



Srednja šola za kemijo, elektrotehniko in računalništvo

# **UPRAVLJANJE OKENSKIH SENČIL S KRMILNIKOM ARDUINO**

raziskovalna naloga

**Avtor:**

Jure Mazej

R-4a

**Mentor:**

Borut Slemenšek, univ. dipl. inž.

Mestna občina Celje, Mladi za Celje

Celje, marec 2019

## **Zahvala**

Rad bi se zahvalil mentorju Borut Slemenšek, univ. dipl. inž. za usmerjanje in pomoč pri nastajanju naloge.

## Kazalo vsebine

|                                  |    |
|----------------------------------|----|
| Povzetek .....                   | 5  |
| Abstract .....                   | 5  |
| 1    Uvod .....                  | 6  |
| 1.1    Hipoteze .....            | 6  |
| 1.2    Raziskovalne metode ..... | 7  |
| 2    Izdelava naloge .....       | 8  |
| 2.1    Priprava izdelka .....    | 8  |
| .....                            | 9  |
| 2.1.1    Strojna oprema .....    | 10 |
| 2.2    Programiranje .....       | 11 |
| 2.2.1    Arduino .....           | 11 |
| 2.2.2    Android .....           | 14 |
| 2.3    Testiranje .....          | 16 |
| 3    Zaključek .....             | 17 |
| 4    Viri in literatura .....    | 18 |
| 4.1    Viri besedila .....       | 18 |
| 5    Priloga .....               | 19 |
| 5.1    Koda Arduino .....        | 19 |
| 5.2    Koda Android .....        | 30 |

## Kazalo slik

|                                                 |    |
|-------------------------------------------------|----|
| Slika 1: Krmilnik Arduino.....                  | 8  |
| Slika 2: Sprednja stran testnega izdelka. ....  | 9  |
| Slika 3: Prikaz uporabljenih elementov. ....    | 9  |
| Slika 4: Primer kode.....                       | 13 |
| Slika 5: Uporaba metode. ....                   | 13 |
| Slika 6: Prikaz prve strani na telefonu. ....   | 14 |
| Slika 7: Prikaz druge strani na telefonu. ....  | 15 |
| Slika 8: Prikaz tretje strani na telefonu. .... | 15 |

## Kazalo tabel

|                                           |    |
|-------------------------------------------|----|
| Tabela 1: Tabela ukazov WiFi modula. .... | 12 |
| Tabela 2: Prikaz vnesenih urnikov. ....   | 13 |

## **Povzetek**

V zadnjih nekaj letih je v svetu opazen velik napredek pri avtomatizaciji poslovnih stavb in stanovanj. Stavbe postajajo vedno bolj »inteligentne« in uporabnikom prijazne. Tudi sam sem želel nekoliko modernizirati naš dom in sem zato izdelal upravljanje okenskih sečil s krmilnikom Arduino. S krmilnikom upravljam senčila na treh oknih in balkonskih vratih. Posamezno senčilo se avtomatsko dviga in spušča po predhodno nastavljenih dnevni urnikih, ročno s pomočjo tipkal in dodatno še z mobilnim telefonom z uporabo aplikacije za Android.

Skupna dolžina programske kode za aplikacijo na Androidu in Arduinu je približno tisoč vrstic. Program za Arduino je napisan v programskem jeziku različice C oziroma C++, Android studio uporablja za izdelavo programa programski jezik Java, za oblikovanje uporabniškega vmesnika pa uporablja XML programski jezik.

## **Abstract**

Over the past few years, there has been a lot of progress in automation of office buildings and apartments. Buildings are becoming increasingly more "intelligent" and user-friendly. Because of this I also wanted to somewhat modernize our home and decided to manage the blinds with an Arduino controller. I control the blinds on three windows and balcony doors. Individual blind automatically rises and lowers according to previously set daily schedules, manually using switches and via mobile phone using Android app.

The total length of the application code for the Android and Arduino application is about a thousand lines. The Arduino program is written in C or C ++ programming language, while the Android studio uses Java programming language to create the program, and XML programming language is used to create the user interface.

Uporabljene kratice:

URC – ura realnega časa

NC – kontakt releja »Normaly close«

NO – kontakt releja »Normaly open«

Led – Light Emitting Led Diode

## 1 Uvod

V zadnjih nekaj letih se je na trgu pojavilo ogromno izdelkov za avtomatizacijo zgradb. V novejših stavbah je avtomatizirano upravljanje svetil, zunanje razsvetljave, ogrevanja, alarmnih naprav, kamer, porabe energije in seveda tudi senčil. Na žalost je cena naprav za avtomatizacijo precej visoka, oprema pa večkrat tudi ne ustreza vsem željam uporabnika.

Doma sem želel upravljati s tremi okenskimi senčili (roletami) in enim vratnim senčilom. Poleg klasičnega upravljanja s tipkami, sem želel tudi avtomatsko dviganje in spuščanje po dnevnih urnikih (različni časi dvigov in spustov za vsako senčilo posebej) in še upravljanje na daljavo z mobilnim telefonom. Prav tako sem želel imeti možnost samo delnega spuščanja senčila.

Glede na to, da trenutno senčila dvigamo in spuščamo ročno, se bo z avtomatizacijo:

- povečalo udobje bivanja, saj se bodo senčila zjutraj samodejno dvignila in zvečer spustila,
- zmanjšala se bo nevarnost vloma v hišo, saj bodo senčila spreminjala položaj, tudi ko stanovalcev ne bo v hiši (npr. dopust),
- izboljšala se bo klima v hiši (žaluzije bodo ponoči, ko je hladno, spuščene, prav tako se bodo lahko spuščale poleti v času močnejšega sonca).

### 1.1 Hipoteze

Z nalogo sem želel preveriti, kako zanesljive in povezljive so različne elektronske komponente, ki se preko spleta lahko kupijo za malo denarja (krmilnik, WiFi modem, napajalnik, pretvornik 3,3 na 5 V, relejski modul in tipkala).

Prav tako sem želel preizkusiti, ali je možno programsko povezati krmilnik Arduino in mobilno aplikacijo za Android ter zagotoviti hkratno opravljanje senčil preko tipkal, časovnega urnika in daljinskega delovanja preko mobilnega telefona.

Programsko sem želel uporabiti URC (uro realnega časa) v povezavi z vnaprej določenimi urniki dviganja in spuščanja senčil.

## **1.2 Raziskovalne metode**

Pred začetkom naloge sem uporabil metodo analize dokumentov, s katero sem pregledal dokumentacijo in forume za Arduino Zero. Prav tako sem to metodo uporabljal za vsako komponento, ki sem jo uporabil v nalogi, da sem ugotovil, kako komponento pravilno povezati in vključiti v nalogo.

Metodo eksperimenta sem uporabljal skozi celotno nalogo, najbolj koristna pa je bila pri končnem testiranju izdelka.

## 2 Izdelava naloge

### 2.1 Priprava izdelka

Za pogon senčil je potrebno v obstoječi mehanizem senčil vgraditi elektromotorni pogon (230 V, 50 Hz). Takšni pogoni že imajo vgrajena končna stikala, ki omejujejo prekomerni dvig ali spust senčila.

Za krmiljenje sem izbral krmilnik Arduino Zero. Na ta krmilnik je mogoče priključiti 20 digitalnih izhodov ali vhodov. Vhod ali izhod se določi programsko. Ker imajo izhodi majhne moči (7 mA) sem za krmiljenje motorjev senčil uporabil relejski modul z osmimi preklopnimi releji (230 V, 5 A). Glede na to, da ima krmilnih 3,3 V izhode, relejski modul pa 5 V tuljave, sem moral uporabiti še napetostni pretvornik 3,3 V / 5 V. Za WiFi modem sem uporabil modul ESP 2866 12E. Za prikaz dvigovanja in spuščanja senčil so v testni napravi vgrajene LED diode (namesto elektromotorne pogona). Za stikala sem uporabil tipkala za senčila proizvajalca TEM Čatež.

Zaradi lažje predstavitve in testiranja je vsa oprema, vgrajena na nosilec, iz belega pleksi stekla.

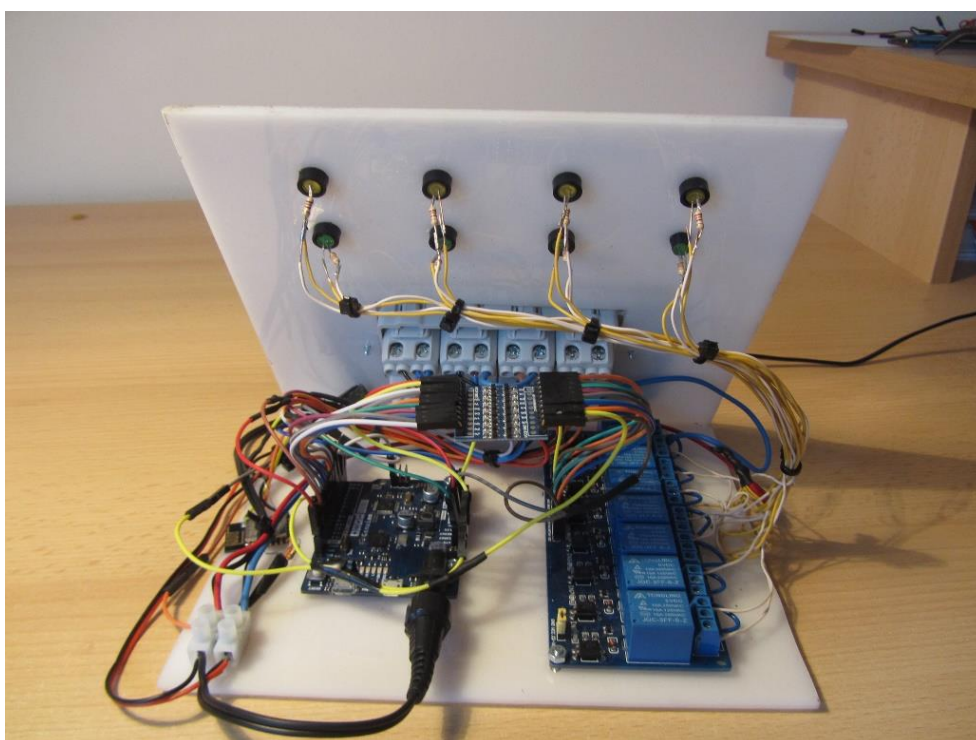
Vsa uporabljena oprema cenovno ne presega 60 EUR (krmilnik Arduino 40 EUR), kar je bistveno manj od primerljivih izdelkov na trgu.



Slika 1: Krmilnik Arduino.



Slika 2: Sprednja stran testnega izdelka.



Slika 3: Prikaz uporabljenih elementov.

### **2. 1. 1 Strojna oprema**

Pred izdelavo programa sem moral po internetu naročiti kompatibilne naprave, ki bodo zadostovale ciljem moje naloge.

Na začetku sem izbral Arduino krmilnik Arduino Zero, saj ima zadostno število digitalnih vhodov/izhodov, prav tako ima tudi URT, ki sem jo želel za uporabo avtomatskega delovanja senčil po časovnem urniku. Problem tega krmilnika je, da deluje s 3.3 V in ker na spletu nisem našel relejskega modula, ki deluje z napetostjo tuljave 3.3 V sem moral kupiti tudi pretvornik napetosti 3.3 V na 5 V.

Za aktuator sem na internetu našel 8-kanalni relejski modul, ki ga sedaj tudi uporabljam. Na delovne kontakte (NO) relejskega modula sem priključil led diode, s katerimi prikazujem dviganje ali spuščanje senčila.

Tipkalo, ki sem ga uporabil v nalogi, je povezano na vhode krmilnika Arduino. S tipkalom v ročnem režimu delovanja spuščam ali dvigam senčila.

Na koncu sem kupil še WiFi modul ESP 8266 12E. Napajalna napetost modula je 3.3 V. Zaradi njegove velike porabe ga nisem mogel priključiti na 3.3 V vir krmilnika Arduino, ampak sem uporabil 5 V napetost iz napajalnika. Napetost 5 V sam pretvoril na 3.3 V z napetostnim pretvornikom.

Za povezovanje posameznih komponent sem uporabljal že tovarniško izdelane povezovale žice dolžine 10 centimetrov. Povezovale žice so že zaključene z ženskimi in moškimi konektorji. Za povezovanje led diod, WiFi modula in napajalnega pretvornika pa sem uporabil žice preseka 0.5 mm<sup>2</sup>, ki sem jih povezal s spajkanjem.

## 2.2 Programiranje

S krmilnikom bom upravljal tri okenska senčila (rolete) in eno senčilo za vrata. Poleg klasičnega upravljanja s tipkami je omogočeno tudi avtomatsko dviganje in spuščanje po dnevnih urnikih. Obstaja možnost nastavitve časa za vsako senčilo posebej, in to ločeno za vsak dan v tednu. Vsak dan je za vsako senčilo možno nastaviti dvig in spust ob različnih časih. V primeru odsotnosti je omogočeno še upravljanje na daljavo preko mobilnega telefona. Pri upravljanju s telefonom se lahko preko drsnika senčilo spusti popolnoma ali samo delno. Enako velja za dvigovanje.

Pri spuščanju po urnikih je možno vpisati tudi čas spuščanja. S tem se lahko posamezna žaluzija spusti samo delno.

Pri testiranju delovanja sem predvidel, da je čas spuščanja (in dvigovanja) senčila 7 sekund. Dejanski čas, potreben za dvig ali spust, bom izmeril, ko bo vgrajen motorni pogon senčila. S programom preračunavam, koliko časa se je posamezno senčilo spuščalo ali dvigovalo, zato je trenutni položaj senčila vedno znan. Pri pisanju programa sem pazil tudi na to, da ne bi prišlo do hkratnega ukaza za dvig in spust istega senčila, saj bi na primer pri spuščanju senčila po urniku lahko prišlo tudi do ročne zahteve za dvig. V kolikor to programsko ne bi bilo preprečeno, bi lahko prišlo do uničenja elektromotorne pogona senčila.

V primeru izpada električnega omrežja upravljanje senčil seveda ne deluje. Po vrnitvi omrežne napetosti se krmilnik inicializira, vsa senčila pa se postavijo v dvignjeni položaj. Po nastavitvi URC se senčila premaknejo v položaj, predviden s trenutnim časovnim programom.

Pri programiranju sem skupaj z Android in Arduino aplikacijo napisal približno tisoč vrstic kode. Celotna aplikacija, ki sestavlja to nalogo je, razdeljena na dva dela, in sicer na del z Arduinom, preko katerega gredo vsi ukazi, in del z Androidom, ki je aplikacijski vmesnik uporabniku. Program za Arduino je napisan v programskem jeziku različice C oziroma C++, Android studio za izdelavo programa uporablja programski jezik Java, za oblikovanje uporabniškega vmesnika pa se uporablja XML programski jezik.

### 2.2.1 Arduino

Delovanje programa v krmilniku lahko razdelimo na štiri dele. Prvi del je inicializacija, ki se izvede samo prvič. Tu najprej določim spremenljivke, ki bodo uporabljene v glavnem programu in metodah, prav tako nastavim uro realnega časa. V »setup« delu nastavim hitrost prenosa

podatkov v bitih na sekundo za prenos zaporednih podatkov. Arduino deluje z 9600 biti na sekundo, WiFi modul pa deluje s 115200 biti na sekundo preko digitalnega vhoda 0 in izhoda 1.

Z ukazom »pinMode« sem nastavil 8 vhodov in 8 izhodov, ki jih bom tudi uporabljal.

Za pravilno delovanje WiFi modula je potrebno nastaviti pravi način uporabe. Modul nudi tri načine delovanja in sicer kot strežnik, kot vhodna točka ali pa oboje. Jaz sem uporabil zadnji način delovanja. Z ukazom »CIPSERVER« ustvarimo server na vratih 80, z ukazom »CWJAP« pa se povežemo z internetom.

Tabela 1: Tabela ukazov WiFi modula.

| Basic   | WiFi layer   | TCPIP Layer  |
|---------|--------------|--------------|
| AT      | AT+CWMODE    | AT+CIPSTATUS |
| AT+RST  | AT+CWJAP     | AT+CIPSTART  |
| AT+GMR  | AT+CWLAP     | AT+CIPSEND   |
| AT+GSLP | AT+CWQAP     | AT+CIPCLOSE  |
| ATE     | AT+CWSAP     | AT+CIFSR     |
|         | AT+CWLIF     | AT+CIPMUX    |
|         | AT+CWDHCP    | AT+CIPSERVER |
|         | AT+CIPSTAMAC | AT+CIPMODE   |
|         | AT+CIPAPMAC  | AT+CIPSTO    |
|         | AT+CIPSTA    | AT+CIUPDATE  |
|         | AT+CIPAP     | +IPD         |

Po končani inicializaciji sem nadaljeval programiranje z drugim delom programa. Drugi del je programiranje delovanje senčil preko tipk. Po pritisku tipke za določeno senčilo program najprej preveri položaj senčila. Če je s tipko zahtevano spuščanje senčila in senčilo še ni v spodnji legi, se vključi rele za spuščanje senčila. Če je senčilo že v spodnji legi, se rele za spuščanje ne vključi. Rele za spuščanje je vključen dokler pritiskamo tipko za spuščanje. Ko se senčilo pomakne v spodnjo lego, se rele izključi. Položaj senčila ugotavljam na podlagi seštevka časa ukazov za spuščane ali dviganje posameznega senčila.

Tretji del glavnega programa je avtomatski dvig in spust senčil. Ta deluje na podlagi ure realnega časa. Program preverja nastavljene čase (tabela 1) in v skladu z njimi dviga oziroma spušča senčila.

Tabela 2: Prikaz vnesenih urnikov.

|           | Žaluzija 1 |       | Žaluzija 2 |       | Žaluzija 3 |       | Žaluzija 4 |       |
|-----------|------------|-------|------------|-------|------------|-------|------------|-------|
| Delavniki | gor        | 6:30  | gor        | 6:30  | gor        | 6:00  | gor        | 6:00  |
|           | dol        | 22:00 | dol        | 22:00 | dol        | 22:15 | dol        | 22:00 |
| Vikend    | gor        | 7:30  | gor        | 7:30  | gor        | 7:00  | gor        | 8:00  |
|           | dol        | 21:30 | dol        | 21:30 | dol        | 22:00 | dol        | 21:00 |

Četrty del programa je namenjen WiFi modulu. Krmilnik Arduino preko WiFi modula komunicira z Android aplikacijo na mobilnem telefonu. Če preko mobilnega telefona pošljemo zahtevo z android aplikacije, to vidimo s »Serial1.find("+IPD, ")«. Po tem ugotovim, katero senčilo bo program spuščal, in za koliko odstotkov.

```
if (Serial1.find("+IPD, "))
{
    espRead();
}
```

Slika 4: Primer kode

Z metodo, ki vrne »true«, če pravilno deluje, in »false«, če ne, ugotovimo, ali se bo glede na odstotke, oziroma trenutno pozicijo senčilo spuščalo ali dvigalo.

```
static bool checkState(int pin) {
    if (stanjeRolet (getOdstetiPin (pin), getStanjeRolet (pin), spust_do_konca) > procent) {
        setPinSpust (pin, procent);
        return true;
    }
    else {
        setPinDvig (pin, procent);
        return true;
    }
}
return false;
}
```

Slika 5: Uporaba metode.

Po končanem premiku senčila krmilnik pošlje potrditev na Android aplikacijo. To naredimo s »CIPSEND«, ki sporoči dolžino podatkov (char), ki jih bomo poslali.

Vsi deli glavnega programa so medsebojno povezani, tako da vsi računajo odstotek spuščенosti senčila. V kolikor je senčilo spuščeno do konca, se ukaz (rele) za spuščanje izključi. Odstotek se računa glede na čas delovanja dviga oziroma spusta senčila. Potreben čas za dvig oziroma spust posameznega senčila bo izmerjen s štoparico, ko bo vgrajen elektromotorni pogon za posamezno senčilo. Trenutno pa je v programu nastavljen čas 7 sekund.

## 2. 2. 2 Android

Pri aplikaciji Android Studio sem moral najprej poskrbeti, da bo imela dovoljenje za dostop do interneta. To sem naredil v »android manifest« datoteki.

Aplikacija ima tri različne strani.



Prva stran vsebuje dva »EditText«, v katere vpišemo dodeljeno uporabniško ime in geslo. Ob pritisku na gumb aplikacija naprej pošlje vpisano uporabniško ime in geslo na Arduino, kjer preverjam, če sta skladna z uporabniškim imenom in geslom, shranjenim v Arduino. Geslo in uporabniško ime uporabljam zaradi preprečitve neželenih dostopov do uporabe aplikacije. Potem se, tudi s pritiskom na gumb, ne pomaknemo na drugo stran aplikacije.

Slika 6: Prikaz prve strani na telefonu.



Slika 7: Prikaz druge strani na telefonu.

Na drugi strani se nahajajo štiri gumbi, s katerimi izberemo, katero senčilo bomo pomikali. Ob kliku na gumb se na zadnjo stran prenese IP, vrata in številka senčila, ki ga želimo upravljati.



Slika 8: Prikaz tretje strani na telefonu.

Na zadnji strani je »SeekBar«, to je drsnik, s katerim nastavimo želeno dolžino spusta. Tam je tudi »TextView«, s katerim ugotovimo točno izbrano dolžino drsnika. Zadnji element na strani je gumb, s katerim potrdimo izbiro in sprožimo ukaz za pomik izbranega senčila. Za lepši izgled drsnika sem uporabil knjižnico »ArcSeekBar«, ki omogoča zakrivljen drsnik. Knjižnico sem našel na spletni strani GitHuba. Na drsniku se spreminja tudi barva.

Na zadnji strani sem moral dodati še dve knjižnici, ki sem jih naložil z interneta ter ju dodal v datoteko »libs«. Knjižnici, ki sem ju dodal, sta »apache-httpcomponents« in »httpclient«, uporabil pa sem ju zaradi funkcij, ki jih ponujata.

## 2.3 Testiranje

Po končanem programiranju sem pričel z zaključnim testiranjem delovanja. Najprej sem testiral ročno delovanje. S pritiskom na tipko za dvig in spust vsakega senčila sem preizkusil, ali pritisku tipke ustreza vklop ustrezne LED diode. Nato je sledilo preizkušanje časovnega delovanja preko urnikov za posamezni dan. Pri testiranju sem vpisal željeni čas in opazoval, ali se vključujejo ustrezne LED. Na koncu je sledilo še testiranje z mobilnim telefonom.

Po končanem testiranju aplikacija deluje brez napak. Seveda pa vse ni potekalo brez težav. V nadaljevanju navajam nekaj problemov, ki sem jih moral rešiti.

Prva težava, ki sem jo imel, je bila, da so bili na relejskem modulu narobe povezani kontakti. Zamenjal sem NO in NC kontakt. Zaradi tega je pri izklopljenem releju svetila led dioda.

Težavo sem imel tudi pri priključitvi WiFi modula, ker potrebuje dva digitalna vhoda/izhoda TX in RX. Na ta digitalna vhoda sem že imel priključena dva signala iz tipk. Zaradi tega sem moral tipke priključiti na dva druga prosta vhoda in spremeniti tudi program. S tem sem vhoda TX in RX krmilnika sprostil za priključitev WiFi modula.

Pri začetnem testiranju WiFi modula sem imel veliko težav, saj se ta modul večinoma uporablja kot samostojen krmilnik, kar je pomenilo, da knjižnice, ki je bila temu namenjena, nisem mogel uporabiti. Problem sem rešil tako, da posamezen ukaz pošljem WiFi modulu preko krmilnika.

Pri sinhronizaciji treh glavnih delov programa (na tipke, avtomatsko delovanje in Android aplikacija) sem imel težavo z računanjem odstotkov položaja senčila, saj sem pri tem uporabljal URC. Ugotovil sem, da zaradi uporabe ure realnega časa metoda ne deluje in sem moral uporabiti metodo `millis()` – časovnik.

Tudi pri Android Studiu sem naletel na nekaj težav. Za prenos podatkov med Androidom in Arduinom sem moral dodati primerne knjižnice, ki jih je bilo težko najti.

Opazil sem tudi, da je pri pošiljanju podatkov s krmilnika odrezalo zadnjo poslano črko. To sem rešil tako, da sem na konec podatkov, ki sem jih želel poslati, dodal presledek.

### 3 Zaključek

Uspelo mi je narediti izdelek, s katerim bomo lahko v naši hiši upravljali s štirimi senčili na južni strani stavbe. S testiranjem sem ugotovil, da program in vse komponente delujejo pravilno in zanesljivo. Po končanih predstavitvah ga bomo vgradili v elektro omarico in povezali z elektromotornimi pogoni senčil.

Med testiranjem, ki je potekalo več dni, so vse komponente delovale zanesljivo. Izdelek sem preizkušal pri sobni temperaturi in tudi pri nižjih temperaturah (cca. 6 °C), pri čemer nisem zaznal sprememb v zanesljivosti delovanja. Ker bo izdelek po vgradnji v uporabi daljši čas, bom lahko preizkusil tudi zanesljivost vgrajene opreme v daljšem časovnem obdobju. V kolikor bo prihajalo do okvar ali nezanesljivega delovanja, bo potrebno vgraditi nekoliko dražji krmilnik in ostalo opremo. Zaenkrat pa lahko hipotezo o zanesljivosti različnih elektronskih komponent nižjega cenovnega razreda potrdim.

Z izdelavo naloge sem potrdil tudi hipotezo, da je možno programsko povezati krmilnik Arduino in mobilno aplikacijo za Android ter zagotoviti hkratno opravljanje senčil preko tipkal, časovnega urnika in daljinskega delovanja preko mobilnega telefona.

Tretjo hipotezo o uporabi URC lahko samo deloma potrdim. Uro realnega časa je možno uporabljati za dvigovanje in spuščanje senčil v vnaprej določenih časovnih urnikih, problem pa se pojavlja pri izpadu napajanja, ko se URC resetira, ker nima rezervnega napajanja (kot na primer pri računalnikih).

V prihodnosti se bom lotil povezovanja vremenskih podatkov interneta in krmilnika. Želel bi uporabiti podatke o trenutnem vremenu v Celju za krmiljenje senčil (npr. če je temperatura visoka in sije sonce, bi se senčila spustila).

## 4 Viri in literatura

### 4.1 Viri besedila

*Android Developer.* (1. 12 2018). Pridobljeno iz Android Developer:

<https://developer.android.com/reference/android/os/AsyncTask>

*Arduino forum.* (22. 10 2018). Pridobljeno iz Arduino:

<https://forum.arduino.cc/index.php?topic=383057.0>

*Arduino Store.* (5. 11 2018). Pridobljeno iz Arduino: <https://store.arduino.cc/genuino-zero>

*ESP8266 - AT Command Reference.* (5. 11 2018). Pridobljeno iz room-15: <https://room-15.github.io/blog/2015/03/26/esp8266-at-command-reference/#AT+CIPSEND>

*How to use an ESP8266 in the Arduino IDE.* (12. 11 2018). Pridobljeno iz CORE Electronics:

<https://core-electronics.com.au/tutorials/using-esp8266-with-arduino-ide.html>

*M2MSupport.net.* (27. 10 2018). Pridobljeno iz M2MSupport.net:

<https://m2msupport.net/m2msupport/atcifsr-get-local-ip-address/>

*Project Hub.* (12. 11 2018). Pridobljeno iz Arduino:

<https://create.arduino.cc/projecthub/circuito-io-team/simple-arduino-uno-esp-8266-integration-dba10b>

*Stack overflow.* (4. 11 2018). Pridobljeno iz Stack overflow:

<https://stackoverflow.com/questions/2091465/how-do-i-pass-data-between-activities-in-android-application>

## 5 Priloga

### 5.1 Koda Arduino

```
#include <RTCZero.h>
RTCZero rtc;
#define DEBUG true
int procent;
double spust_do_konca[] {7,7,7,7}; //v sekundah
int state_kuhna_s; //6,7
int state_kuhna_b; //8,9
int state_dnevna_v; //10,11
int state_dnevna_o; //12,13

//////////////////////////////////NASTAVITVE//////////////////////////////////
int cas_dviga_kuhinja_stopnisce = 20;
int cas_dviga_kuhinja_stopnisce_m = 30;
int cas_dviga_kuhinja_balkon = 20;
int cas_dviga_kuhinja_balkon_m =30;
int cas_dviga_dnevna_vrata = 20;
int cas_dviga_dnevna_vrata_m =30;
int cas_dviga_dnevna_okno = 20;
int cas_dviga_dnevna_okno_m = 30;
int trajanje_dviga =7;

int cas_spusta_kuhinja_stopnisce = 21;
int cas_spusta_kuhinja_stopnisce_m = 30;
int cas_spusta_kuhinja_balkon = 21;
int cas_spusta_kuhinja_balkon_m = 30;
int cas_spusta_dnevna_vrata = 21;
int cas_spusta_dnevna_vrata_m = 30;
int cas_spusta_dnevna_okno = 21;
int cas_spusta_dnevna_okno_m = 30;

int stetje_dnevov = 0;

void spremeniVikend(){ //sprememba urnika med vikendom
  cas_dviga_kuhinja_stopnisce = 7;
  cas_dviga_kuhinja_stopnisce_m = 30;
  cas_dviga_kuhinja_balkon = 7;
  cas_dviga_kuhinja_balkon_m =30;
  cas_dviga_dnevna_vrata = 7;
  cas_dviga_dnevna_vrata_m = 0;
  cas_dviga_dnevna_okno = 8;
  cas_dviga_dnevna_okno_m = 0;
  trajanje_dviga =1;

  cas_spusta_kuhinja_stopnisce = 21;
  cas_spusta_kuhinja_stopnisce_m = 30;
  cas_spusta_kuhinja_balkon = 21;
  cas_spusta_kuhinja_balkon_m = 30;
  cas_spusta_dnevna_vrata = 22;
  cas_spusta_dnevna_vrata_m = 0;
  cas_spusta_dnevna_okno = 21;
  cas_spusta_dnevna_okno_m = 0;
}

void spremeniTedn(){ //sprememba urnika med tednom
  cas_dviga_kuhinja_stopnisce = 6;
```

```
cas_dviga_kuhinja_stopnisce_m = 30;
cas_dviga_kuhinja_balkon = 6;
cas_dviga_kuhinja_balkon_m = 30;
cas_dviga_dnevna_vrata = 6;
cas_dviga_dnevna_vrata_m = 0;
cas_dviga_dnevna_okno = 6;
cas_dviga_dnevna_okno_m = 0;
trajanje_dviga = 1;

cas_spusta_kuhinja_stopnisce = 22;
cas_spusta_kuhinja_stopnisce_m = 0;
cas_spusta_kuhinja_balkon = 22;
cas_spusta_kuhinja_balkon_m = 0;
cas_spusta_dnevna_vrata = 22;
cas_spusta_dnevna_vrata_m = 15;
cas_spusta_dnevna_okno = 22;
cas_spusta_dnevna_okno_m = 0;
}

////////////////////////////////////

int val_20 = 0; //vrednost pritiska
int val_21 = 0;
int val_2 = 0;
int val_3 = 0;
int val_4 = 0;
int val_5 = 0;
int val_22 = 0;
int val_38 = 0;

//////////////////////////////////NASTAVITVE URE//////////////////////////////////
const byte seconds = 0;
const byte minutes = 59;
const byte hours = 23;

const byte day = 9;
const byte month = 2;
const byte year = 18;

int sec = 0;

bool done_1 = false;
bool done_2 = false;
bool done_3 = false;
bool done_4 = false;

bool done_5 = false;
bool done_6 = false;
bool done_7 = false;
bool done_8 = false;

void setup() {
    Serial.begin(9600);
    Serial1.begin(115200);
    //-----RTC-----
    rtc.begin();
    rtc.setTime(hours, minutes, seconds);
    rtc.setDate(day, month, year);
}
```

```
//-----OUTPUT-----
for(int i=6; i<=13; i++){//HIGH==LOW
    pinMode(i, OUTPUT);
}

//-----INPUT-----
pinMode(38, INPUT); //analog pin 0-1023; 0-10=HIGH; A0==15
pinMode(22, INPUT); //HIGH==HIGH
pinMode(20, INPUT);
pinMode(21, INPUT);
pinMode(2, INPUT);
pinMode(3, INPUT);
pinMode(4, INPUT);
pinMode(5, INPUT);

//-----MODE-----

for(int i=0; i<=13; i++){
    digitalWrite(i, HIGH); //HIGH==LOW
}
digitalWrite(20, HIGH);
digitalWrite(21, HIGH);
digitalWrite(22, HIGH);
digitalWrite(38, HIGH);

//-----ESP-----
sendCommand("AT+CWMODE=2\r\n", 1000, DEBUG);
sendCommand("AT+CIPSERVER=1,80\r\n", 1000, DEBUG);
sendCommand("AT+CWJAP='internet','overlord'\r\n", 3000, DEBUG);
delay(1000);
sendCommand("AT+CIFSR\r\n", 1000, DEBUG);
sendCommand("AT+CIPMUX=1\r\n", 1000, DEBUG);
sendCommand("AT+CIPSERVER=1,80\r\n", 1000, DEBUG);

zacentnaInicializacija();
}

void loop() {

    void serialFlush();
    if(Serial1.find("+IPD,"))
    {
        espRead();
    }

    checkAll();
    while(val_20 == LOW || val_21 == LOW || val_2 == LOW || val_3 == LOW ||
val_4 == LOW || val_5 == LOW || val_22 == LOW || val_38 == LOW ){

        if(val_22==LOW){
            if(stanjeRolet(0,state_kuhna_s,spust_do_konca)<100){
                state_kuhna_s = dvig_spust(6,state_kuhna_s);
            }
            else digitalWrite(6,HIGH);
        }
        else digitalWrite(6,HIGH);
        if(val_38 == LOW){
            if(stanjeRolet(0,state_kuhna_s,spust_do_konca)>0){
                state_kuhna_s = dvig_spust(7,state_kuhna_s);
            }
            else digitalWrite(7,HIGH);
        }
    }
}
```

```

    }
    else digitalWrite(7,HIGH);
    if(val_20==LOW){
        if(stanjeRolet(0,state_kuhna_b,spust_do_konca)<100){
            state_kuhna_b = dvig_spust(8,state_kuhna_b);
        }
        else digitalWrite(8,HIGH);
    }
    else digitalWrite(8,HIGH);
    if(val_21==LOW){
        if(stanjeRolet(0,state_kuhna_b,spust_do_konca)>0){
            state_kuhna_b = dvig_spust(9,state_kuhna_b);
        }
        else digitalWrite(9,HIGH);
    }
    else digitalWrite(9,HIGH);
    if(val_2==LOW){
        if(stanjeRolet(0,state_dnevna_v,spust_do_konca)<100){
            state_dnevna_v = dvig_spust(10,state_dnevna_v);
        }
        else digitalWrite(10,HIGH);
    }
    else digitalWrite(10,HIGH);
    if(val_3==LOW){
        if(stanjeRolet(0,state_dnevna_v,spust_do_konca)>0){
            state_dnevna_v = dvig_spust(11,state_dnevna_v);
        }
        else digitalWrite(11,HIGH);
    }
    else digitalWrite(11,HIGH);
    if(val_4==LOW){
        if(stanjeRolet(0,state_dnevna_o,spust_do_konca)<100){
            state_dnevna_o = dvig_spust(12,state_dnevna_o);
        }
        else digitalWrite(12,HIGH);
    }
    else digitalWrite(12,HIGH);
    if(val_5==LOW){
        if(stanjeRolet(0,state_dnevna_o,spust_do_konca)>0){
            state_dnevna_o = dvig_spust(13,state_dnevna_o);
        }
        else digitalWrite(13,HIGH);
    }
    else digitalWrite(13,HIGH);

    checkAll();
}

for(int i = 0; i <=13; i++){
    digitalWrite(i,HIGH);
}

//////////AVTOMATSKO-
DVIG(7,9,11,13)//////////

    if(rtc.getHours()>cas_dviga_kuhinja_stopnisce &&
    rtc.getMinutes()>cas_dviga_kuhinja_stopnisce_m && done_1 !=
    true){ //kuhna stopnisce
        sec = rtc.getSeconds() + trajanje_dviga;
        while(rtc.getSeconds() <= sec){

```

```
        digitalWrite(7,LOW);
    }
    state_kuhna_s-=trajanje_dviga;
    done_1=true;
    digitalWrite(7,HIGH);
}
if(rtc.getHours()>cas_dviga_kuhinja_balkon &&
rtc.getMinutes()>cas_dviga_kuhinja_balkon_m && done_2 != true){    //kuhna
balkon
    sec = rtc.getSeconds() + trajanje_dviga;
    while(rtc.getSeconds() <= sec){
        digitalWrite(9,LOW);
    }
    state_kuhna_b-=trajanje_dviga;
    done_2=true;
    digitalWrite(9,HIGH);
}
if(rtc.getHours()>cas_dviga_dnevna_vrata &&
rtc.getMinutes()>cas_dviga_dnevna_vrata_m && done_3 != true){    //dnevna
vrata
    sec = rtc.getSeconds() + trajanje_dviga;
    while(rtc.getSeconds() <= sec){
        digitalWrite(11,LOW);
    }
    state_dnevna_v-=trajanje_dviga;
    done_3=true;
    digitalWrite(11,HIGH);
}
if(rtc.getHours()>cas_dviga_dnevna_okno &&
rtc.getMinutes()>cas_dviga_dnevna_okno_m && done_4 != true){    //dnevna
okno
    sec = rtc.getSeconds() + trajanje_dviga;
    while(rtc.getSeconds() <= sec){
        digitalWrite(13,LOW);
    }
    state_dnevna_o-=trajanje_dviga;
    done_4=true;
    digitalWrite(13,HIGH);
}

/////////////////////////AVTOMATSKO-
SPUST(6,8,10,12)/////////////////////////////////////////

    if(rtc.getHours()>cas_spusta_kuhinja_stopnisce &&
rtc.getMinutes()>cas_spusta_kuhinja_stopnisce_m && done_5 !=
true){    //kuhna stopnisce
    sec = rtc.getSeconds() + trajanje_dviga;
    while(rtc.getSeconds() <= sec){
        digitalWrite(6,LOW);
    }
    state_kuhna_s+=trajanje_dviga;
    done_5=true;
    digitalWrite(6,HIGH);
}
if(rtc.getHours()>cas_spusta_kuhinja_balkon &&
rtc.getMinutes()>cas_spusta_kuhinja_balkon_m && done_6 != true){    //kuhna
balkon
    sec = rtc.getSeconds() + trajanje_dviga;
    while(rtc.getSeconds() <= sec){
        digitalWrite(8,LOW);
    }
}
```

```

        state_kuhna_b+=trajanje_dviga;
        done_6=true;
        digitalWrite(8,HIGH);
    }
    if(rtc.getHours()>cas_spusta_dnevna_vrata &&
rtc.getMinutes()>cas_spusta_dnevna_vrata_m && done_7 != true){    //dnevna
vrata
        sec = rtc.getSeconds() + trajanje_dviga;
        while(rtc.getSeconds() <= sec){
            digitalWrite(10,LOW);
        }
        state_dnevna_v+=trajanje_dviga;
        done_7=true;
        digitalWrite(10,HIGH);
    }
    if(rtc.getHours()>cas_spusta_dnevna_okno &&
rtc.getMinutes()>cas_spusta_dnevna_okno_m && done_8 != true){    //dnevna
okno
        sec = rtc.getSeconds() + trajanje_dviga;
        while(rtc.getSeconds() <= sec){
            digitalWrite(12,LOW);
        }
        state_dnevna_o+=trajanje_dviga;
        done_8=true;
        digitalWrite(12,HIGH);
    }

    if(rtc.getHours()== 23 && rtc.getMinutes()== 59 && rtc.getSeconds()>55){
        reset();
    }

}

void checkAll(){
    val_22 = digitalRead(22);
    val_38 = digitalRead(38);
    val_20 = digitalRead(20);
    val_21 = digitalRead(21);
    val_2 = digitalRead(2);
    val_3 = digitalRead(3);
    val_4 = digitalRead(4);
    val_5 = digitalRead(5);
}

int dvig_spust(int x, int cas){
    unsigned long sec = millis()+1000;
    while(millis()<=sec){
        digitalWrite(x,LOW);
    }
    if(x%2==0){
        cas++;
    }
    else cas--;
    return cas;
}

void reset(){
    done_1 = false;
    done_2 = false;
    done_3 = false;
    done_4 = false;

```

```

done_5 = false;
done_6 = false;
done_7 = false;
done_8 = false;

if(stetje_dnevov==6 || stetje_dnevov == 7){
    spremeniVikend();
    if(stetje_dnevov == 7){
        stetje_dnevov=0;
    }
}
else{
    spremeniTeden();
}
stetje_dnevov++;

delay(5000);
}

//-----ESP-----
bool espRead(){
    String content;
    delay(1000);

    int connectionId = Serial1.read()-48;
    String closeCommand = "AT+CIPCLOSE=";
    closeCommand+=connectionId;
    closeCommand+="\r\n";

    Serial1.find("pin=");

    int pinNumber = (Serial1.read()-48);
    if(pinNumber==1){
        pinNumber = pinNumber*10;
        pinNumber += (Serial1.read()-48);
    }

    procent = (Serial1.read()-48);
    if(procent == -15){
        procent = Serial1.read()-48; //prva stevka
        int procentTemp = Serial1.read()-48; //druga stevka
        if(procentTemp != -16){
            procent = procent*10;
            procent += procentTemp;
            procentTemp = Serial1.read()-48; //tretja stevka
        }
        if(procentTemp != -16){
            procent = procent*10;
            procent += procentTemp;
        }
    }
    else if(procent == 15){//ascii ?
        content =
stanjeRolet(getOdstetiPin(pinNumber),getStanjeRolet(pinNumber),spust_do_kon
ca);
        sendHTTPResponse (connectionId, content);

        sendCommand(closeCommand, 500, DEBUG);
        return true;
    }
}

```

```
        content = "Pin ";
        content += pinNumber;
        content += " is ";

        if(checkState(pinNumber)){
            content += "Working";
        }
        else{
            content += "Error";
        }
        Serial.println(content);
        sendHTTPResponse (connectionId, content);

        sendCommand(closeCommand, 500, DEBUG);
        return true;
    }
String sendData(String command, const int timeout, boolean debug)
{
    String response = "";
    int dataSize = command.length();
    char data[dataSize];
    command.toCharArray(data,dataSize);

    Serial1.write(data,dataSize);
    if(debug){
        Serial.println("\r\n===== HTTP Response From Arduino =====");
        Serial.write(data,dataSize);
        Serial.println("\r\n=====");
    }
    long int time = millis();

    while((time+timeout)>millis()){
        while(Serial1.available()){
            char c = Serial1.read();
            response+=c;
        }
    }
    if(debug){
        Serial.print(response);
    }
    return response;
}

void sendHTTPResponse(int connectionId, String content){
    String httpResponse;
    String httpHeader;

    httpHeader = "HTTP/1.1 200 OK\r\nContent-Type: text/html; charset=UTF-
8\r\n";
    httpHeader += "Content-Length: ";
    httpHeader += "\r\n";
    httpHeader += "Connection: close\r\n\r\n";
    httpResponse = httpHeader + content + " "; //presledek, drugaÄe odreÄe
    sendCIPData(connectionId,httpResponse);
}

void sendCIPData(int connectionId, String data){
    String cipSend = "AT+CIPSEND=";
    cipSend += connectionId;
    cipSend += ",";
    cipSend += data.length();
    cipSend += "\r\n";
```

```
    sendCommand(cipSend, 500, DEBUG);
    sendData(data, 500, DEBUG);
}
String sendCommand(String command, const int timeout, boolean debug){
    String response = "";

    Serial1.print(command);
    long int time = millis();
    while((time+timeout)>millis()){
        while(Serial1.available()){
            char c = Serial1.read();
            response+=c;
        }
    }
    if(debug){
        Serial.print(response);
    }
    return response;
}

int stanjeRolet(int x, int state, double spust_do_konca[]){
    int procent;
    procent = (100*state)/spust_do_konca[x];
    return procent;
}

static bool checkState(int pin){
    if(stanjeRolet(getOdstetiPin(pin), getStanjeRolet(pin), spust_do_konca)>procent){

        setPinSpust(pin, procent);
        return true;
    }
    else{
        setPinDvig(pin, procent);
        return true;
    }
}
return false;
}

void setPinSpust(int pin, int procent){
    digitalWrite(pin+1, LOW);
    unsigned long previousSec = 0;
    unsigned long sec = millis();
    previousSec = sec;
    while(stanjeRolet(getOdstetiPin(pin), getStanjeRolet(pin), spust_do_konca)>procent){ //+6
        sec = millis();
        if(sec-previousSec>=1000){
            previousSec = sec;
            odstejStanjeRolet(pin);
            delay(10);
        }
    }
    digitalWrite(pin+1, HIGH);
}

void setPinDvig(int pin, int procent){
    digitalWrite(pin, LOW);
    unsigned long previousSec = 0;
    unsigned long sec = millis();
    previousSec = sec;
```

```
while(stanjeRolet(getOdstetiPin(pin),getStanjeRolet(pin),spust_do_kon
ca)<procent){
    sec = millis();
    if(sec-previousSec>=1000){
        previousSec = sec;
        pristejStanjeRolet(pin);
        delay(10);
    }
    digitalWrite(pin, HIGH);
}
int getStanjeRolet(int pin){
    if(pin == 6 || pin == 7){
        return state_kuhna_s;
    }
    else if(pin == 8 || pin == 9){
        return state_kuhna_b;
    }
    else if(pin == 10 || pin == 11){
        return state_dnevna_v;
    }
    else if(pin == 12 || pin == 13){
        return state_dnevna_o;
    }
}
int getOdstetiPin(int pin){
    if(pin == 6 || pin == 7){
        return 0;
    }
    else if(pin == 8 || pin == 9){
        return 1;
    }
    else if(pin == 10 || pin == 11){
        return 2;
    }
    else if(pin == 12 || pin == 13){
        return 3;
    }
}
void odstejStanjeRolet(int pin){
    if(pin == 6 || pin == 7){
        state_kuhna_s--;
    }
    else if(pin == 8 || pin == 9){
        state_kuhna_b--;
    }
    else if(pin == 10 || pin == 11){
        state_dnevna_v--;
    }
    else if(pin == 12 || pin == 13){
        state_dnevna_o--;
    }
}
void pristejStanjeRolet(int pin){
    if(pin == 6 || pin == 7){
        state_kuhna_s++;
    }
    else if(pin == 8 || pin == 9){
        state_kuhna_b++;
    }
    else if(pin == 10 || pin == 11){
```

```
    state_dnevna_v++;
}
else if(pin == 12 || pin == 13){
    state_dnevna_o++;
}
}
void zacentnaInicializacija(){
    digitalWrite(6,LOW);
    digitalWrite(8,LOW);
    digitalWrite(10,LOW);
    digitalWrite(12,LOW);
    unsigned long previousSec = 0;
    unsigned long sec = millis();
    previousSec = sec;
    int stetje_konca = 0;
    while(stetje_konca != 4){
        sec = millis();
        if(spust_do_konca[0]>sec/1000-previousSec/1000 && digitalRead(6) ==
LOW){
            digitalWrite(6,HIGH);
            stetje_konca++;
        }
        if(spust_do_konca[1]>sec/1000-previousSec/1000 && digitalRead(6) ==
LOW){
            digitalWrite(8,HIGH);
            stetje_konca++;
        }
        if(spust_do_konca[2]>sec/1000-previousSec/1000 && digitalRead(6) ==
LOW){
            digitalWrite(10,HIGH);
            stetje_konca++;
        }
        if(spust_do_konca[3]>sec/1000-previousSec/1000 && digitalRead(6) ==
LOW){
            digitalWrite(12,HIGH);
            stetje_konca++;
        }
    }
}
```

## 5.2 Koda Android

```
//Prva stran
package com.example.jure.arduinoprojekt;

import android.app.AlertDialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.AsyncTask;
import android.renderscript.Sampler;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import org.apache.http.HttpResponse;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.DefaultHttpClient;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URI;
import java.net.URISyntaxException;

public class Prva_stran extends AppCompatActivity implements
View.OnClickListener {
    public final static String PREF_USERNAME = "PREF_USERNAME";
    public final static String PREF_PASS = "PREF_PASS";

    private EditText editTextIPAddress, editTextPortNumber;
    private Button nextButton;
    private boolean autor;

    SharedPreferences.Editor editor;
    SharedPreferences sharedPreferences;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_prva_stran);

        sharedPreferences = getSharedPreferences("USERNAME_PREFS",
Context.MODE_PRIVATE);
        editor = sharedPreferences.edit();

        nextButton = (Button)findViewById(R.id.nextButton);

        editTextIPAddress = (EditText)
findViewById(R.id.editTextIPAddress);
        editTextPortNumber = (EditText)
findViewById(R.id.editTextPortNumber);

        nextButton.setOnClickListener(this);
```

```

editTextIPAddress.setText(sharedPreferences.getString(PREF_USERNAME, ""));
    editTextPortNumber.setText(sharedPreferences.getString(PREF_PASS,
""));
}

@Override
public void onClick(View v) {

    // get the ip address
    String username = editTextIPAddress.getText().toString().trim();
    // get the port number
    String pass = editTextPortNumber.getText().toString().trim();

    editor.putString(PREF_USERNAME, username); // set the ip address
value to save
    editor.putString(PREF_PASS, pass); // set the port number to save
    editor.commit(); // save the IP and PORT

    if(v.getId()==nextButton.getId())
    {
        String parameterValue = "6";
        new Prva_stran.HttpRequestAsyncTask(v.getContext(),
parameterValue, "192.168.4.1", "80", "pin",username,pass).execute();

    }

}

public void openMain() {

    if(autor){
        Intent intent = new Intent(this, Druga_stran.class);
        this.startActivity ( intent );
    }

}

public String sendRequest(String parameterValue, String ipAddress,
String portNumber, String parameterName,String username, String pass) {
    String serverResponse = "ERROR";

    try {

        HttpClient httpclient = new DefaultHttpClient(); // create an
HTTP client
        // npr. http://myIpaddress:myport/?pin=13!12
        URI website = new
URI("http://" +ipAddress+": "+portNumber+"/?"+parameterName+"="+username+"/"
+pass);
        HttpGet getRequest = new HttpGet(); // create an HTTP GET
object
        getRequest.setURI(website); // set the URL of the GET request
        HttpResponse response = httpclient.execute(getRequest); //
execute the request
        // get the ip address server's reply
        InputStream content = null;
        content = response.getEntity().getContent();
        BufferedReader in = new BufferedReader(new InputStreamReader(
            content
        ));
    }
}

```

```
        serverResponse = in.readLine();
        // Close the connection
        content.close();
    } catch (ClientProtocolException e) {
        // HTTP error
        serverResponse = e.getMessage();
        e.printStackTrace();
    } catch (IOException e) {
        // IO error
        serverResponse = e.getMessage();
        e.printStackTrace();
    } catch (URISyntaxException e) {
        // URL syntax error
        serverResponse = e.getMessage();
        e.printStackTrace();
    }
    // return the server's reply/response text
    return serverResponse;
}

private /*static*/ class HttpRequestAsyncTask extends AsyncTask<Void,
Void, Void> {

    private String requestReply, ipAddress, portNumber, username, pass;
    private Context context;
    private AlertDialog alertDialog;
    private String parameter;
    private String parameterValue;

    public HttpRequestAsyncTask(Context context, String parameterValue,
String ipAddress, String portNumber, String parameter, String username,
String pass)
    {
        this.context = context;

        alertDialog = new AlertDialog.Builder(this.context)
            .setTitle("HTTP Response From IP Address:")
            .setCancelable(true)
            .create();

        this.ipAddress = ipAddress;
        this.parameterValue = parameterValue;
        this.portNumber = portNumber;
        this.parameter = parameter;
        this.username = username;
        this.pass = pass;
    }
    @Override
    protected Void doInBackground(Void... voids) {
        alertDialog.setMessage("Data sent, waiting for reply from
server...");
        if(!alertDialog.isShowing())
        {
            alertDialog.show();
        }
        requestReply = sendRequest(parameterValue, ipAddress, portNumber,
parameter, username, pass);
        return null;
    }
}
```

```
//Druga stran
package com.example.jure.arduinoprojekt;

import android.app.AlertDialog;
import android.content.Context;
import android.content.Intent;

import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;

import org.apache.http.HttpResponse;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.DefaultHttpClient;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URI;
import java.net.URISyntaxException;

public class Druga_stran extends AppCompatActivity implements
View.OnClickListener{
    private Button buttonPin6, buttonPin8, buttonPin10,buttonPin12;
    private String percent = "neki";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_druga_stran);

        // assign buttons
        buttonPin6 = (Button)findViewById(R.id.buttonPin6);
        buttonPin8 = (Button)findViewById(R.id.buttonPin8);
        buttonPin10 = (Button)findViewById(R.id.buttonPin10);
        buttonPin12 = (Button)findViewById(R.id.buttonPin12);

        // set button listener (this class)
        buttonPin6.setOnClickListener(this);
        buttonPin8.setOnClickListener(this);
        buttonPin10.setOnClickListener(this);
        buttonPin12.setOnClickListener(this);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) { //settings prikaz
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.menu_settings, menu);
        return true;
    }
}
```

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {//detect
    menu_settings
    if (item.getItemId() == R.id.your_item_id) {
        Intent intent = new Intent(this, String.class);
        startActivity(intent);
        return true;
    }
    return super.onOptionsItemSelected(item);
}

public void openPosameznaRoleta(String ipAddress, String portNumber,
String pinNumber, String percent) { //sending data tru activities
    Intent intent = new Intent(this, Posamezna_roleta.class);

    Bundle extras = new Bundle();
    extras.putString("EXTRA_IP", ipAddress);
    extras.putString("EXTRA_PORT", portNumber);
    extras.putString("EXTRA_PIN", pinNumber);
    extras.putString("EXTRA_PERCENT", percent);
    intent.putExtras(extras);

    startActivity(intent);
}

@Override
public void onClick(View view) {

    // assign text inputs
    String IP = "192.168.4.1";
    String PORT = "80";

    // get the pin number
    String parameterValue = "";
    // get the ip address
    String ipAddress = IP;

    String portNumber = PORT;

    // get the pin number from the button that was clicked
    if(view.getId()==buttonPin6.getId())
    {
        parameterValue = "6";//+7
    }
    else if(view.getId()==buttonPin8.getId())
    {
        parameterValue = "8";//+9
    }
    else if(view.getId()==buttonPin10.getId())
    {
        parameterValue = "10";//+11
    }
    else
    {
        parameterValue = "12";//+13
    }

    if(ipAddress.length()>0 && portNumber.length()>0) {
        new Druga_stran.HttpRequestAsyncTask(view.getContext(),
parameterValue, ipAddress, portNumber, "pin").execute();
    }
}
```

```
    }
    public String sendRequest(String parameterValue, String ipAddress,
String portNumber, String parameterName) {
        String serverResponse = "ERROR";

        try {

            HttpClient httpClient = new DefaultHttpClient(); // create an
HTTP client
            // npr. http://myIpAddress:myport/?pin=13!12
            URI website = new
URI("http://" + ipAddress + ":" + portNumber + "/" + parameterName + "=" + parameterValu
e + "?");
            HttpGet getRequest = new HttpGet(); // create an HTTP GET
object
            getRequest.setURI(website); // set the URL of the GET request
            HttpResponse response = httpClient.execute(getRequest); //
execute the request
            // get the ip address server's reply
            InputStream content = null;
            content = response.getEntity().getContent();
            BufferedReader in = new BufferedReader(new InputStreamReader(
                content
            ));

            serverResponse = in.readLine();
            // Close the connection
            content.close();
        } catch (ClientProtocolException e) {
            // HTTP error
            serverResponse = e.getMessage();
            e.printStackTrace();
        } catch (IOException e) {
            // IO error
            serverResponse = e.getMessage();
            e.printStackTrace();
        } catch (URISyntaxException e) {
            // URL syntax error
            serverResponse = e.getMessage();
            e.printStackTrace();
        }
        // return the server's reply/response text
        return serverResponse;
    }

    private /*static*/ class HttpRequestAsyncTask extends AsyncTask<Void,
Void, Void> {

        // declare variables needed
        private String requestReply, ipAddress, portNumber;
        private Context context;
        private AlertDialog alertDialog;
        private String parameter;
        private String parameterValue;

        public HttpRequestAsyncTask(Context context, String parameterValue,
String ipAddress, String portNumber, String parameter)
```

```
{
    this.context = context;

    this.ipAddress = ipAddress;
    this.parameterValue = parameterValue;
    this.portNumber = portNumber;
    this.parameter = parameter;
}
@Override
protected Void doInBackground(Void... voids) {
    requestReply = sendRequest(parameterValue, ipAddress, portNumber,
parameter);
    return null;
}
@Override
protected void onPostExecute(Void aVoid) {
    GetPercent();
    openPosameznaRoleta(ipAddress, portNumber, parameterValue,
percent);
}
protected void GetPercent(){
    percent = requestReply;
}
}
}
```

```
//Posamezno senčilo
```

```
package com.example.jure.arduinoprojekt;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Button;
import android.widget.TextView;
import android.app.AlertDialog;
import android.content.Context;
import android.os.AsyncTask;
import android.view.View;

import org.apache.http.HttpResponse;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.DefaultHttpClient;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URI;
import java.net.URISyntaxException;

import com.marcinmoskala.arcseekbar.ArcSeekBar;
import com.marcinmoskala.arcseekbar.ProgressListener;

public class Posamezna_roleta extends AppCompatActivity implements
View.OnClickListener {

    ArcSeekBar gradientSeekBar;
    TextView textView;
    Button button;
    private String percent;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_posamezna_roleta);

        gradientSeekBar = findViewById(R.id.gradientSeekBar);
        button = findViewById(R.id.button);
        button.setOnClickListener(this);

        //gradient
        int[] colorArray = getResources().getIntArray(R.array.gradient);
        gradientSeekBar.setProgressGradient(colorArray);

        textView = findViewById(R.id.textView);
        gradientSeekBar.setOnProgressChangeListener(new ProgressListener()
        {
            @Override
            public void invoke(int i) {
                // gradientVal[0] =i;
                textView.setText(""+i);
                percent=String.valueOf(i);
            }
        });
    }
}
```

```
        Bundle extras = getIntent().getExtras();
        String percent = extras.getString("EXTRA_PERCENT");
        textView.setText(percent);

gradientSeekBar.setProgress(Integer.parseInt(percent.trim())); //trim the
blank spaces
    }

    @Override
    protected void onResume()
    {
        super.onResume();
    }

    @Override
    public void onClick(View v) {

        Bundle extras = getIntent().getExtras();
        String IP = extras.getString("EXTRA_IP");
        String PORT = extras.getString("EXTRA_PORT");
        String PIN = extras.getString("EXTRA_PIN");

        // get the pin number
        String parameterValue = PIN;
        // get the ip address
        String ipAddress = IP;

        String portNumber = PORT;

        // execute HTTP request
        if(ipAddress.length()>0 && portNumber.length()>0) {
            new Posamezna_roleta.HttpRequestAsyncTask(v.getContext(),
parameterValue, ipAddress, portNumber, "pin", percent).execute();
        }
        public String sendRequest(String parameterValue, String ipAddress,
String portNumber, String parameterName, String percent) {
            String serverResponse = "ERROR";

            try {

                HttpClient httpclient = new DefaultHttpClient(); // create an
HTTP client
                // npr. http://myIpaddress:myport/?pin=13!12
                URI website = new
URI("http://" + ipAddress + ":" + portNumber + "/" + parameterName + "=" + parameterValu
e + "!" + percent);
                HttpGet getRequest = new HttpGet(); // create an HTTP GET
object
                getRequest.setURI(website); // set the URL of the GET request
                HttpResponse response = httpclient.execute(getRequest); //
execute the request
                // get the ip address server's reply
                InputStream content = null;
                content = response.getEntity().getContent();
                BufferedReader in = new BufferedReader(new InputStreamReader(
                    content
                ));

                serverResponse = in.readLine();
```

```
        // Close the connection
        content.close();
    } catch (ClientProtocolException e) {
        // HTTP error
        serverResponse = e.getMessage();
        e.printStackTrace();
    } catch (IOException e) {
        // IO error
        serverResponse = e.getMessage();
        e.printStackTrace();
    } catch (URISyntaxException e) {
        // URL syntax error
        serverResponse = e.getMessage();
        e.printStackTrace();
    }
    // return the server's reply/response text
    return serverResponse;
}

private /*static*/ class HttpRequestAsyncTask extends AsyncTask<Void,
Void, Void> {

    // declare variables needed
    private String requestReply, ipAddress, portNumber, percent;
    private Context context;
    private AlertDialog alertDialog;
    private String parameter;
    private String parameterValue;

    public HttpRequestAsyncTask(Context context, String parameterValue,
String ipAddress, String portNumber, String parameter, String percent)
    {
        this.context = context;

        alertDialog = new AlertDialog.Builder(this.context)
            .setTitle("HTTP Response From IP Address:")
            .setCancelable(true)
            .create();

        this.percent = percent;
        this.ipAddress = ipAddress;
        this.parameterValue = parameterValue;
        this.portNumber = portNumber;
        this.parameter = parameter;
    }

    @Override
    protected Void doInBackground(Void... voids) {
        alertDialog.setMessage("Data sent, waiting for reply from
server...");
        if(!alertDialog.isShowing())
        {
            alertDialog.show();
        }
        requestReply = sendRequest(parameterValue, ipAddress, portNumber,
parameter, percent);
        return null;
    }

    @Override
    protected void onPostExecute(Void aVoid) {
```

```
        alertDialog.dismiss();
        GetPercent();
    }

    @Override
    protected void onPreExecute() {
        alertDialog.setMessage("Sending data to server, please
wait...");
        if(!alertDialog.isShowing())
        {
            alertDialog.show();
        }
    }
    protected void GetPercent(){
        textView.setText(requestReply);
    }
}
}
```

#### IZJAVA\*

Mentor Borut Slemenšek v skladu z 2. in 17. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi z naslovom Upravljanje okenskih senčil, katere avtorica je Jure Mazej:  
s krmilnikom Arduino

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje.

Celje, 8.3.2019



Podpis mentorja

Podpis odgovorne osebe

#### POJASNILO

V skladu z 2. in 17. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja (-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja (-ice) fotografskega gradiva, katerega ni avtor (-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.