

Sustav za upravljanje uređajima u smart-home okruženju

Marković, Rebeca

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:748178>

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

Download date / Datum preuzimanja: **2024-11-23**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA**

Sveučilišni studij

**SUSTAV ZA UPRAVLJANJE UREĐAJIMA
U SMART-HOME OKRUŽENJU**

Završni rad

Rebeca Marković

Osijek, 2023.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju****Osijek, 06.06.2023.****Odboru za završne i diplomske ispite****Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju**

Ime i prezime Pristupnika:	Rebeca Marković
Studij, smjer:	Sveučilišni prijediplomski studij Elektrotehnika i informacijska tehnologija
Mat. br. Pristupnika, godina upisa:	4771, 23.09.2019.
OIB Pristupnika:	91266383718
Mentor:	izv. prof. dr. sc. Davor Vinko
Sumentor:	,
Sumentor iz tvrtke:	
Naslov završnog rada:	Sustav za upravljanje uređajima u smart-home okruženju
Znanstvena grana rada:	Telekomunikacije i informatika (zn. polje elektrotehnika)
Zadatak završnog rad:	Zadatak završnog rada je izraditi sustav za upravljanje uređajima u smart-home okruženju korištenjem mobilne aplikacije i Arduino razvojne platforme. Tema je rezervirana za: Rebeca Marković
Prijedlog ocjene završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	06.06.2023.
Datum potvrde ocjene od strane Odbora:	08.09.2023.
Potvrda mentora o predaji konačne verzije rada:	<i>Mentor elektronički potpisao predaju konačne verzije.</i>
	Datum:

IZJAVA O ORIGINALNOSTI RADA

Osijek, 08.09.2023.

Ime i prezime studenta:	Rebeca Marković
Studij:	Sveučilišni prijediplomski studij Elektrotehnika i informacijska tehnologija
Mat. br. studenta, godina upisa:	4771, 23.09.2019.
Turnitin podudaranje [%]:	9

Ovom izjavom izjavljujem da je rad pod nazivom: **Sustav za upravljanje uređajima u smart-home okruženju**

izrađen pod vodstvom mentora izv. prof. dr. sc. Davor Vinko

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija. Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD	1
1.1. Zadatak završnog rada	1
2. SMART-HOME SUSTAVI	2
2.1. Područja primjene smart-home sustava	2
2.2. Prednosti i nedostaci	3
2.3. Specifičnosti	4
3. IZRADA SMART-HOME SUSTAVA	4
3.1. Hardware	4
3.1.1. Shema sklopovlja	5
3.1.2. Arduino Uno R3 pločica	6
3.1.3. Četverokanalni relej modul	7
3.1.4. Testna pločica	8
3.1.5. Servo motor SG90	8
3.1.6. Bluetooth modul HC-05	9
3.1.7. LE dioda	9
3.1.8. Vodiči	10
3.2. Software	10
3.2.1. Arduino programski kod	10
3.2.2. Izrada mobilne aplikacije	15
3.2.3. Dizajn mobilne aplikacije	18
3.3. Integracija sustava	18
4. TESTIRANJE	19
5. ZAKLJUČAK	23
LITERATURA	24
SAŽETAK	25

ABSTRACT	26
ŽIVOTOPIS	27
PRILOZI.....	28

1. UVOD

Može se reći da su danas pametni sustavi neizbježni. Tehnologija se svaki dan sve više i brže razvija te je prisutna u svakom aspektu svakodnevnice. Imaju široku primjenu u svakodnevnim životima. Postoje jer čovjeku olakšavaju život, pružaju određenu sigurnost i ubrzavaju izvođenje određenih procesa. Nerijetko su vrlo složeni i skupi. Potražnja za ovakvim sustavima je velika, ponuda na tržištu je ogromna, ali zbog cijene samih uređaja i održavanja cijelog sustava nisu svi u mogućnosti priuštiti si takvu vrstu luksuza. Na internetu postoje mnogi DIY (uradi sam) projekti koji mogu poslužiti za samostalnu izradu vlastitog pametnog sustava, ali za to je potrebno određeno predznanje o elektronici, programiranju i mrežama. Pametni sustavi obuhvaćaju širok spektar područja. Svoju primjenu imaju u kućanstvima, gradovima za regulaciju prometa, u navigacijskim aplikacijama, omogućavaju upravljanje otpadom, nadzor javnih mjesta, prikupljanje solarne energije i razvoja energetske učinkovitosti. Također, koriste se u autoindustriji i industriji općenito, u poljoprivredi za navodnjavanje i praćenje vremenskih prilika te u zdravstvu, trgovini i obrazovnom sustavu.

1.1. Zadatak završnog rada

Zadatak završnog rada je izraditi sustav za upravljanje uređajima u smart-home okruženju korištenjem mobilne aplikacije i Arduino razvojne platforme.

2. SMART-HOME SUSTAVI

Pametni sustavi su sustavi koji povezuju elektroničke uređaje i kućanske uređaje u jedinstvenu mrežu koja omogućava automatizaciju, nadzor i upravljanje određenim uređajima na daljinu putem mobilnih uređaja, web sučelja ili glasovnih naredbi. Omogućavaju korisnicima praćenje događaja u njihovom domu i obavljanje kućanskih poslova dok nisu kod kuće s ciljem olakšavanja svakodnevnice i postizanja ugodnijeg života.[1]

2.1. Područja primjene smart-home sustava

Na tržištu postoje razni pametni sustavi koji obuhvaćaju širok spektar područja. Neki od njih su: kontrola osvjetljenja, upravljanje temperaturom u prostorijama, sigurnost i nadzor doma, upravljanje energetikom, automatizacija kućanskih uređaja te multimedija i zabava. Svim tim sustavima moguće je upravljati na daljinu putem mobilne aplikacije ili imaju sposobnost uključivanja putem senzora te daljinskih priključaka.

Kontrola osvjetljenja je jedna od ključnih funkcionalnosti pametnog sustava. Omogućava korisniku upravljanje rasvjetom na praktičan i prilagodljiv način. Korisnik može uključivati, isključivati ili prigušivati rasvjetu pomoću bežičnih daljinskih upravljača ili mobilne aplikacije. Također može postaviti raspored osvjetljenja koji će se automatski aktivirati u određeno vrijeme. Pomoću *Amazon Alexa* ili *Google Assistant* moguće je davati glasovne naredbe za upravljanje rasvjetom.[2]

Upravljanje temperaturom još je jedna od pogodnosti pametnog sustava. Korisnik može pomoću pametnih termostata i senzora temperature podešavati željenu temperaturu unutar svog doma ili poslovnog prostora. Pomoću pametnih termostata moguće je kontinuirano održavati željenu temperaturu.

Sigurnost i nadzor kućanstva ili poslovnog prostora je važan aspekt pametnog sustava. Pametni uređaji mogu pružiti dodatne razine zaštite i nadzora doma. Neki od uređaja su nadzorne kamere koje omogućavaju nadzor putem video nadzora. Kamere imaju mogućnost snimanja, detekcije pokreta i obavještanja putem mobilne aplikacije. Detektori dima i plina obavještauju putem alarma ili obavijesti na mobilnoj aplikaciji o povišenom nivou plina u prostoriji. Pametne brave omogućavaju daljinsko otključavanje i zaključavanje te pružaju mogućnost generiranja privremenih ili

jednokratnih kodova za pristup. U ovu skupinu pripada i alarmni sustav koji obavještava o mogućim provalnicima ili nepoželjnim aktivnostima.

Postoje i pametni sustavi za praćenje i upravljanje potrošnjom energije u domu kako bi se postigla veća ušteda.

Korisnici također mogu u svoj pametni sustav integrirati i uređaje za zabavu i multimediju poput audio sustava, streaming usluga, pametne televizije itd.

U automatizaciju kućanskih aparata pripadaju svakidašnji kućanski uređaji poput hladnjaka, perilice, sušilice, pećnica, roletna koji također mogu biti povezani na mrežu i biti upravljani putem mobilne aplikacije neovisno gdje se korisnik nalazi.[3]

2.2. Prednosti i nedostaci

Ugradnja pametnog sustava i uređaja u kućanstvo te njihova primjena uvelike olakšava čovjekovu svakodnevnicu, smanjuje stres i brigu te štedi vrijeme i novac. Korisnost pametnog sustava vidljiva je u svakom aspektu čovjekove svakodnevnice. Automatizacija različitih uređaja u domu, poput upravljanja rasvjetom, temperaturom i sigurnosim sustavima povećava udobnost i praktičnost unutar svakog doma. Povećana sigurnost i nadzor kućanstva ili poslovnog prostora putem sigurnosnih kamera, senzora pokreta i sustava za alarm pruža mogućnost trenutnog promatranja situacije te time pravovremeno reagiranje i poduzimanje odgovarajućih mjera za sigurnost i zaštitu. Pomoću pametnih uređaja za grijanje i rasvjetu štedi se energija i smanjuju se svakodnevni životni troškovi. Također, kao najveća prednost pametnih sustava je mogućnost upravljanja cijelim sustavom bežičnim daljinskim ili mobilnom aplikacijom pa čak i onda kada korisnik nije kod kuće. Međutim, osim brojnih prednosti i kvaliteta koje ima pametni sustav, postoje i nedostaci. Kao najveći nedostatak pametnih sustava izdvaja se visoki početni trošak za kupovinu pametnih uređaja i njihovu instalaciju u kućanstvo. Instalacija i konfiguracija pametnog sustava može zahtijevati tehničko znanje, a neki složeniji sustavi mogu biti izazovni za postavljanje i održavanje. Kako bi cijeli sustav funkcionirao, potrebno je imati stabilnu mrežnu povezanost. Mnogi uređaji se povezuju na bežičnu lokalnu mrežu (Wi-Fi), putem Bluetooth tehnologija ili putem bežičnih protokola kao što su Zigbee i Z-Wave. Ponekad može doći do prekida napajanja te to uvelike može narušiti kvalitetu života i stvoriti dodatne probleme. Također, mnoge softvere je potrebno u određenom razdoblju ažurirati i ukomponirati s novim tehnologijama. Potrebno je napomenuti da su prednosti i nedostaci pametnih sustava prvobitno uvjetovani čimbenicima poput

individualnih potreba i preferencija, kvalitete instalacije te pouzdanosti mrežne povezanosti.

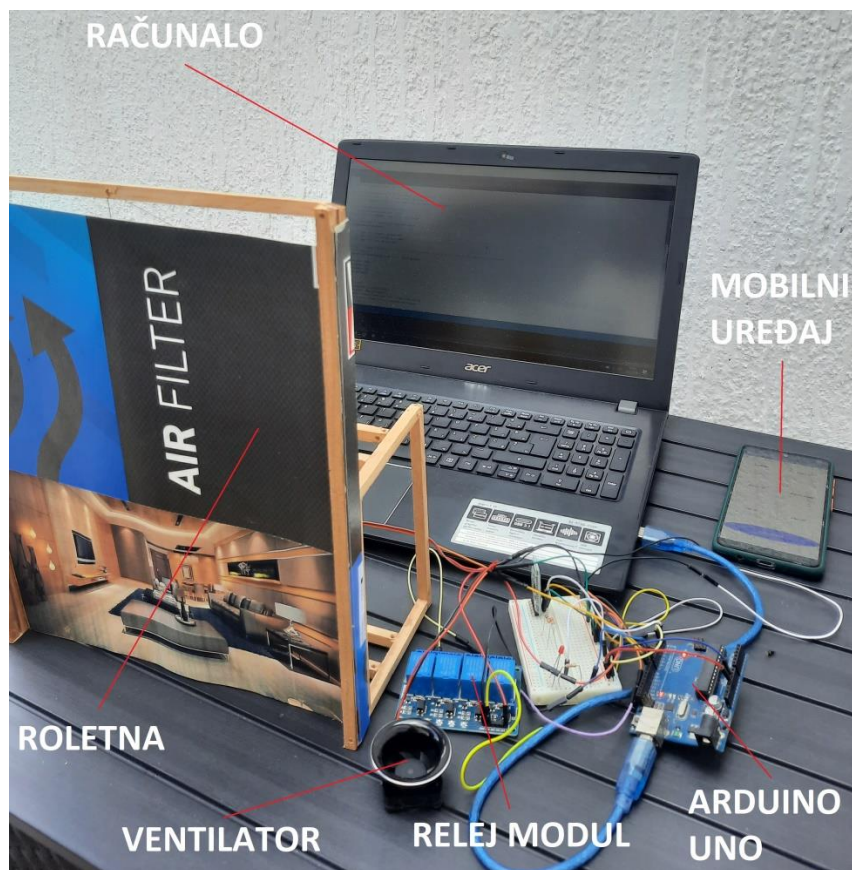
2.3. Specifičnosti

Pametni kućanski sustavi specifični su s obzirom na tradicionalne sustave u kućanstvima. Specifični su po tome što omogućavaju automatizaciju različitih funkcija, odnosno omogućuju da se određene radnje izvrše automatski prema unaprijed postavljenim uvjetima ili rasporedu. Nadalje, pametnim sustavima se upravlja pomoću mobilne aplikacije koja služi za davanje obavijesti ili izvršavanje određenih naredbi definiranih funkcijama u kodu mobilne aplikacije. Također, mnogim pametnim sustavima moguće je upravljati glasovnim naredbama putem popularnih virtualnih asistenata kao što su *Amazon Alexe*, *Google Assistant* ili *Apple Siri*. Osim toga, predviđeni su za praćenje potrošnje energije i pružanje analitike vezane za navike ukućana što korisnicima omogućuje smanjenje troškova te pružanja sigurnosti i nadzora nad kućanstvom. Jedna od ključnih karakteristika pametnih sustava je ta što su prilagodljivi potrebama korisnika. Sve funkcije se mogu prilagoditi načinu života i potrebama korisnika.

3. IZRADA SMART-HOME SUSTAVA

3.1. Hardware

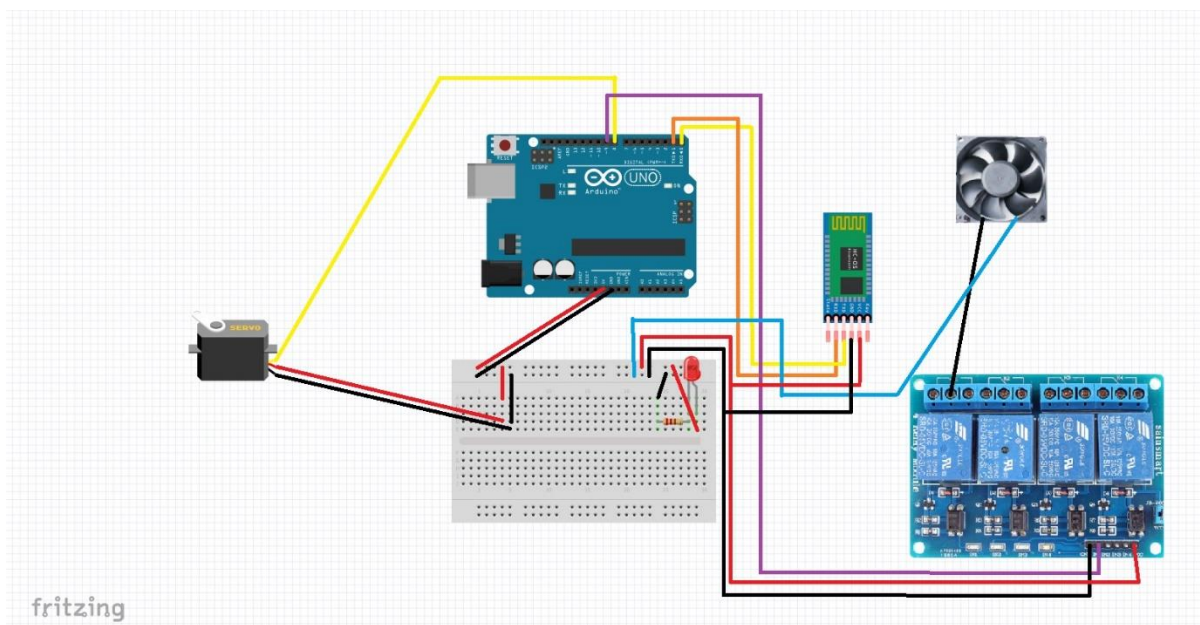
Maketa sustava za upravljanje uređajima u smart-home okruženju sastoji se od: Arduino Uno 3R pločice, testne pločice, servo motora SG90 koji podiže i spušta roletnu, ventilatorom, četverokanalnog Arduino relej modula koji upravlja ventilatorom, roletne, bluetooth modula HC-05 te jedne LE diode. Cijelim sustavom upravlja se pomoću mobilne aplikacije dizajnirane za android operacijski sustav. Predviđena fizička dimenzija smart-home sustava zauzima 30 cm².



Slika 3.1. Dijelovi pametnog sustava

3.1.1. Shema sklopovlja

Cijeli sustav se napaja preko Arduino Uno R3 pločice. Bluetooth HC-05 modul komunicira s Arduinom s ukupno dva pina: RX i TX. LE dioda je spojena serijski s otpornikom od 220 Ohma kako bi se ograničila jakost struje. Releji modul spojen je na digitalni pin 9. Ventilator se uključuje kada relej zatvori strujni krug zavojnice, a isključuje kada je strujni krug zavojnice otvoren. Signal servo motora je upravljani digitalnim pinom 8. Svi korišteni arduino motori, moduli, ventilator i LEDica spojeni su na 5V DC. Shema sklopovlja izrađena je u računalnom programu Fritzing.[4]



Slika 3.2. Shema sklopovlja u Fritzing računalnom programu

3.1.2. Arduino Uno R3 pločica

Arduino Uno R3 programerska pločica je mikroupravljački sustav koju je razvila tvrtka Arduino. Koristi 8-bitni mikrokontroler Atmega328P koji radi na 16MHz. Pločica ima 14 digitalnih ulazno/izlaznih (I/O) pinova i 6 analognih ulaznih pinova. Također, pločica sadrži 6 PWM pinova koji podržavaju PWM (*Pulse Width Modulation*) signal koji se koristi za kontrolu brzine motora i svjetline LE diode. Uno se programira Arduino IDE sketch programom koji je besplatan za Windows, Mac i Linux operativne sustave te ne zahtjeva pristup internetu. Arduino Uno ima ugrađeno USB sučelje preko kojeg se USB B tipom kabela povezuje na računalo.[5]



Slika 3.3. Arduino Uno R3 pločica

Tablica 3.1. Ostali važni tehnički podaci Arduino Uno R3 pločice

Arhitektura	AVR
Ulazni napon	7-12 V
Flash memorija	32KB
SRAM	2KB
Potrošnja struje	50-100mA
Dimenzije pločice	53.4 x 68.6mm
Masa	25g

3.1.3. Četverokanalni relej modul

Četverokanalni relej modul je elektronski modul koji omogućava kontrolu četiri različita uređaja putem releja. Svaki kanal na modulu sadrži relej koji se može uključiti ili isključiti pomoću električnog signala. [6]



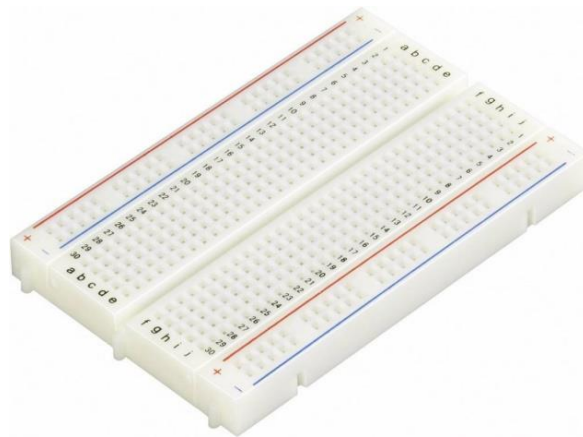
Slika 3.4. Četverokanalni relej modul

Tablica 3.2. Ostali važni tehnički podaci četverokanalnog relej modula

Ulani napon	5V DC
Maksimalna struja	10A
Maksimalni napon	250V AC
Uključenje/isključenje vremena	10ms
Radna temperatura	-40 do +85°C
Dimenzije pločice	75 x 55mm
Masa	30g

3.1.4. Testna pločica

Testna pločica je elektronička ploča koja se koristi za jednostavno povezivanje elektroničkih dijelova bez potrebe za lemljenjem. Pločica sadrži mrežu rupica koje su raspoređene u redove i stupce. Obično se koristi standardni raspored od 2,54 mm između rupica što omogućuje kompatibilnost s većinom standardnih komponenti poput otpornika, kondenzatora, tranzistora, mikrokontrolera itd. Svaka rupica ima metalni kontakt unutar nje. Kontakti su također povezani u redove i stupce te omogućuju povezivanje elektroničkih komponenti. Kontakti u istom redu i u istom stupcu su međusobno povezani. Za napajanje pločice postoje dodatni redovi označeni s + i -. Oni omogućuju priključenje napajanja na pločicu.[7]



Slika 3.5. Testna pločica

3.1.5. Servo motor SG90

Servo motor korišten za podizanje i spuštanje roletne je TowerPro SG90. Napaja se s 5V reguliranog Arduino Uno 3R naponskog izlaza. Ima raspon okretanja 180 stupnjeva (90 stupnjeva sa svake strane neutralne pozicije) i sadrži plastične zupčanike zbog kojih ima tiši rad.[8]



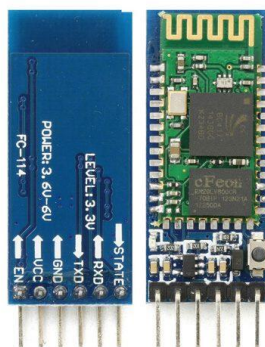
Slika 3.6. TowerPro SG90 servo motor

Tablica 3.3. *Ostali važni tehnički podaci TowerPro SG90 servo motora*

Tip servo motora	mikro servo motor
Moment	1.5 - 1.8 kg/cm
Ulazni signal	PWM signal
Dimenzije	22,60 x 12,40 x 26,60mm

3.1.6. Bluetooth modul HC-05

HC-05 je Bluetooth modul koji se koristi za bežičnu komunikaciju između pametnog telefona i računala. Baziran je na Bluetooth 2.0 specifikaciji koja podržava brzine prijenosa podataka do 2.1 Mbps i ima domet do 10 metara. Radi na naponu od 3.3 V i komunicira s mikrokontrolerom preko serijske komunikacije (UART).[9]



Slika 3.7. *Bluetooth modul HC-05*

3.1.7. LE dioda

LE dioda je elektronička komponenta koja emitira svjetlost kada struja proteče kroz nju u provodnom smjeru. Vrlo su popularne i korištene zbog svoje energetske učinkovitosti, dugog vijeka trajanja (tipično 20 000 – 100 000 sati) i mogućnosti emitiranja svjetlosti u različitim bojama (crvena, zelena, plava, žuta, bijela itd.). Za napajanje koristi radni napon od 1.8 - 3.3 V, iako radni napon ovisi o vrsti LE diode i boji svjetlosti. Potrebna struja je od 5 – 20 mA. Snaga LE diode ovisi o modelu i svjetlosnom intenzitetu. Korištena LE dioda je promjera 3mm.[10]



Slika 3.8. *LE dioda*

3.1.8. Vodiči

Jumper žice su vodiči koji se koriste za privremeno povezivanje elektroničkih komponenti na prototipnoj pločici. Omogućuju jednostavno spajanje i rasklapanje veza između komponenti bez potrebe za lemljenjem. Jumper žice dolaze u različitim duljinama, kao što su 10 cm, 20 cm, 30 cm i slično. Duljina žice se odabire ovisno o potrebnoj udaljenosti između komponenti. Također, dolaze u setovima s različitim bojama kako bi se olakšalo prepoznavanje i organizacija veza. Imaju terminal na svakom kraju. Terminali mogu biti muški ili ženski što omogućuje jednostavno priključivanje na pinove komponenti ili prototipnu pločicu. Izrađene su od provodljivog materijala poput bakra okruženog izolacijskim materijalima poput plastike ili gume.[11]



Slika 3.9. *Jumper žice*

3.2. Software

U ovoj cjelini će biti opisane sve funkcije vezane za Arduino programski kod i kod potreban za izradu mobilne aplikacije namijenjene za android operacijski sustav.

3.2.1. Arduino programski kod

Nakon što se maketa smart-home sustava poveže prema osmišljenoj shemi prema slici 3.1. započinje programiranje. Na računalo je potrebno instalirati program Arduino IDE. Programiranje se vrši isključivo Arduino IDE Sketch programskim jezikom. Kod

se piše u dvije glavne funkcije: setup() i loop(). Setup() funkcija se koristi za postavljanje trenutnih stanja te se izvršava samo jednom na početku prilikom pokretanja programa. Nakon izvršavanja setup() funkcije, izvršava se loop() funkcija u koju se pišu ostale programske naredbe te se ona poziva tempom radnog takta. Iz loop() funkcije moguće je pozivati i druge korisničke napisane funkcije.

U radu je korištena zaglavna biblioteka Servo.h. Navedena biblioteka sadrži sve potrebne funkcije za upravljanje servo motorom koje se pozivaju nad objektom samog servo motora. [12]

```
1 #include <VarSpeedServo.h>
```

Slika 3.10. Deklaracija biblioteke Servo.h

Na početku programa vrši se deklaracija pinova za LE diodu, servo motor i četverokanalni relej modul, deklarira se objekt servo motora i varijabla koja prima string vrijednost iz mobilne aplikacije (numerička vrijednost od 1-6 ovisno o tome koji gumb na mobilnoj aplikaciji korisnik odabere).

```
3 const int redLEDPin = 7; // the number of the LED pin
4 const int servoPin = 8; // the number of servo pin
5 const int relayPin = 9; // the number of the RELAY digital pin
6
7 const int blindsDownPosition = 60;
8 const int blindsUpPosition = 160;
9 const int blindsSpeed = 20; // 0 - 255 range
```

Slika 3.11. Deklaracija LED pina, servo pina i relej pina te deklaracija konstanti za poziciju roletne i brzinu spuštanja/podizanja roletne

Također, vrši se deklaracija konstanti integer tipa za poziciju roletne i brzina za spuštanje/podizanje roletne kako bi se kodom moglo lakše upravljati.

```
11 char deviceId; // Bluetooth received value
12 VarSpeedServo myServo;
```

Slika 3.12. Deklaracija varijable deviceId koja prima string - brojeve od 1-6 i deklaracija objekta servo motora

U glavnoj setup() funkciji vrši se implementacija setupSerial() funkcije koja vrši inicijalizaciju serijske komunikacije u Arduino IDE-u između Arduino Una i računala te postavlja brzinu prijenosa podataka na 9600 baud-a. Osim navedene funkcije,

implementira se `setupLed()` funkcija koja postavlja pin LE diode na digitalni pin 7 i postavlja stanje LOW i `setupServo()` funkcija koja postavlja pin servo motora na digitalni pin 8, postavlja servo motor u poziciju tako da je roletna spuštена i prima varijablu `blindsSpeed` koja predstavlja brzinu kojom će se roletna podizati/spuštati. Također, implementira se funkcija za relej; `setupRelay()` koja postavlja pin releja na digitalni pin 9 i postavlja stanje na HIGH kako bi ventilator prilikom pokretanja programa bio isključen sve dok na relej ne dođe odgovarajući električni signal.

```
40 void setupSerial() {
41   Serial.begin(9600);
42 }
43
44 void setupLed() {
45   pinMode(redLEDPin, OUTPUT);
46   digitalWrite(redLEDPin, LOW);
47 }
48
49 void setupServo() {
50   myServo.attach(servoPin, blindsDownPosition, blindsUpPosition);
51   myServo.slowmove(blindsDownPosition, blindsSpeed);
52 }
53
54 void setupRelay() {
55   pinMode(relayPin, OUTPUT); //attachs the relay on pin 9
56   digitalWrite(relayPin, HIGH); //turns relay off
57 }
```

Slika 3.13. Prikaz funkcija kojima se uređaji prilikom paljenja sustava podešavaju na odgovarajuća stanja i pozicije

Arduino Uno R3 stalno šalje string vrijednost '0' te se implementirana `checkReceivedDeviceId()` funkcija vrti u petlji sve dok korisnik na mobilnoj aplikaciji ne odabere neku funkciju koju želi izvršiti (upaliti/ugasiti LE diodu, podići/spustiti roletnu ili upaliti/ugasiti ventilator). Ukoliko je vrijednost očitana sa `Serial.read()` veća od '0', pohranjuje se u `deviceId` varijablu koja se predaje `checkReceivedDeviceId()` funkciji te se provjeravaju uvjeti u `if` naredbama i sukladno zadovoljenom uvjetu poziva se odgovarajuća funkcija.

```

88 void checkReceivedDeviceId(int deviceId){
89   if (deviceId == '1') {
90     lightsOn();
91   }
92   else if (deviceId == '2') {
93     lightsOff();
94   }
95   else if (deviceId == '3') {
96     blindsUp();
97   }
98   else if (deviceId == '4') {
99     blindsDown();
100  }
101  else if (deviceId == '5') {
102    fanOn();
103  }
104  else if (deviceId == '6') {
105    fanOff();
106  }
107  else {
108  }
109 }

```

Slika 3.14. *Prikaz if-else petlje za odabir primljenog stringa s mobilne aplikacije*

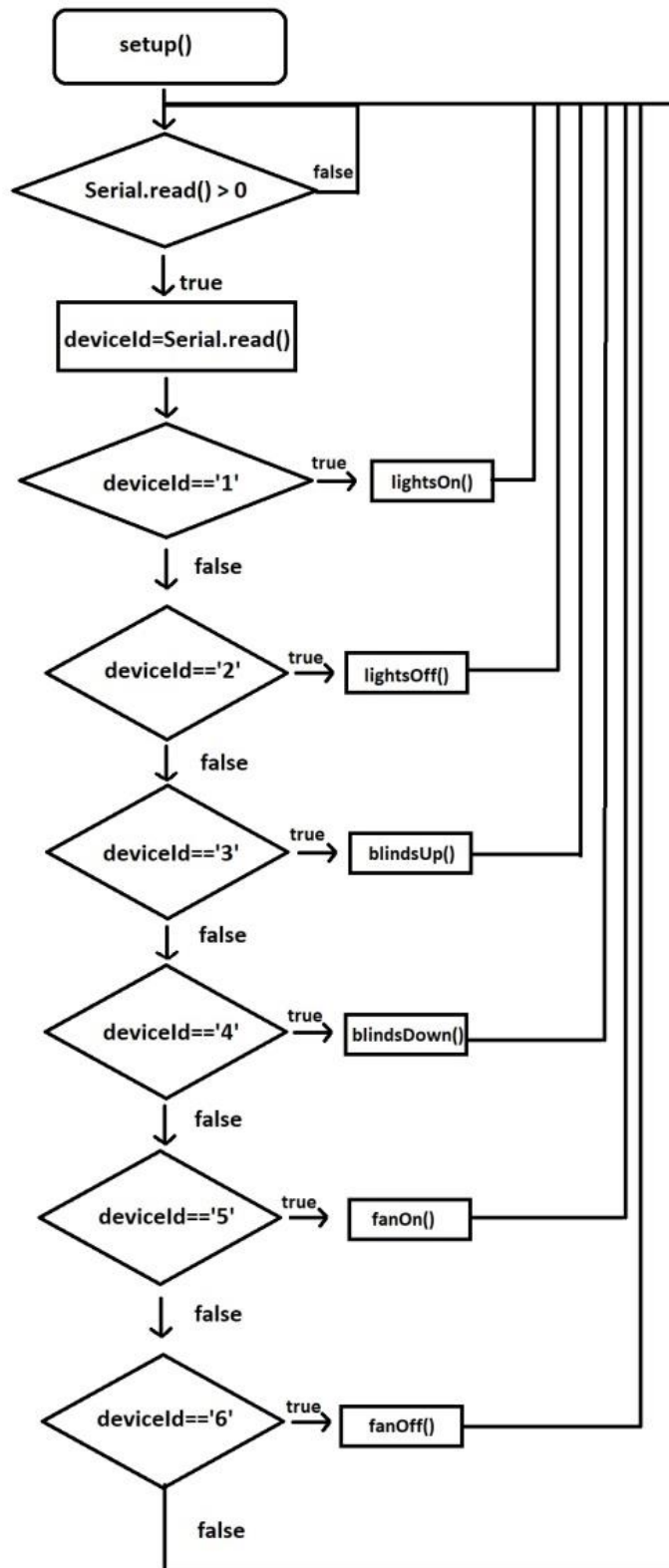
Funkcije za upaliti/ugasiti LE diodu, podići/spustiti roletnu ili upaliti/ugasiti ventilator deklarirane su i implementirane od 60 - 85 linije u kodu.

```

60 void lightsOn() {
61   delay(150);
62   digitalWrite(redLEDPin, HIGH);
63 }
64 void lightsOff() {
65   delay(150);
66   digitalWrite(redLEDPin, LOW);
67 }
68
69 void blindsUp() {
70   delay(150);
71   myServo.slowmove(blindsUpPosition, blindsSpeed);
72 }
73 void blindsDown() {
74   delay(150);
75   myServo.slowmove(blindsDownPosition, blindsSpeed);
76 }
77
78 void fanOn() {
79   delay(150);
80   digitalWrite(relayPin, LOW);
81 }
82 void fanOff() {
83   delay(150);
84   digitalWrite(relayPin, HIGH);
85 }

```

Slika 3.15. *Implementirane funkcije za upravljanje uređajima*



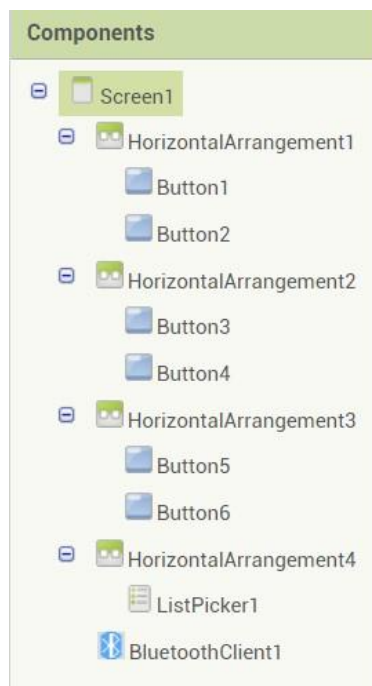
Slika 3.16. Dijagram tijeka smart-home sustava

3.2.2. Izrada mobilne aplikacije

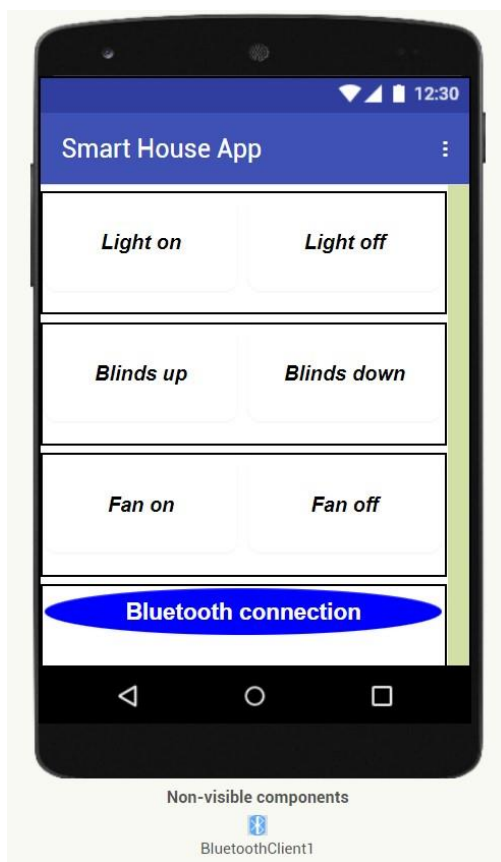
Za izradu mobilne aplikacije korišten je online alat *Massachusetts Institute of Technology* (MIT) App Inventor. MIT App Inventor omogućava dizajniranje i izgradnju mobilnih aplikacija za Android operacijski sustav. Koristi blokovski vizualni jezik programiranja. Sadrži paletu blokova i povlačenje i spuštanje (*drag-and-drop*) funkcionalnosti. Korisnik umjesto pisanja linija koda, sastavlja blokove s određenim naredbama i povezuje ih kako bi definirali funkcionalnost aplikacije. Tijekom izrade aplikacije, korisnici mogu koristiti ugrađeni emulator kako bi u realnom vremenu vidjeli kako će aplikacija izgledati i funkcionirati na Android uređaju. MIT App Inventor sadrži brojne komponente koje mogu korisnicima koristiti za izgradnju aplikacije kao što su: tipke, tekstopisi, slike, senzori, GPS, baza podataka itd. Brojnost komponenti omogućava korisnicima različite funkcionalnosti aplikacija.[13]



Slika 3.17. Prikaz paleta blokova za dizajniranje mobilnih aplikacija



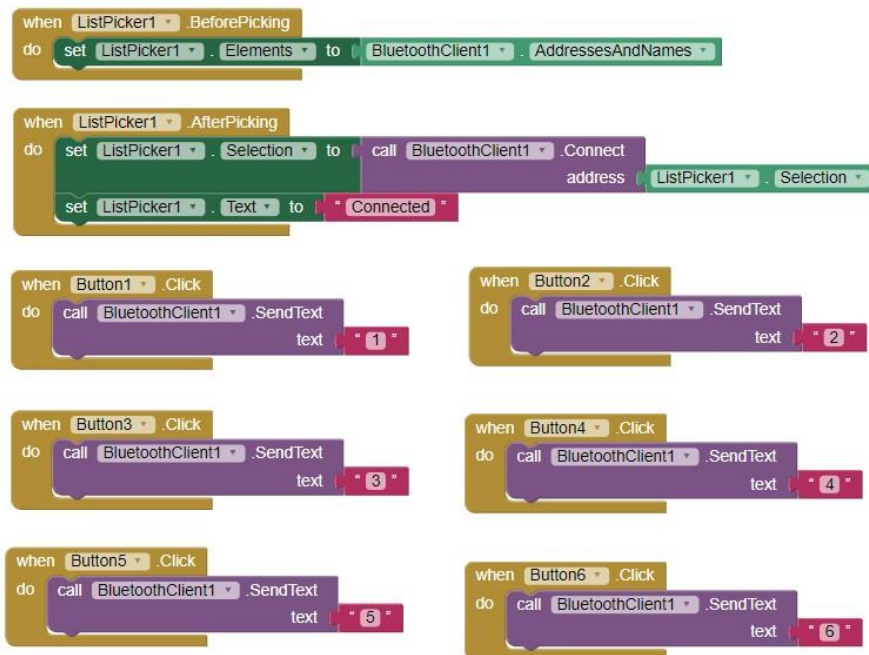
Slika 3.18. Prikaz korištenih komponenti za mobilnu aplikaciju za upravljanje uređajima u smart-home okruženju



Slika 3.19. Prikaz dizajna mobilne aplikacije za upravljanje uređajima u smart-home okruženju



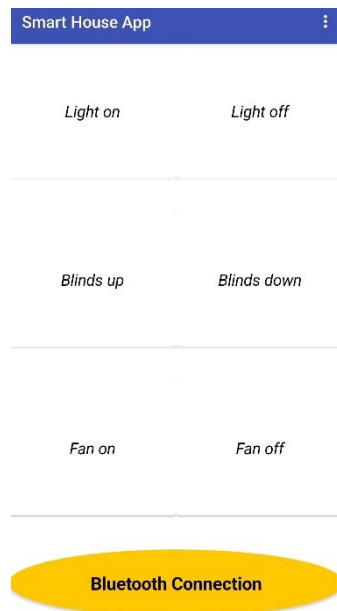
Slika 3.20. Prikaz paleta blokova za izradu funkcionalnosti mobilnih aplikacija



Slika 3.21. Prikaz korištenih blokova naredbi za izradu funkcionalnosti mobilne aplikacije za upravljanje uređajima u smart-home okruženju

3.2.3. Dizajn mobilne aplikacije

Dizajn mobilne aplikacije vrlo je jednostavan. Glavni izbornik mobilne aplikacije sadrži sedam gumbova (*button*). Gumbovi su specificirani za: upaliti/ugasiti LE diodu, podići/spustiti roletnu i upaliti/ugasiti ventilator te sedmi gumb predstavlja opciju za Bluetooth povezanost. Svaki gumb šalje string – broj od 1 do 6 (ovisno koji gumb korisnik odabere) u funkciju u kodu Arduino IDE programa pomoću koje sustav određuje koja naredba će biti izvršena. Broj 1 vrši naredbu za paljenje LE diode, broj 2 vrši naredbu za gašenje LE diode, broj 3 vrši naredbu za podizanje roletne, broj 4 vrši naredbu za spuštanje roletne, broj 5 vrši naredbu za paljenje ventilatora te broj 6 vrši naredbu za gašenje ventilatora.



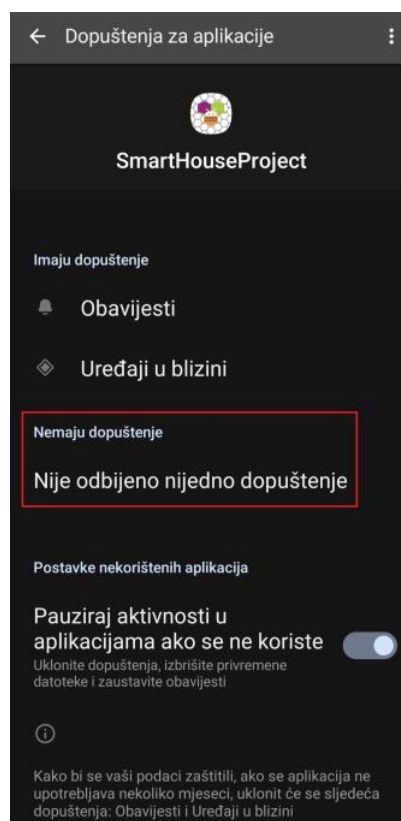
Slika 3.22. Prikaz dizajna mobilne aplikacije Smart House

3.3. Integracija sustava

Kada se napisao programski kod za Arduino platformu odrađeno je učitavanje koda na Arduino kontroler. Također je na Samsung Galaxy A51 pametni uređaj instalirana Android aplikacija te je bila u tijeku integracija sustava. Integracija sustava je završila kada je napravljena i prigodna testna maketa roletne koja je izrađena od 5x5mm drvenih letvica na koju je vijcima pričvršćen servo motor, a ulogu roletne ima ukrasni tanki karton koji se vertikalno giba ovisno o položaju poluge koja je fiksno spojena na ruku servo motora.

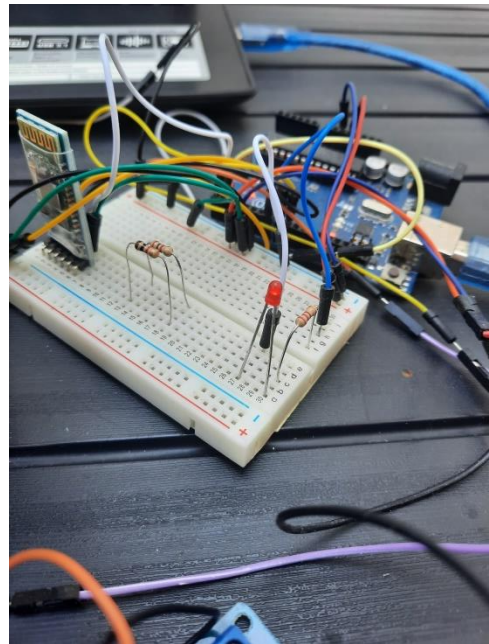
4. TESTIRANJE

Maketa sustava za upravljanje uređajima u smart-home okruženju je napravljena i spojena, Arduino Uno programerska pločica isprogramirana te mobilna aplikacija u potpunosti izrađena pa je napravljeno testiranje pametnog sustava. Da bi se uspješno testirao rad pametnog sustava potrebno je mobilnu aplikaciju instalirati na android uređaj i povezati mobilni uređaj s Bluetoothom HC-05 modulom koji se nalazi na Arduino pločici. Prilikom povezivanja s Bluetooth modulom potrebno je prvobitno u postavkama mobilnog uređaja odobriti dopuštenje za mobilnu aplikaciju, inače se mobilni uređaj neće moći povezati te se sustavom neće moći upravljati.

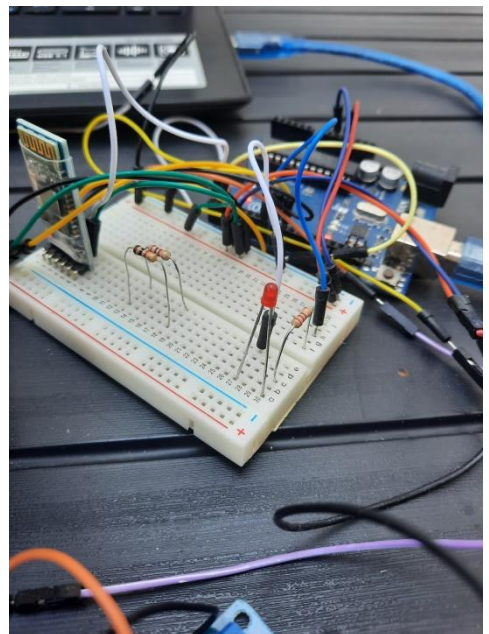
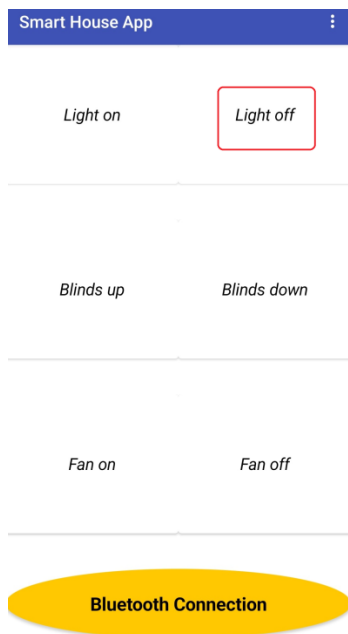


Slika 4.1. Prikaz postavki za davanje dopuštenja android uređaja za povezivanje s Bluetooth HC-05 modulom

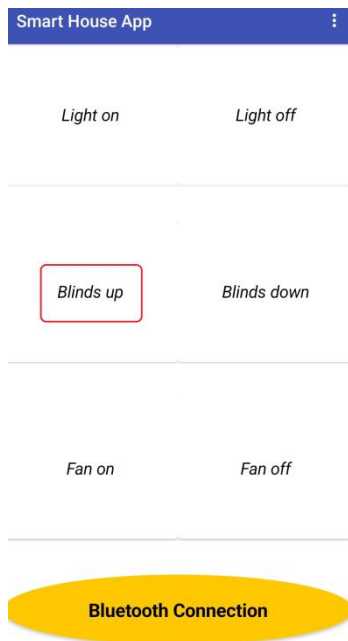
Da bi se uspješno testirao rad pametnog sustava potrebno je USB B tip kabela uključiti u računalo. Rad sustava može se prikazati kroz ukupno 6 koraka:



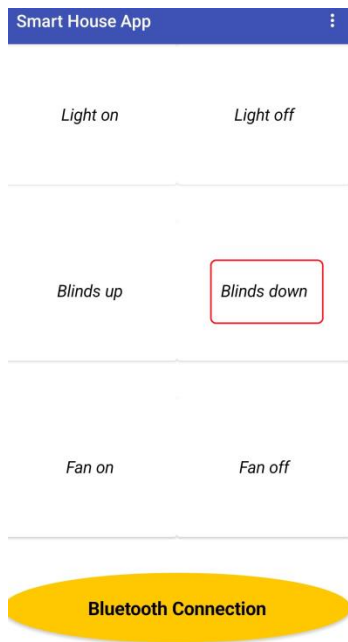
Slika 4.2. Pritisnut je gumb na aplikaciji za paljenje rasvjete, ledica je upaljena.



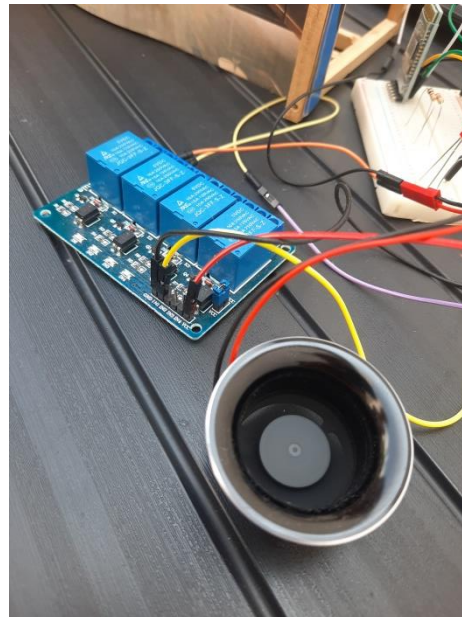
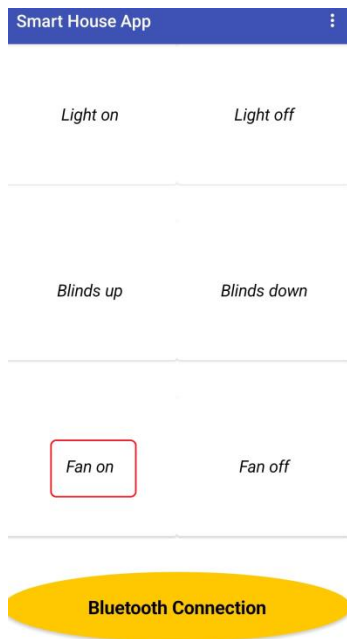
Slika 4.3. Pritisnut je gumb na aplikaciji za gašenje rasvjete, ledica je ugašena.



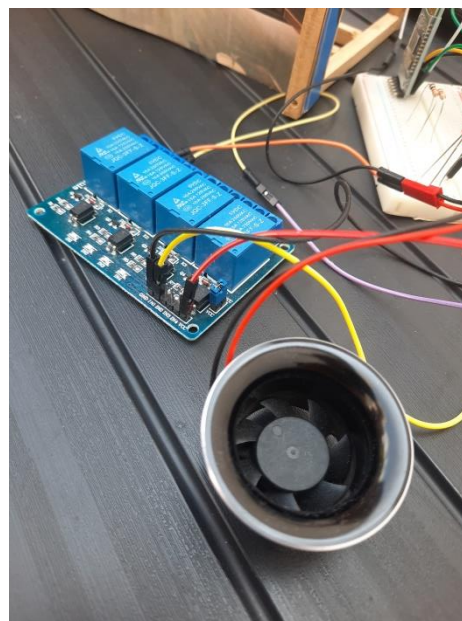
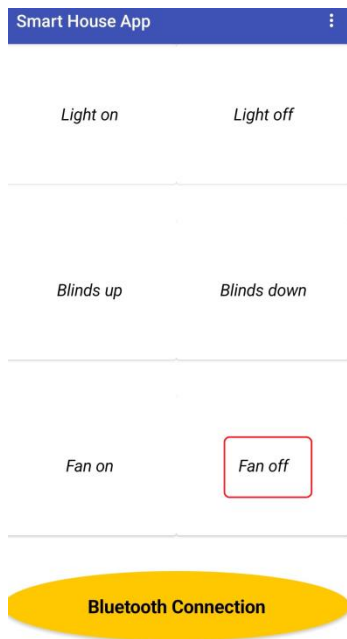
Slika 4.4. Pritisnut je gumb na aplikaciji za podizanje roletne, roletna je podignuta.



Slika 4.5. Pritisnut je gumb na aplikaciji za spuštanje roletne, roletna je spuštена.



Slika 4.6. Pritisnut je gumb na aplikaciji za uključenje ventilatora, ventilator je uključen.



Slika 4.7. Pritisnut je gumb na aplikaciji za isključenje ventilatora, ventilator je isključen.

5. ZAKLJUČAK

Pametni sustav je sustav koji je koristan u više područja te olakšava život korisnicima. U ovom radu izrađen je pametni sustav za paljenje i gašenje rasvjete, podizanje i spuštanje roletne te paljenje i gašenje ventilatora pomoću mobilne aplikacije namijenjene za android operacijski sustav. Opisana je shema spajanja sustava te izrada i dizajn mobilne aplikacije. Sustav je testiran kroz šest koraka kako bi se prikazao njegov rad. Za daljnji napredak ovog sustava može se dodati više kućanskih uređaja kojima bi se također upravljalo android uređajem.

LITERATURA

- [1] “Smart home sustavi,” p. <https://www.nabava.net/clanci/savjeti/smarthome-sto-znaci-imati-pametnu-kucu-1545t6> [Pristupljeno: 20.3.2023.].
- [2] “Alexa vs google home,” p. <https://www.goodhousekeeping.com/uk/product-reviews/tech/a39384401/alexa-vs-google-home/> [Pristupljeno: 20.3.2023.].
- [3] “Amplituda d.o.o.,” p. <https://amplituda.hr/hr/> [Pristupljeno: 20.3.2023.].
- [4] “Fritzing schematics,” p. <https://fritzing.org/> [Pristupljeno: 20.3.2023.].
- [5] “Arduino UnoR3,” p. <https://docs.arduino.cc/hardware/uno-rev3> [Pristupljeno: 20.3.2023.].
- [6] “4ch Arduino Relay Module,” p. http://wiki.sunfounder.cc/index.php?title=4_Channel_5V_Relay_Module [Pristupljeno: 20.3.2023.].
- [7] “Breadboard,” p. <https://www.adafruit.com/product/64>. [Pristupljeno: 20.3.2023.]
- [8] “Servo motor SG90,” p. <https://components101.com/motors/servo-motor-basics-pinout-datasheet> [Pristupljeno: 20.3.2023.].
- [9] “Bluetooth Module HC05,” p. <https://www.electronicwings.com/sensors-modules/bluetooth-module-hc-05-> [Pristupljeno: 20.3.2023.].
- [10] “LED component,” p. https://www.woodvalleyworks.co.uk/index.php?route=product/product&product_id=136 [Pristupljeno: 20.3.2023.].
- [11] “Jumper wires explained,” p. <https://www.wiltronics.com.au/wiltronics-knowledge-base/what-are-jumper-wires/> [Pristupljeno: 20.3.2023.].
- [12] “Servo speed control,” p. <https://forum.arduino.cc/t/varspeedservo-a-modified-servo-library-with-speed-control/61404> [Pristupljeno: 20.3.2023.].
- [13] “MIT App Inventor,” p. <https://appinventor.mit.edu/> [Pristupljeno: 20.3.2023.].

SAŽETAK

U ovom radu izrađen je pametni sustav za upravljanje LE diodom, roletnom i ventilatorom. Sustav se temelji na Arduino Uno razvojnom sustavu. Opisana je shema spajanja sustava te izrada i dizajn mobilne aplikacije. Sustav je testiran kroz šest koraka kako bi se prikazala njegova ispravnost i funkcionalnost. Pametni sustav je koristan sustav, jednostavan za korištenje i olakšava život korisniku.

ABSTRACT

Title: Smart-home system

This bachelor's thesis explains the details of the process of creating a system for managing devices in a smart-home environment using a mobile application and the Arduino development platform. This one the system can be very useful and applicable for every household. This system is developed using Arduino Uno development board. The system consists of one LED diode, blind, fan, one servo motor, four-channel relay module and Bluetooth HC-05 module. The system is powered with a 5V USB type B cable. The result met expectations.

Keywords: Arduino Uno, smart system, mobile application

ŽIVOTOPIS

Rebeca Marković rođena je u Osijeku 6. srpnja 2000. Osnovnu školu pohađala je u Osijeku, a 2015. upisuje Opću gimnaziju u Osijeku. Tijekom školovanja počinje se interesirati za programiranje te 2019. godine upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija.

Potpis autora

PRILOZI

Arduino kod:

```
1 #include <VarSpeedServo.h>
2
3 const int redLEDPin = 7;
4 const int servoPin = 8;
5 const int relayPin = 9;
6
7 const int blindsDownPosition = 60;
8 const int blindsUpPosition = 160;
9 const int blindsSpeed = 20;
10
11 char deviceId;
12 VarSpeedServo myServo;
13
14 // '1' and '2' - lights ON and OFF
15 // '3' and '4' - blinds UP and DOWN
16 // '5' and '6' - fan ON and OFF
17
18 void setup() {
19   setupSerial();
20   setupLed();
21   setupServo();
22   setupRelay();
23 }
24
25 void loop() {
26   while (Serial.available() > 0) {
27     deviceId = Serial.read();
28     delay(30);
29     checkReceivedDeviceId(deviceId);
30   }
31 }
32
33 void setupSerial() {
34   Serial.begin(9600);
35 }
36
37 void setupLed() {
38   pinMode(redLEDPin, OUTPUT);
39   digitalWrite(redLEDPin, LOW);
40 }
41
42 void setupServo() {
43   myServo.attach(servoPin, blindsDownPosition, blindsUpPosition);
44   myServo.slowmove(blindsDownPosition, blindsSpeed);
45 }
46
47 void setupRelay() {
48   pinMode(relayPin, OUTPUT);
49   digitalWrite(relayPin, HIGH);
50 }
51 void lightsOn() {
52   delay(150);
```

```

53  digitalWrite(redLEDPin, HIGH);
54 }
55
56 void lightsOff() {
57  delay(150);
58  digitalWrite(redLEDPin, LOW);
59 }
60
61 void blindsUp(){
62  delay(150);
63  myServo.slowmove(blindsUpPosition, blindsSpeed);
64 }
65 void blindsDown(){
66  delay(150);
67  myServo.slowmove(blindsDownPosition, blindsSpeed);
68 }
69
70 void fanOn(){
71  delay(150);
72  digitalWrite(relayPin, LOW);
73 }
74 void fanOff(){
75  delay(150);
76  digitalWrite(relayPin, HIGH);
77 }
78
79 void checkReceivedDeviceId(int deviceId){
80  if (deviceId == '1') {
81    lightsOn();
82  }
83  else if (deviceId == '2') {
84    lightsOff();
85  }
86  else if (deviceId == '3') {
87    blindsUp();
88  }
89  else if (deviceId == '4') {
90    blindsDown();
91  }
92  else if (deviceId == '5') {
93    fanOn();
94  }
95  else if (deviceId == '6') {
96    fanOff();
97  }
98  else {
99  }
100 }
101
102
103
104
105
106
107
108
109

```

Vizualizacija koda mobilne aplikacije:

```
when ListPicker1 .BeforePicking
do set ListPicker1 .Elements to BluetoothClient1 .AddressesAndNames

when ListPicker1 .AfterPicking
do set ListPicker1 .Selection to call BluetoothClient1 .Connect
address ListPicker1 .Selection
do set ListPicker1 .Text to " Connected "

when Button1 .Click
do call BluetoothClient1 .SendText
text " 1 "

when Button2 .Click
do call BluetoothClient1 .SendText
text " 2 "

when Button3 .Click
do call BluetoothClient1 .SendText
text " 3 "

when Button4 .Click
do call BluetoothClient1 .SendText
text " 4 "

when Button5 .Click
do call BluetoothClient1 .SendText
text " 5 "

when Button6 .Click
do call BluetoothClient1 .SendText
text " 6 "
```