

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

IZVEDBENO RJEŠENJE

PetCare

Skrb za kućne ljubimce

Ivan Gredičak
Ivan Matković
Tomislav Matulić
Vito Papa
Matteo Samsa

Zagreb, svibanj 2019.

Sadržaj

1. Specifikacije.....	2
1.1. Hranilica.....	2
1.2. Kamera	6
1.3. Mobilna aplikacija.....	9
1.4. Detektor buke	15
1.5. Laser za mačke	16
1.6. Sustav za izbacivanje loptice.....	17
1.7. Server	19
2. Faze i rokovi	23
3. Primopredajni protokol	23

1. Specifikacije

Sustav će se sastojati od Arduino i Raspberry Pi razvojnih pločica te mobilne aplikacije. Arduino će služiti za prikupljanje mjerenja sa senzora i njihovu obradu, dok će Raspberry Pi imati ulogu servera putem kojeg će se ostvarivati komunikacija s mobilnom aplikacijom. Arduino i Raspberry Pi bit će povezani serijskom komunikacijom. Potrebno je korištenje Arduina uz Raspberry Pi zbog nepostojanja analognih pinova na Raspberry Pi-u. Sustav će biti napajan iz gradske mreže. Vezu s internetom Raspberry Pi ostvaruje integriranim WiFi modulom.

Mobilna aplikacija i Raspberry Pi će razmjenjivati podatke preko web servera, dok će Raspberry Pi i Arduino podatke razmjenjivati preko serijske komunikacije.

Za olakšati obradu serijske komunikacije, na početku svakog skupa podataka se nalazi podatak za provjeru. Ako podatak za provjeru ne odgovara zadanom, poruka se odbacuje.

Nakon njega nalazi se identifikator koji služi kako bi odlučili koja funkcija će obraditi koje podatke. Odabir funkcije je realiziran Switch Nakon navedena dva podatka nalaze se još 3 podatka. Podaci su svi tipa int, gdje je int duljine 16 bitova u arduino programskom jeziku.

Svaki modul će imati vlastitu funkciju inicijalizacije u kojoj se inicijaliziraju pinovi i potrebne funkcije.

1.1.Hranilica

Hranilica će biti izvedena od komponenti navedenih u tablici

Tablica 1 Popis komponenata

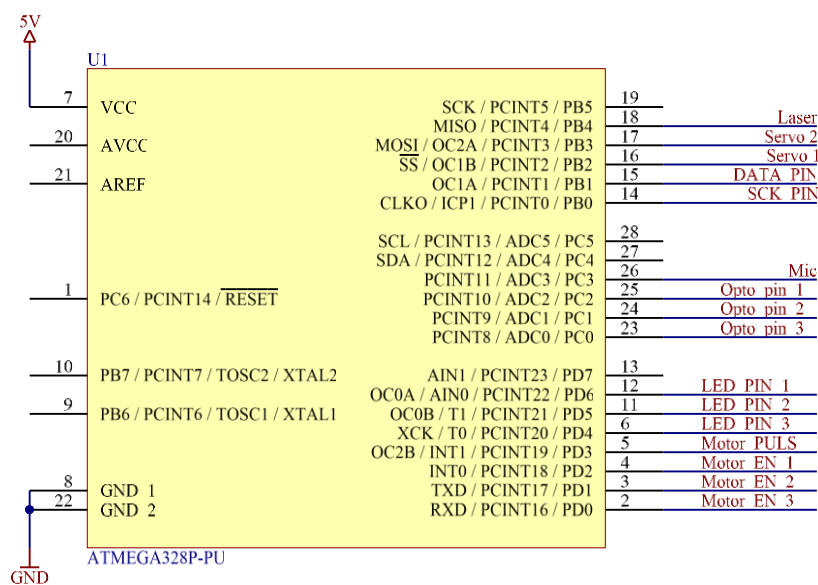
Naziv	Opis	Dobavljač	Cijena	Količina
Nema 23	Stepper motor	STEPPERONLINE	190.76 kn	3
M335	Driver motora	Makerstore	164.45 kn	3
Arduino Uno -R3**	Mikrokontroler	Arduino	147.81 kn	1
S-360-24**	Izvor napajanja	AliExpres	105.25 kn	1
MP1584EN	Izvor napajanja	Amazon	52.62 kn	1
Postolje	*	*	*	*
Zupčanik i njegovo kućište	*	*	*	*

MCCFR0S2J0	Otpornik	Farnell	0.45 kn	3
MCF 2W 330	Otpornik	Farnell	1.06 kn	3
Spremnici za hranu	*	*	*	*
NSL 19M51.	Fotootpornik	Farnell	6.64 kn	3
YZC-133	Senzor pritiska	Diykits	57.50 kn	1
HX711	Driver senzora	Diykits	29.00 kn	1

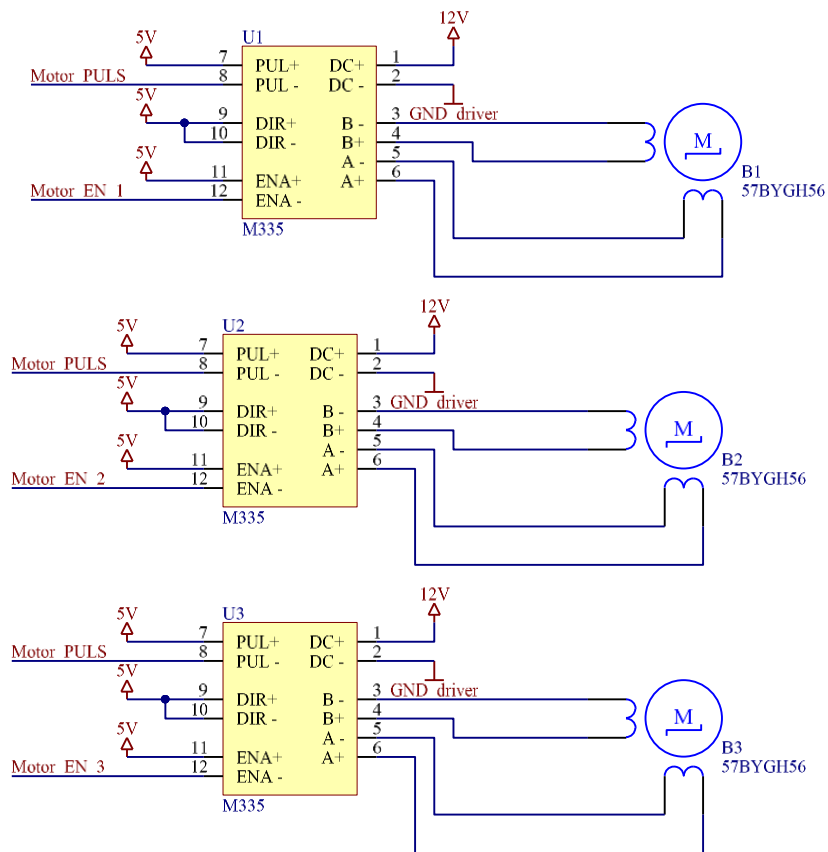
* - Komponente su samostalno izrađene

** - Komponente se koriste i u drugim djelovima projekta

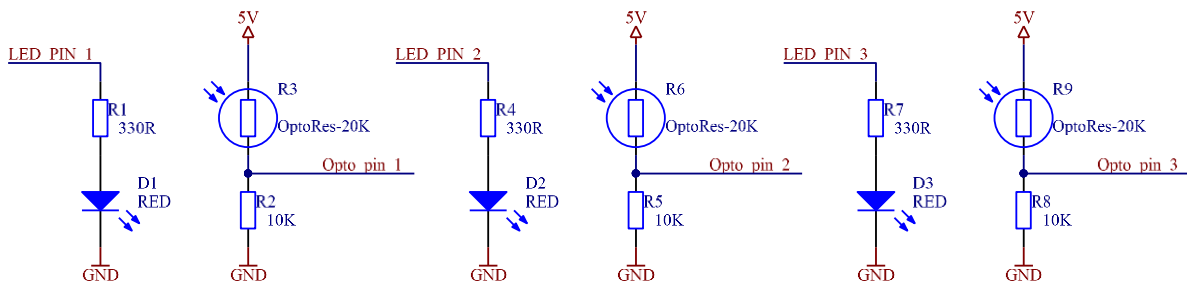
S obzirom da je u pitanju izrada prototipa, korišteni su dostupne komponente, bez obzira na njihovu cijenu i prekoračenje potrebnih tehničkih specifikacija. Električne sheme prikazane su u nastavku.



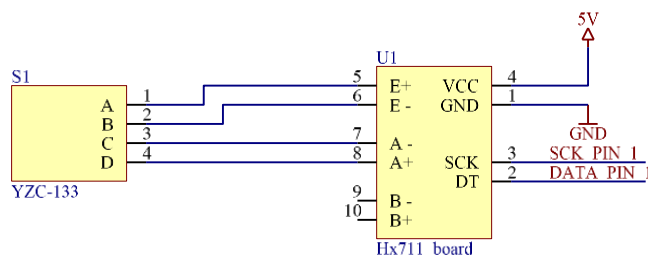
Slika 1: Spojna shema mikrokontrolera



Slika 2: Spojna shema motora



Slika 3: Spojna shema senzora za nedostatak hrane



Slika 4: Spojna shema senzora mase zdjelice

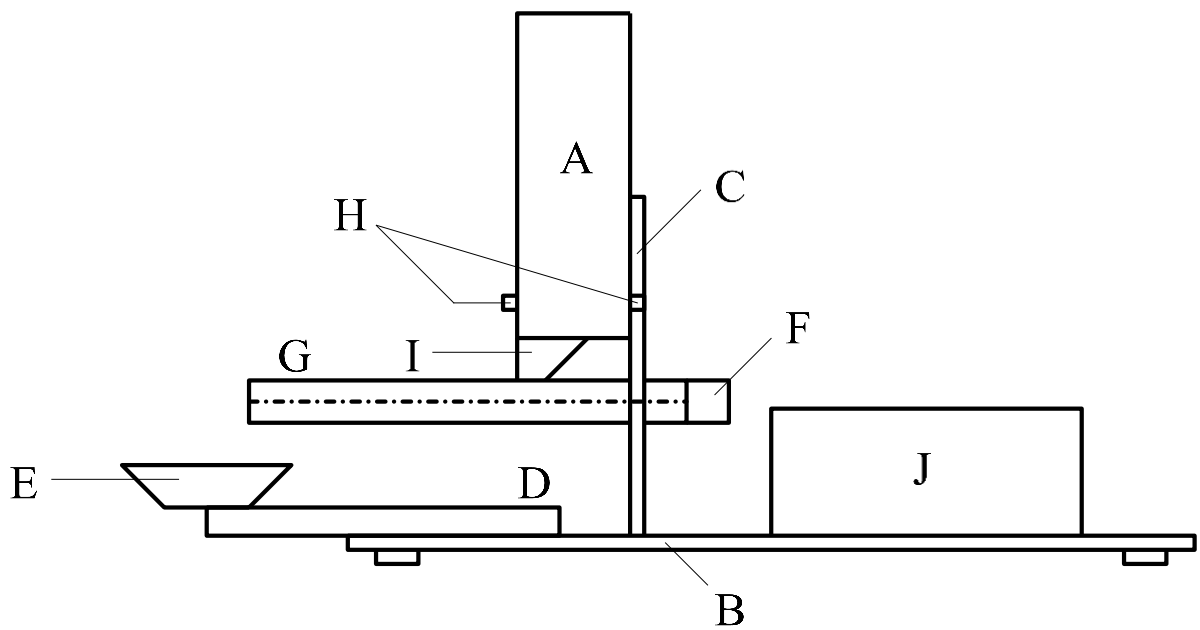
Kao je prikazano na slikama, komponente su žicama spojene na mikrokontroler. Senzor mase i njegov drajver učvršćeni su vijcima na postolje. Zupčanik u kućištu spojen je nosačima za

postolje. Motor je osovinskom spojnicom spojen na zupčanik i učvršćen na kućište. Spremnici hrane nosačima (istim kao i zupčanik) spojeni su na postolje, te učvršćeni vezicama. Svaki spremnik hrane ima oblik valjka i pri dnu se nalaze dva dijametralno suprotna otvora. U otvorima nalaze se LED dioda i fotootpornik čija je funkcija detekcija nestanka hrane u pojedinom spremniku. Pretvornici napajanja, mikrokontroler i drajver za motor učvršćeni su na postolje. Na slici 6 nalazi se prikaz hranilice sa jednim spremnikom hrane. Dodatni spremnici bit će postavljeni radijalno oko zdjelice. Dijelovi potrebni za svaki dodatan spremnik u tablici su označeni plavom bojom.

Slika 5: Prikaz pojedinih dovoda hrane u zdjelicu

Tablica 2: Komponente prikazane na slici 5

A	Spremnik hrane
B	Postolje
C	Nosač
D	Senzor mase
E	Zdjelica
F	Motor
G	Zupčanik u kućištu
H	Senzor hrane
I	Lijevak
J	Pomoćna elektronika

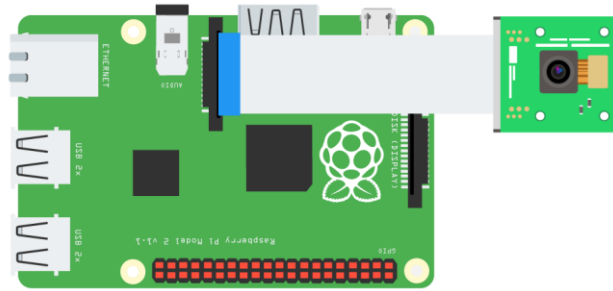


Slika 6: Shema hranilice

1.2.Kamera

Kamera korištena za video prijenos uživo kako bi korisnik mogao pratiti stanje ljubimca i prostora u kojem on boravi je Camera Board V2, 8MP namijenjena specifično za Raspberry Pi. Cijena kamere je 249 kn, a dobavljiva je u trgovini Chipoteka, korištena je u ovom projektu jer je bila dostupna izvođačima.

Spajanje kamere na Raspberry Pi je vrlo jednostavno jer postoji konektor predviđen za priključivanje kamere. Na sljedećoj slici je prikazan Raspberry Pi na koji je spojen Camera Board V2.



Slika 7. Raspberry Pi na koji je spojen Camera Bord V2

Kod za podizanje jednostavnog http servera na određenom portu je prikazan u nastavku. Na taj način se live streamu pristupa preko {ipaddress}:8080/stream.mjpg gdje je 8080 broj porta. Kao što je vidljivo iz prethodne ekstenzije format video prijenosa uživo je mjpg. Sljedeći kod će biti ubačen u skriptu u Pythonu na Raspberry Pi-u koja će biti zadužena za komunikaciju od mobilne aplikacije do PetCare sustava.

```
import io
import picamera
import logging
import socketserver
from threading import Condition
from http import server

PAGE="""\
<html>
<head>
<title>Raspberry Pi - Surveillance Camera</title>
</head>
<body>
<center><h1>Raspberry Pi - Surveillance Camera</h1></center>
<center></center>
</body>
</html>
"""

class StreamingOutput(object):
    def __init__(self):
        self.frame = None
        self.buffer = io.BytesIO()
```



```

        self.condition = Condition()

    def write(self, buf):
        if buf.startswith(b'\xff\xd8'):
            # New frame, copy the existing buffer's content and notify
            # clients it's available
            self.buffer.truncate()
            with self.condition:
                self.frame = self.buffer.getvalue()
                self.condition.notify_all()
            self.buffer.seek(0)
        return self.buffer.write(buf)

class StreamingHandler(server.BaseHTTPRequestHandler):
    def do_GET(self):
        if self.path == '/':
            self.send_response(301)
            self.send_header('Location', '/index.html')
            self.end_headers()
        elif self.path == '/index.html':
            content = PAGE.encode('utf-8')
            self.send_response(200)
            self.send_header('Content-Type', 'text/html')
            self.send_header('Content-Length', len(content))
            self.end_headers()
            self.wfile.write(content)
        elif self.path == '/stream.mjpg':
            self.send_response(200)
            self.send_header('Age', 0)
            self.send_header('Cache-Control', 'no-cache, private')
            self.send_header('Pragma', 'no-cache')
            self.send_header('Content-Type', 'multipart/x-mixed-
replace; boundary=FRAME')
            self.end_headers()
            try:
                while True:
                    with output.condition:
                        output.condition.wait()
                        frame = output.frame
                    self.wfile.write(b'--FRAME\r\n')
                    self.send_header('Content-Type', 'image/jpeg')
                    self.send_header('Content-Length', len(frame))
                    self.end_headers()
                    self.wfile.write(frame)
                    self.wfile.write(b'\r\n')
            except Exception as e:
                logging.warning(
                    'Removed streaming client %s: %s',
                    self.client_address, str(e))
        else:

```

```

        self.send_error(404)
        self.end_headers()

class StreamingServer(socketserver.ThreadingMixIn, server.HTTPServer):
    allow_reuse_address = True
    daemon_threads = True

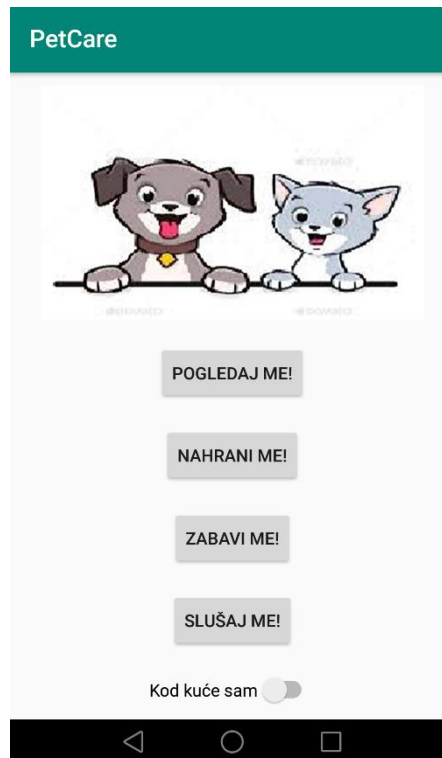
with picamera.PiCamera(resolution='640x480', framerate=24) as camera:
    output = StreamingOutput()
    #Uncomment the next line to change your Pi's Camera rotation (in
degrees)
    #camera.rotation = 90
    camera.start_recording(output, format='mjpeg')
    try:
        address = ('', 8080)
        server = StreamingServer(address, StreamingHandler)
        server.serve_forever()
    finally:
        camera.stop_recording()

```

1.3. Mobilna aplikacija

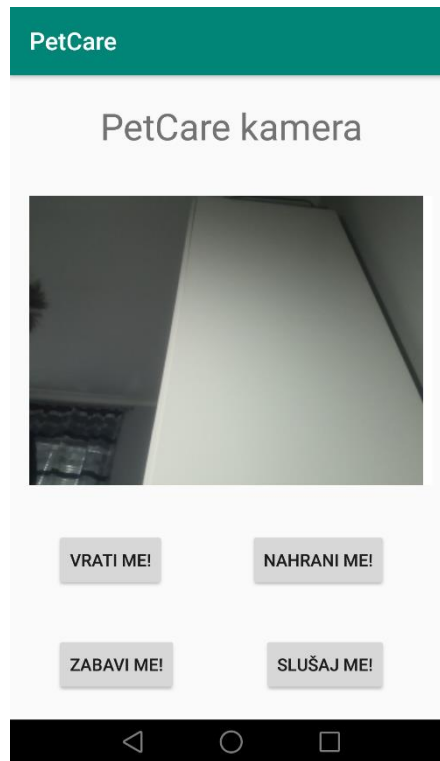
Kako bi cijeli sustav ispunio svoju svrhu potrebna je izrada mobilne aplikacije. Mobilna aplikacija mora pružiti korisniku mogućnost upravljanja dijelovima PetCare sustava i uvid u stanje sustava. Iz početnog ekrana korisnik može izabrati nekoliko opcija ovisno o tome što želi učiniti ili saznati iz sustava. Cijela interakcija između sustava i mobilne aplikacije je putem servera koji je podignut na Raspberry Pi 3B+ o čemu više informacija ima u poglavlju o serveru.

Aplikacija je pisana za Android uređaje koji imaju API level 21 i više. Aplikacija je pisana u programskom okruženju Android Studio i programskim jezikom Java. Aplikacija se sastoji od ukupno 5 ekrana. Početni ekran sadrži 4 gumba i jednu sklopku. Pritiskom na određeni gumb mijenja se ekran aplikacije, a korisnik ovisno o tome što želi učiniti odabire gumb. Svaki od navedenih ekrana će biti objašnjeni u nastavku. Sklopka 'Kod kuće sam' služi kako bi korisnik mogao sustavom upravljati i preko lokalne mreže, znajući lokalnu IP adresu Raspberry Pi-a, kada je sklopka u 'OFF' stanju tada se podrazumijeva da korisnik nije spojen na istu mrežu kao Raspberry Pi. Koncept početnog ekrana aplikacije je prikazan na sljedećoj slici.



Slika 8. Koncept početnog ekrana PetCare aplikacije

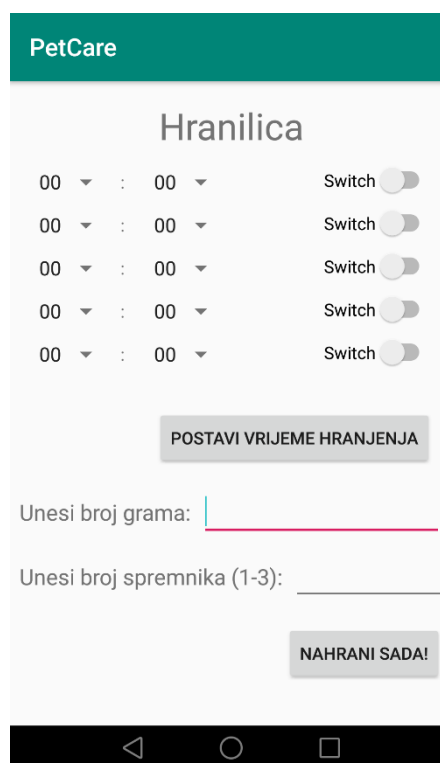
Pritiskom na gumb 'POGLEDAJ ME' otvara se novi ekran, a korisnik ga koristi u slučaju da želi provjeriti video nadzorom stanje svojeg ljubimca. Pri vrhu ekrana se nalazi natpis 'PetCare kamera' a odmah ispod se nalazi prostor u kojem je u slučaju ispravne internetske veze video prijenos uživo (s Raspberry Pi kamere). Video prijenos uživo je u formatu Motion JPG, a njega se jednostavno prikazuje pomoću objekta 'WebView', samo je potrebno poznavati URL prijenosa. U donjoj polovici ekrana se nalaze 4 gumba od kojih gumb 'VRATI SE!' služi za povratak na početni ekran, a ostali natpisom odgovaraju gumbima na početnom ekranu i pritiskom na nekog od njih korisnik otvara ekran koji bi otvorio i pritiskom na gumb sa istim natpisom na početnom ekranu. Izlaskom iz tog ekrana prekida se video prijenos uživo. Osim pritiskom na gumb 'VRATI SE!', povratak na početni ekran je moguć pritiskom na klasičan 'Back' gumb u donjem lijevom kutu (može se razlikovati za različite verzije Androida). Ekran PetCare kamere je prikazan na sljedećoj slici.



Slika 9 Koncept ekrana s video prijenosom uživo

Pritiskom na gumb 'NAHRANI ME!' (na početnom ekranu ili PetCare kamera ekranu) otvara se ekran koji se sastoji nekoliko para padajućih izbornika uz koje je pripadajuća sklopka. Svaki par predstavlja kombinaciju sati i minuta u kojima korisnik može izabrati vrijeme u kojem želi da se njegov ljubimac nahrani. Sklopke služe kako bi korisnik mogao definirati 0-5 termina hranjenja. Sati se odabiru od 0 do 23, dok se minute odabiru od 0 do 55 u koracima po 5 minuta. Pritiskom na gumb 'POSTAVI VRIJEME HRANJENJA' se navedeni aktivni termini (aktivni termini su oni kojima je sklopka u 'ON' stanju) prenose putem servera do upravljačkog dijela hranilice (upravljački dio hranilice ima konstantnu vrijednost mase i broj spremnika iz kojeg hrani ljubimca). Aplikacija najprije ulazi u funkciju koja šalje podatke na server te zatim detektira koje su sklopke u 'ON' stanju i pročita njihove pripadne sate i minute, pretvara ih u integer tip podataka (u izborniku su podaci tipa string) i šalje na server. Osim mogućnosti postavljanja vremena hranjenja, korisnik može procijeniti kako je njegovom ljubimcu potrebna hrana odmah (to može procijeniti uvidom u stanje preko kamere) te mu može određenu količinu hrane dati trenutno. To se čini tako da se u polje pokraj natpisa 'Unesi broj grama' unese masu koju želi dati ljubimcu (u gramima), a u polje pokraj natpisa 'Unesi broj spremnika (1-3)' unosi iz kojeg spremnika želi ljubimcu dati hranu (potpoglavlje Hranilica). Nakon unosa potrebnih informacija kada je korisnik siguran da želi nahraniti ljubimca prema unesenim podacima

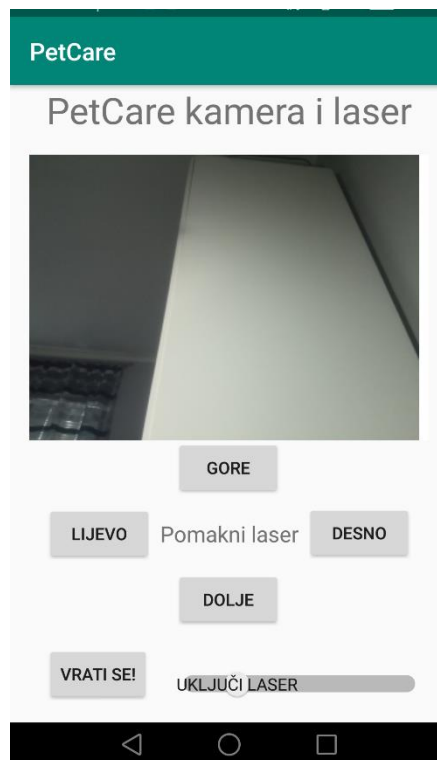
pritisne gumb 'NAHRANI SADA!' koji poziva funkciju koja šalje navedene podatke na server. Najprije je potrebno podatke pretvoriti iz tipa podataka string u tip integer, a zatim poslati putem servera hranilici PetCare sustava. Ukoliko je broj grama prevelik (potrebno dogovoriti u dogovoru s korisnikom jer ovisi o ljubimcu) ili broj spremnika nije ispravan (1-3) podaci se neće poslati PetCare sustavu i pritisak gumba 'NAHRANI SADA!' neće imati efekta. Pritiskom na gumb 'Nahrani sada' u slučaju ispravno unesenih podataka podaci će se poslati hranilici, a aplikacija će biti vraćena na početni ekran. Donji lijevi kut aplikacije je namijenjen za prikaz obavijesti o hranilici (greške ili potvrde) u tekstualnom obliku. Koncept ekrana hranilica prikazan je na sljedećoj slici.



Slika 10. Koncept ekrana hranilice

Pritiskom na gumb 'ZABAVI ME!' (na početnom ekranu ili PetCare kamera ekranu) otvara se ekran kojemu je gornji dio identičan kao ekranu PetCare kamere. Na njemu je video prijenos uživo u .mjpg formatu. Video prijenos je na ovom ekranu potreban zbog prirode njegove funkcije. Putem ovog ekrana korisnik može uključiti korištenje lasera postavljanjem sklopke u donjem desnom kutu ekrana u 'ON' stanje (tada na sklopki piše 'ISKLJUČI LASER' jer se ponovnim pritiskom na nju laser isključuje, a nakon toga na sklopki piše 'UKLJUČI LASER' itd.). Uključenjem sklopke aplikacija ulazi u funkciju koja šalje serveru informaciju kako želi

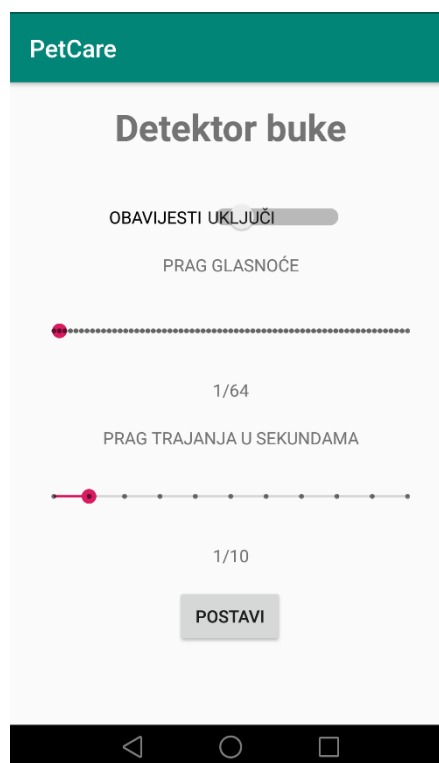
uključiti laser (tip boolean) koju on zatim prosljeđuje sustavu. Nakon što je laser uključen korisnik može pomicati smjer lasera pritiskom na gumbе koji se nalaze oko natpisa 'Pomakni laser'. Natpisi na gumbima za pomicanje lasera su smjerovi u kojima se laser miče pritiskom na gumb ('GORE', 'DOLJE', 'LIJEVO', 'DESNO'), a smjerovi se pri tome odnose na položaj lasera na video prijenosu koji je iznad navedenih komandi. Pritiskom na određeni gumb ulazi se u funkciju za slanje te se šalje relativni pomak koordinate laserske zrake (tip integer). Iz ekrana se izlazi pritiskom na gumb 'VRATI SE!' u donjem lijevom kutu ili pritiskom na 'Back' gumb. Koncept ekrana za upravljanje laserom prikazan je na sljedećoj slici.



Slika 11. Koncept ekrana za upravljanje laserom

Pritiskom na gumb 'SLUŠAJ ME' (na početnom ekranu ili PetCare kamera ekranu) otvara se ekran detektora buke. Na njemu se nalazi sklopka 'Obavijesti' kojom korisnik ovisno o stanju (u stanju 'ON' korisnik želi primati obavijesti) sklopke podešava želi li na svoj mobilni uređaj primati obavijesti u slučaju 'prevelike' buke. Koliko je prevelika buka korisnik sam definira unutar aplikacije. Na ekranu su dva 'slidera' od kojeg prvi ('PRAG GLASNOĆE') služi za podešavanje minimalne razine buke (potrebno definirati dodatno s korisnikom, ovisno o raznim faktorima) na koju će se sustav aktivirati i poslati obavijest, a drugi ('PRAG TRAJANJA U SEKUNDAMA') služi da podešavanje minimalnog trajanja 'prevelike' buke u sekundama

(dodatno definirati ovisno o raznim faktorima) nakon kojeg će sustav poslati obavijest aplikaciji. Ispod navedenih 'slidera' se nalazi brojčani prikaz trenutne vrijednosti 'slidera' i maksimalne vrijednosti. Nakon navedenih podešavanja kada je korisnik siguran da je ispravno podesio sustav pritiskom na gumb 'POSTAVI' poziva funkciju koja sve navedeno šalje preko servera do detektora buke PetCare sustava (integer tip podataka). Iz ekrana se izlazi u slučaju uspješnog izvršavanja funkcije za slanje na server (u pravilu uvijek) ili pritiskom na gumb 'Back' u donjem lijevom kutu. Koncept ekrana za podešavanje detektora buke je prikazan na sljedećoj slici.



Slika 12. Koncept ekrana za podešavanje detektora buke

Pojedini ekrani su podložni promjenama ovisno o ljubimcu i korisniku, pa je tako umjesto lasera na ekranu za zabavu moguće upravljati bacanjem loptice (prikladnije za psa), a moguće je da obje opcije budu aktivne istovremeno ako korisnik ima oba ljubimca.

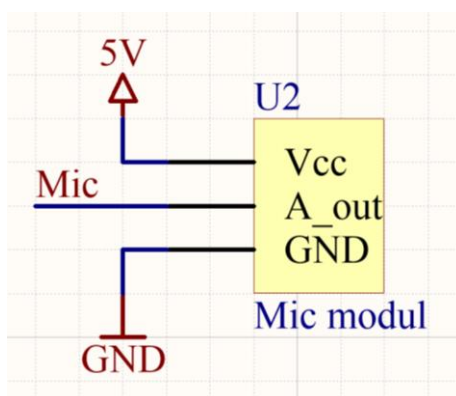
1.4. Detektor buke

Detektor buke dijeli Arduino Uno sa ostalim komponentama te osim te razvojne pločice sastoji se od:

Tablica 3. Popis komponenata

Naziv	Opis	Dobavljač	Cijena	Količina
Arduino® kompatibilni senzor zvuka, sa mikrofonom	Senzor	Protis	34.00 kn	1

Senzor zvuka ima četiri pina: Napajanje (Vcc), masu (gnd), te analogni i digitalni izlaz iz senzora. Na digitalnom izlazu će biti logička jedinica ukoliko je generirani signal iz senzora zvuka snažniji od graničnog koji se namješta pomoću trimera na samom modulu. Digitalni izlaz se u ovom projektu neće koristiti. Koristiti će se analogni izlaz koji daje analognu vrijednost napona kao funkciju glasnoće koju detektira mikrofon. Pomoću te razine moći će se odrediti ima li buke u prostoru u okolini detektora.



Slika 13. Shema povezivanja modula s mikrofonom

Programski će sustav biti riješen tako da ima funkciju za postavljanje minimalne razine buke koju detektira i potrebnog trajanja buke da bi se poslala obavijest. Postavljanje će biti vršeno preko mobilne aplikacije. Sustav će, također, imati i funkciju koja se izvršava u svakoj iteraciji beskonačne petlje mikrokontrolera koja će provjeravati trenutnu vrijednost buke te ju uspoređivati sa zadanom razinom. Ako je razina buke veća od zadane razine, od tog trenutka

započinje brojanje vremena. Ako buka traje duže od zadanog poslati će se podatak putem serijske komunikacije na Raspberry Pi razvojnu pločicu te će taj podatak biti zapisan na web server kojeg će aplikacija na mobitelu pročitati. U tom trenutku na mobitelu se ispisuje obavijest.

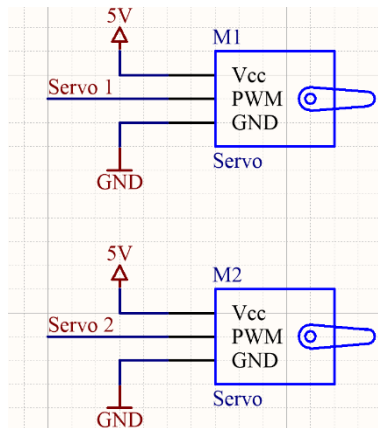
```
void init_detektor(void);
void set_detektor(int enable_notification, int volume_threshold, int duration_threshold);
void detektor(unsigned long current_time);
```

1.5. Laser za mačke

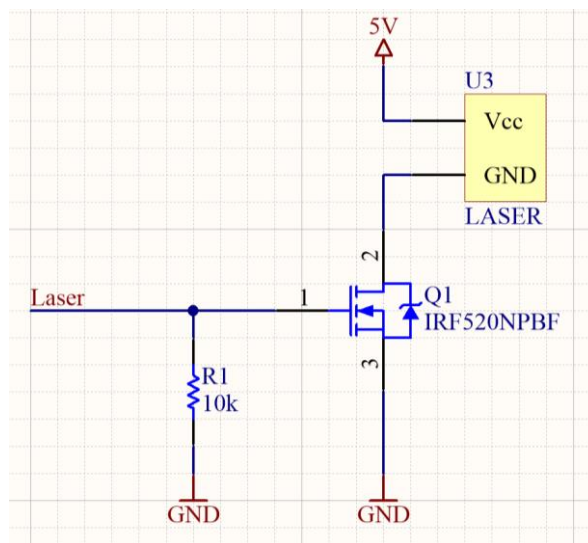
Sustav igračke lasera za mačke sastoji se od 2 servo motora te laser diode. Servo motori su postavljeni tako da jedan motor služi za rotaciju vertikalne osi, dok je drugi pričvršćen na njega i upravlja nagib u odnosu na podlogu. I laser dioda i motori su napajani vanjskim izvorom napajanja (ne preko arduino razvojne pločice) kako se nebi pojavili problemi u radu razvojnih pločica zbog limitirane struje koju imaju na ulazu. Laser dioda se uključuje pomoću N-kanalnog MOSFET-a tako da se na njegovu upravljačku elektrodu dovede signal. Kao N-kanalni MOSFET se koristi komponenta IRF520.

Tablica 4. Popis komponenata

Naziv	Opis	Dobavljač	Cijena	Količina
Servo motor mini, VMA600	Servo motor	Protis	35.00 kn	2
Laser dioda, 5V, 5mW, 650nm	Laser dioda, crvena svjetlost	Ebay	1.33 kn	1
IRF520	N-kanalni MOSFET	Mouser	6.77 kn	1
Otpornik 10k	Otpornik	X	X	1



Slika 14. Shema povezivanja servo motora



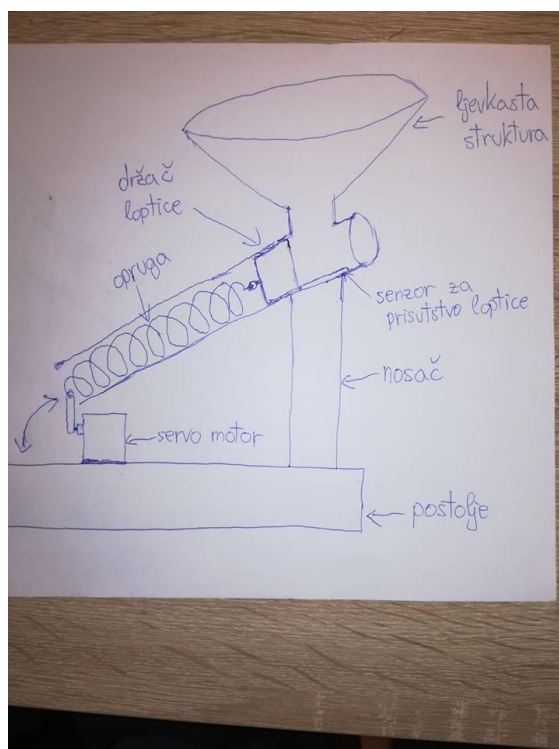
Slika 15. Shema povezivanja laser diode

Sustav zabave za mačke će programski biti riješen tako da prima naredbe s mobilne aplikacije. Podaci koje će primiti su: „enable“ koji ako iznosi 1 će upaliti laser, te koordinate x i y, odnosno položaje na koje je potrebno postaviti 2 servo motora.

```
void init_laser(void);
void laser(int E, int x, int y);
```

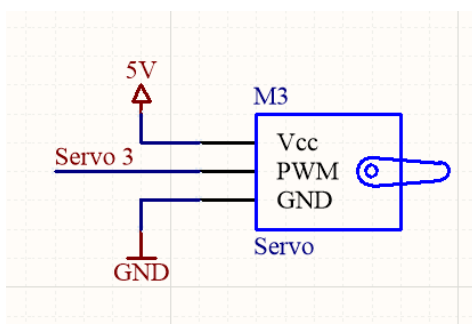
1.6.Sustav za izbacivanje loptice

Slika 16. prikazuje skicu sustava za izbacivanje loptice. Sustav se sastoji od postolja, nosača, servo motora, opruge, senzora prisustva, ljevkastrure, cijevi i držača loptice. Budući da se radi o prototipu, postolje, nosač i cijev iz koje se loptica ispucava bit će izrađeni samostalno.

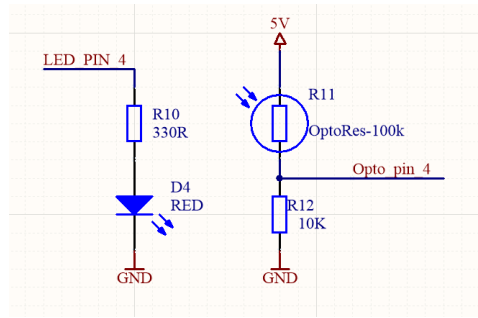


Slika 16. Skica sustava za izbacivanje loptice

Kod držača gdje loptica dolazi u stabilno stanje, nalazi se senzor prisustva koji se sastoji od fotootpornika, 2 otpornika i LED diode. LED dioda osigurava konstantnu razinu osvijetljenosti fotootpornika pri čemu je njegov otpor jako nizak te je pad napona na njemu mali (niska razina). Kada loptica uđe u sustav, ona zakloni svjetlost, otpor fotootpornika poraste višestruko, a time i pad napona na njemu (visoka razina). Nakon što se detektira prisustvo loptice, servo motor napne oprugu i sustav ispuca lopticu njenim otpuštanjem. Spojne sheme za senzor prisustva i servo motor prikazane su na slici 17 i slici 18.



Slika 17. Shema povezivanja servo motora



Slika 18. Shema povezivanja senzora za prisutstvo loptice

Tablica 5. Popis komponenata

Naziv	Opis	Dobavljač	Cijena	Količina
Servo motor EMX-SV-0272	Servo motor	Chipoteka	75,05 kn	1
Fotootpornik 100k	Fotootpornik	x	x	1
Otpornik 10k	Otpornik	x	x	1
Otpornik 330	Otpornik	x	x	1
LED	LED – crvena	x	x	1
Opruga	Opruga	x	x	x

1.7. Server

U svrhu izrade servera koristit ćemo Raspberry Pi 3B+ na kojem je instaliran operacijski sustav Raspbian. Unutar operacijski sustav Raspbian pokrenut ćemo *Apache* server. To je *open source web server* za operacijske sustave temeljen na Unixu. *Apache* server pokrećemo s razlogom kako bismo mogli ostvariti komunikaciju koristeći pametni telefon putem kojeg će korisnik biti u mogućnosti upravljati laserom, kontrolirati rad sustava loptice za izbacivanje te se po potrebi povezati sa sustavom kako bi dobio sliku s kamere. Za pokretanje, resetiranje ili gašenje servera na Unix sustavima u terminalu pišemo naredbe redom:

```
$ sudo service apache2 start
$ sudo service apache2 restart
$ sudo service apache2 stop
```

Ako se na zaslonu terminala pojavi upozorenje da *apache2* nije instaliran moramo ga instalirati i podesiti parametre. Pozivom naredbe

```
sudo apt install apache2
```

instaliramo *apache2* na sustav. Dodatno, u datoteci `/etc/apache2/sites-available/gci.conf` možemo podešavati razne postavke.

Osim Apache servera, omogućit ćemo i SSH vezu. Za podizanje SSH servera prvo moramo instalirati `openssh-server` naredbom:

```
sudo apt install openssh-client  
sudo apt install openssh-server
```

Datoteka koja se nalazi na `/etc/ssh/sshd_config` omogućuje konfiguraciju SSH servera. Slično kao i Apache serverom, SSH server možemo pokrenuti, resetirati i ugasiti naredbama

```
$ sudo systemctl start ssh
```

```
$ sudo systemctl restart ssh
```

```
$ sudo systemctl stop ssh
```

Osim toga, u datoteci za konfiguraciju moramo promijeniti u

```
PasswordAuthentication yes
```

Pomoću naredbi

```
$ sudo useradd -s /path/to/shell -d /home/{dirname} -m -G {secondary-group} {username}
```

```
$ sudo passwd {username}
```

napravimo korisnika i dodijelimo ga u grupu. Ovim postupkom je omogućujemo prijavljivanje korisnika (aplikacije).

JSON format koristimo za komunikaciju između servera i android aplikacije. Preko aplikacije se čita datoteka u kojoj je JSON format te nam daje informacije o sustavu – količinu hrane u hranilici, položaj lasera, stanje sustava za izbacivanje loptice (upaljeno/ugašeno)... Osim toga, aplikacija može slati JSON format na server i tako mijenjati položaj lasera, stanje sustava za izbacivanje loptice... JSON format je oblika:

```
{
```

```
"loptica" : "1",  
  
"laserx" : "135",  
  
"lasery" : "113",  
  
"laserz" : "66",  
  
"hran" : "44"  
  
}
```

Po potrebi se mogu dodati ili maknuti stavke. Dok je "loptica" : "1" sustav za izbacivanje loptice će raditi. Ako korisnik želi prekinuti izbacivanje loptice šalje JSON format na server te stavlja "loptica" : "0". Takvu promjenu će Raspberry Pi proslijediti dalje. U nastavku dajemo pseudokod za *python* skriptu koja rješava čitanje JSON formata i prosljeđuje informaciju dijelu sustava kojem je informacija potrebna.

Učitaj datoteku sa stranice <http://www.xyz.com/yzx.html>

Pročitaj i spremi stavke JSON datoteke u varijable

Pošalji varijable na serijski (USB) port

Osim prethodnog, moramo napraviti i čitanje sa serijskog (USB) porta. Naime, preko USB-a ostvarena je komunikacija s Arduinoom. Čitanje na Raspberry Pi-u možemo ostvariti pomoću koda (*baud rate* je 9600bps, a uređaj je spojen na USB0):

```
import serial  
  
ser = serial.Serial('/dev/ttyUSB0', 9600)  
  
while 1:  
  
    if(ser.in_waiting >0):  
  
        line = ser.readline()  
  
        print(line)
```

Arduino kod za slanje je:

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

void loop(){
  //Salji na seriju pomoću Serial.write()
  //Po potrebi delay(1000); (npr. 1000 milisekundi=1 sekunda)
}
```

Ovime smo ostvarili komunikaciju s Arduina prema Raspberry Pi-u. Za slanje s Raspberry Pi-a prema Arduinu koristimo kod za čitanje na Arduinu koji je:

```
int r = 1;
void setup(){
  Serial.begin(9600);
}
void loop(){
  if(Serial.available()){
    r = r * (Serial.read() - '0'); //Pretvara tip varijable char u int
    Serial.println(r);
  }
}
```

Konačno, Raspberry Pi pomoću sljedećeg koda može slati podatke prema Arduinu.

```
import serial
ser = serial.Serial('/dev/ttyUSB0', 9600)
#ser.write(b'3') slanje
#ser.write(b'5') varijabli
```

2. Faze i rokovi

Datum	Očekivani rezultat
21.03.	Upoznavanje s problematikom.
28.03.	Prvi intervju s korisnicima.
04.04.	Upoznavanje s postojećim rješenjima i mogućnostima
11.04.	Drugi intervju s korisnicima.
18.04.	Izrada projektnog zadatka.
02.05.	Idejno rješenje
16.05.	Izvedbeno rješenje
	Izrada sustava
30.05.	Provjera s korisnicima
	Dovršenje sustava
13.06.	Prezentacija sustava

3. Primopredajni protokol

Izrađeni sustav se predstavlja korisniku radi provjere. U slučaju da korisnik to zatraži, određeni dijelovi sustava se mogu promijeniti u skladu s navedenim rokovima.

Dokumentacija i video uradak se predaju putem wiki stranice projekta: http://pametne-kuce.zesoi.fer.hr/doku.php?id=2019:projekt_petcare .

Sustav se prezentira 13. lipnja 2019. pred nastavnicima i studentima kolegija „Sustavi za praćenje i vođenje procesa“. Nastavnici i studenti će ocijeniti rješenje.