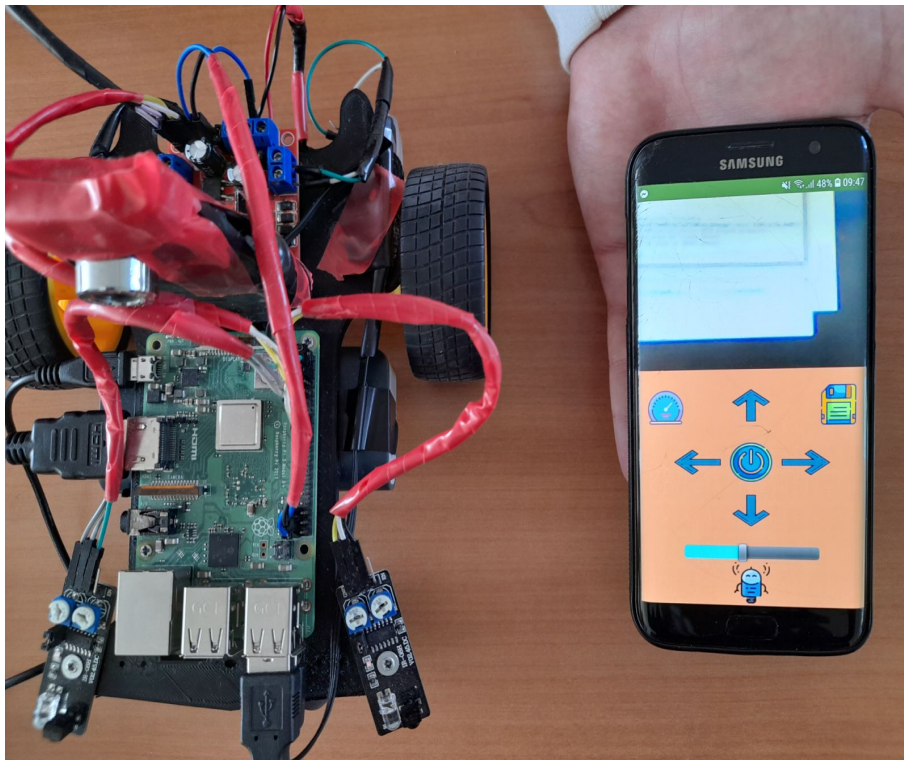
4. Eksperimentalna analiza rada sistema

U ovom poglavlju prikazana je funkcionalnost stvarnog sistema, odnosno analiza rada sistema. Interakcija krajnjeg korisnika i sistema, prikazana je u nekoliko koraka.



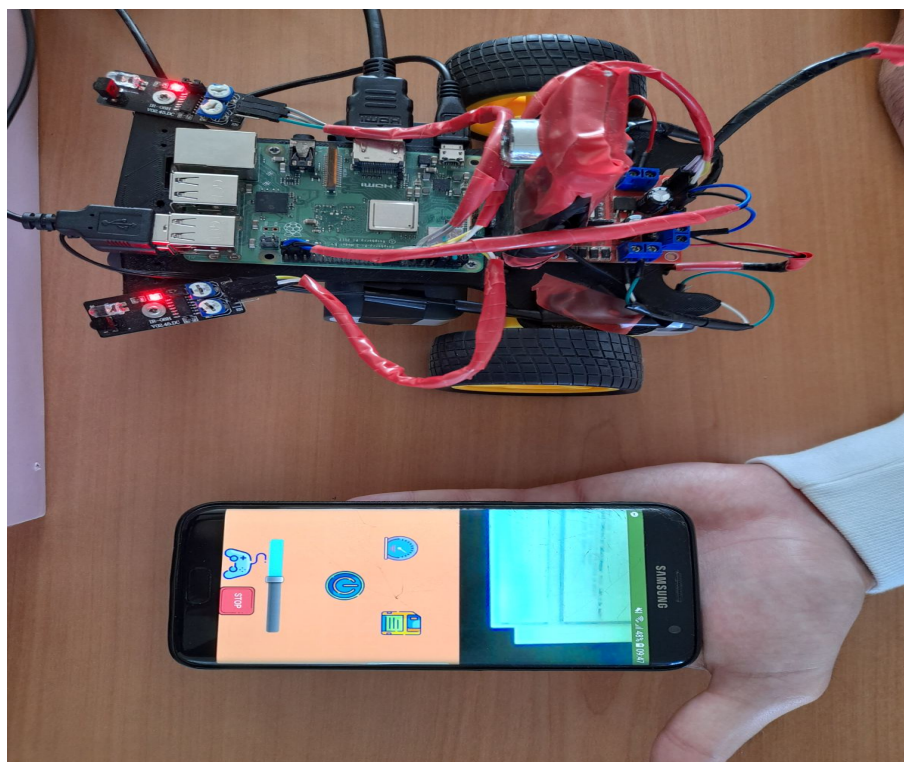
Slika 17: Pokretanje aplikacije sa dugmetom za paljenje

Nakon što korisnik pritisne dugme za paljenje, pojavljuje mu se sučelje za odabir upravljanja sistemom. Pojavljuju se tri opcije, za manuelno upravljanje, automatsko upravljanje i za prekid aplikacije.



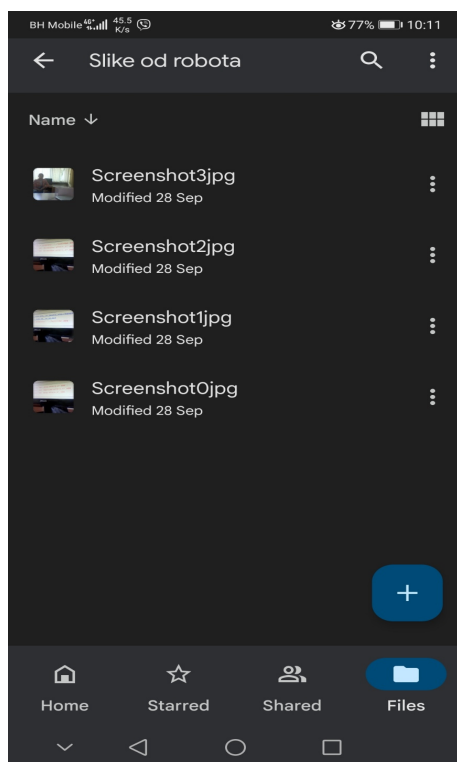
Slika 18: Manuelno upravljanje mini automobilom

Na Slici 18 je prikazano sučelje za manuelno upravljanje mini automobilom. Korisnik pritiskom na strelice, usmjerava mini automobil (robot). Ukoliko korisnik promjeni brzinu, mora je spremiti klikom na ikonu "brzinomjera". Ako korisnik želi screenshot sa kamere, potrebno je da pritisne ikonu "spremanja", kako bi se njegova slika spremila na njegov Google Drive u folder pod nazivom .



Slika 19: Automatsko upravljanje mini automobilom

Sučelje za automatsko upravljanje mini automobilom, prikazano je na Slici 19. Kao što se može uočiti, senzori su upaljeni, što znači da detektuju prepreku. Na ovaj način, se mini automobil kreće sam, izbjegavajući prepreke. Korisniku je omogućeno mijenjanje brzine, kao i spremanje fotografije, odnosno screenshot. Kao i u objašnjenju Slike 18, uslikane fotografije se spremaju u određeni folder na korisnikov Google Drive, što se može vidjeti na sljedećoj slici.



Slika 20: Folder u koji se spremaju slike

Na Slici 19 prikazane su slike (screenshot) snimljene sa kamere, i spremljene na korisnikov Google Drive. Na Google Drive-u je korisnik napravio folder pod nazivom "Slike od robota" u koji se spremaju snimljene slike.

5. Zaključak

Izradom ovog projektnog zadatka, naučeno je mnogo novih stvari, a postojeće znanje je nadograđeno. Cilj projektnog zadatka je bio taj, da se pokaže kako se uz pomoć Raspberry Pi-a može napraviti jedan veoma funkcionalan sistem, sa različitim funkcijama. Tako je od par komponenata sačinjen autić (robot) koji se kreće izbjegavajući prepreke. Da bi kretanje bilo omogućeno, bilo je potrebno koristiti dva DC motora, na koja su postavljeni točkovi, te H-most koji kontrolira vrtnju motora. Ono što daje specifičnost ovom sistemu, jesu IR-08H senzori, bez kojih sistem ne bi imao funkciju kakvu sad ima. Podešavanjem njihove osjetljivosti, te pozicioniranjem senzora, kontroliše se udaljenost senzora od predmeta u blizini. Detekcijom tog senzora, autić (robot) se može kretati ne dodirujući predmete u blizini. Međutim, čitavu funkcionalnost sistemu omogućuje Raspberry Pi (kao glavna komponenta), zajedno sa kôdom u Pythonu. Ono što projekat čini interesantnijim jeste, kamera koja je povezana na robot, te se uz pomoć nje vrši direktan video prenos prilikom kretanja robota. Dodatna stvar koja obogaćuje projekat je android aplikacija. Ona omogućuje da sistemom upravlja krajnji korisnik i da sve ovisi o njemu.

Literatura

- [1] Mikroprocesorski sistemi skripta, Dostupno na: <https://www.scribd.com/document/123570423/mikroprocesorski-sistemi-skripta>, Posjećeno: Juni, 2021. god.
- [2] Fritzing, Dostupno na: <https://learn.sparkfun.com/tutorials/make-your-own-fritzing-parts/what-is-fritzing>, Posjećeno: Juni, 2021.
- [3] Nikola Živković, Raspberry Pi, Završni rad, Osijek 2017.
- [4] Jeffrey S. Waller; Raspberry Pi 3 Model B for Beginners: Explore What Raspberry Pi 3 Model B Can Do, 2019.
- [5] KKM: DC motor dual H-Bridge, Dostupno na: <https://e-radionica.com/hr/blog/2017/12/10/kkm-dc-motor-driver-dual-h-bridge/>, Posjećeno: Juni, 2021.
- [6] Istosmjerni Motor, Dostupno na: <https://www.scribd.com/document/364799112/Istosmjerni-Motor>, Posjećeno: Juni, 2021.
- [7] Milenko Letić; Python 3: Uvod u programski jezik Python, 2020.
- [8] Tim Cox; Raspberry Pi Cookbook for Python Programmers, 2014.
- [9] Dostupno na: <https://www.chipoteka.hr/artikl/127339/raspberry-pi-3-model-b-wifi-bluetooth-onboard-8208000130>, Posjećeno: Juni, 2021.
- [10] Dostupno na: <https://osoyoo.com/2017/07/24/arduino-lesson-obstacle-avoidance-sensor/>, Posjećeno: Juni, 2021.

-
- [11] Dostupno na: <https://alexnlld.com/product/3pcs-smart-car-obstacle-avoidance-sensor-module-measuring-distance-infrared-sensor-photoelectric-switch-module/>, Posjećeno: Juni, 2021.
- [12] Dostupno na: <https://themachineshop.uk/product/dual-h-bridge-motor-driver-l298n/>, Posjećeno: Juni, 2021.
- [13] Dostupno na: <https://octopart.com/711-adafruit+industries-28550756>, Posjećeno: Juni, 2021.
- [14] Dostupno na: http://pvprm.zesoi.fer.hr/2005-2006-web/studenti_rad/dbozic/Seminar.htm, Posjećeno: Septembar, 2021.
- [15] Dostupno na: <https://www.bukalapak.com/p/komputer/aksesoris-226/webcam/2sc87n-jual-m-tech-web-camera-5-0-mp-wb-200>, Posjećeno: Septembar, 2021.