Glasovno upravljanje doma z RPi

raziskovalna naloga

Mentor:
mag. Boštjan Resinovič, univ. dipl. inž. rač. in inf.

Avtor:
Jakob Dorn, R-4.b

Mestna občina Celje, Mladi za Celje

Celje, februar 2020

Zahvala

Zahvaljujem se vsem, ki so mi pomagali pri izdelavi raziskovalne naloge. Mentorju, profesorju Boštjanu Resinoviču za napotke, in profesorici Valentini Hrastnik za lektoriranje raziskovalne naloge.

Zahvala gre tudi mnogim neimenovanim članom spletnih forumov ter avtorjem člankov ter vodičev s katerimi sem si pomagal pri izdelavi naloge in širjenju znanja.

Povzetek

V raziskovalni nalogi je opisano pametno in glasovno upravljanje doma ter opravil v njem. Izpostavljen je način izdelave takšnega sistema in njegova vpeljava ter aktivna vloga v domu. Predstavljena je strojna in programska oprema ter sodelovanje med napravami in protokoli. Opisana so tudi vsa uporabljena orodja. Ugotovil sem, da je takšen sistem praktičen, a težek za izdelavo ter vzdrževanje. K raziskovanju me je gnalo lastno zanimanje za to področje.

Abstract

In this research paper, I dealt with smart and voice oriented controling of a home and the tasks within it. I described the bulding proces of such a system and its enrolement into a home as an active part of it. The hardware and software sides are both explained, as is the cooperation between the devices and protocols used. I conclude that such a system is practical but hard to make and maintain. My motivation for research is interest in the topic.

Ključne besede

upravljanje doma, pametni dom/hiša, avtomatizacija nalog, raspberry pi, RPi, MQTT, ESP

Key words

home control, smart home/house, task automatization, raspberry pi, RPi, MQTT, ESP

Kratice in okrajšave

RPi – Raspberry Pi

OS – Operacijski sistem

SDK – Software development kit

Kazalo vsebine

Kazalo slik

Kazalo grafov

1 Uvod

Inteligentno in avtomatsko upravljanje doma je v naših življenjih vse bolj pogosto in neizogibno. Na mnogo načinov nam izboljšuje kvaliteto življenja ter skrajša ali olajša vsakdanja opravila. Pri izbiri sistema za nadzor doma se lahko odločimo, koliko nadzora bomo nad tem sistemom imeli. Največ ga imamo, če tak sistem sestavimo in sprogramiramo sami.

1.1 Predstavitev problema

Glasovno upravljanje doma nam je kot potrošnikom na voljo v raznih oblikah, nad katerimi pa nimamo nadzora. Če se odločimo, da bomo tak sistem zgradili sami, se lahko odločamo, kako bo deloval in reagiral na določene parametre, saj smo pri vnaprej zgrajenem sistemu glede tega močno omejeni. Glasovno upravljanje takega sistema nam omogoča lažji dostop ter več načinov uporabe.

1.2 Hipoteze

Postavil sem nekaj hipotez v zvezi s pametnim upravljanjem doma in tudi hipotezi glede odgovorov anketirancev.

Hipoteza 1: Izdelava pametnega sistema za upravljanje doma je težka, saj se malokdo ukvarja s tem.

Hipoteza 2: Podlago za znanje imam solidno, a se bom moral naučiti več o protokolih, strojni opremi in delovanju takšnega sistema.

Hipoteza 3: Svoj sistem bom zlahka razširil ter poleg osnovnih dodal nove funkcije.

Hipoteza 4: Večina anketirancev bi takšen sistem uporabljalo in vključilo v svoj vsakdan.

1.3 Raziskovalne metode

Za raziskovanje sem uporabil metodo anketiranja, v kateri sem anketirance povprašal glede njihovih izkušenj s takšnimi sistemi. Vprašal sem jih tudi o tem, ali so pripravljeni takšen sistem sestaviti in uporabljati.

Preučil sem internetne vire ter primere delujočih komercialnih sistemov, kar mi je pomagalo pri načrtovanju in izgradnji svojega.

Rezultate sem ugotavljal z izdelavo izdelka ter eksperimentiranjem z različnimi načini izdelave.

1.4 Cilji

Cilj moje raziskovalne naloge je, da ustvarim sistem za nadzor doma s pomočjo glasovnega upravljanja preko aplikacije za mobilni telefon. Uporabnik lahko tako prižge ter ugasne luč ali druge naprave.

2 Teoretični del

2.1 Pametni dom

Pametni dom ali pametna hiša je način poenostavljenja vsakdanjih nalog v našem življenju, predvsem hišnih opravil. Takšen sistem ima razne ravni dostopa do hiše ter različne nivoje integracije v naše življenje. Njegove najosnovnejše naloge so ugašanje luči ter podobnih naprav, opravlja pa tudi bolj kompleksne naloge, kot je varovanje doma pred vlomom. Najpogostejša avtomatizirana opravila so:

- zapiranje garažnih vrat po odhodu,
- zalivanje zelenice,
- javljanje morebitne poplave in
- nadzor gibanja/varnosti.

Za odločanje o opravilih/dejanjih potrebuje vhodne podatke. Te pridobi iz vrste senzorjev (senzorji za vlago, svetlobo, temperaturo …) ter od samega uporabnika (preko uporabniških vmesnikov znanih kot »UI«).

2.2 Strojna oprema

Pri izbiri strojne opreme sem se odločal glede na ceno, hitrost in razpoložljivo dokumentacijo. Mini računalnik Raspberry Pi sem že imel, prav tako sem od prejšnjega projekta imel mikrokrmilnik ESP-8266. Za oba produkta je veliko dokumentacije in pomoči, kar je prišlo prav, ko sem izdeloval izdelek.

Dodatna zahteva pri izbiri je bila ta, da je vsa oprema tiha in majhna. Če bi želel, bi lahko večino stvari izvedel na navadnem domačem računalniku; zataknilo bi se pri postavitvi tega v prostor in pri omejevanju širjenja hrupa ventilatorjev za hlajenje procesorja in ostalih komponent. RPi in ESP izstopata, saj sta oba majhna in izredno tiha, ker se zanašata le na pasivno hlajenje, ne pa na aktivnega.

2.2.1 Raspberry Pi

2.2.1.1 Osnovni podatki

Raspberry Pi je mikroračunalnik, ki je malo manjši kot bankovec za 5 € in predstavlja idealno napravo za cenovno ugodno reševanje težav z avtomatizacijo. Obstaja več raznih modelov z različnimi specifikacijami, jaz sem uporabil Raspberry Pi 3 Model B, ki se ponaša z naslednjimi specifikacijami ter priklopi:

- 4 USB Porti,
- Ethernet port,
- Wi-Fi,
- priklop za kamero ter zaslon,
- HDMI,
- AUX,
- Bluetooth,
- GPIO konektorji (za povezavo raznih naprav ter komponent).

Vsebuje tudi štirijedrni 64-bitni procesor, ki deluje s hitrostjo 1,4 GHz, kar ni dovolj hitro za napredne aplikacije in zahtevne naloge, a vseeno dovolj za moj cilj, ki temelji na bolj pasivnem delovanju, saj RPi uporabljam le kot posrednik.

Cena novega je okoli 30 €, kar pomeni, da si ga lahko privošči skoraj vsak. Za uporabo potrebujemo le še nekaj drugih komponent:

- micro USB napajalnik, ki lahko dovaja vsaj 2,5 A,
- micro SD kartico, na kateri je nameščen operacijski sistem,
- opcijsko ohišje, da RPi zaščitimo,
- opcijski »heatsink« za odvajanje toplote od procesorja in ostalih elementov.

Pri svojem sistemu sem za nadzor uporabil miško, tipkovnico ter monitor. To v resnici ni potrebno, saj se lahko na RPi povežemo preko SSH (secure shell), ki nam omogoča poganjanje programov ter aplikacij preko oddaljene konzole.

Luči bi lahko zaradi GPIO pinov upravljal tudi preko samega RPI-ja, ampak ob tem nastane težava, da mora biti luč neposredno vezana na RPi, kar pa v primeru razpršenosti luči po celem stanovanju ni mogoče.

Slika 1: Raspberry Pi ob bankovcu za 5 €

2.2.1.2 Operacijski sistem

Za ta model Raspberry Pi obstaja več operacijskih sistemov. Vsi so naloženi na micro SD kartico, razlikujejo se večinoma v količini tovarniško naloženih programov ter razvojnih okolij. Najbolj pogosto uporabljen operacijski sistem je Raspbian (verzija Linuxa), za katerega sem se odločil, saj je uradni in deluje najbolje ter je zanj prilagojeno največje število orodij in programov. Je prilagojena verzija operacijskega sistema Debian, ki je distribucija oz. različica operacijskega sistema Linux. Ima enostaven uporabniški vmesnik in omogoča upravljanje preko konzole. Vsi potrebni programi niso bili naloženi, a to ni bila težava, saj je inštalacija enostavna, v kolikor program podpira Linux oz. Raspbian.

Večina drugih operacijskih sistemov je prilagojenih za določene namene:

- Windows 10 IoT – posebna izdaja Windows 10 je primerna za Raspberry Pi,
- Kodi – izdelan za predvajanje multimedije,
- RISC OS Pi – enostavnejši OS, ki zaseda manj prostora (le 16 MB),
- Ark OS – ustvarjen za gostovanje spletnih in poštnih strežnikov,
- CentOS – popolnoma osnovni OS brez nepotrebnih dodatkov in
- še mnogo drugih.

2.2.2 ESP 8266

ESP 8266 je cenovno ugodni mikročip z zmožnostjo uporabe polnega TCP/IP protokola. Izdeluje ga Espressif Systems na Kitajskem. Čip je prvič postal popularen leta 2014 v verziji ESP-01. Ta mali modul je bil cenovno ugoden in je omogočal dober dostop do Wi-Fi omrežij v majhni velikosti. Po ESP-01 je bilo izdelanih veliko boljših verzij, vključno z ESP-8266 (uporabljen v tej raziskovalni nalogi). Najnovejša različica je ESP-32, ki ima dodatno jedro za zmožnost operacije z dvema jedroma. Na njem je vgrajenih 16 GPIO (General Purpose Input Output) priključkov za krmiljenje drugih naprav. Podpira do 16 MiB vgrajenega spomina za hranjenje programa. Vgrajeno ima tudi podporo za I²C protokol, ki omogoča povezavo z raznimi senzorji ter branje podatkov iz njih. Napajan je preko baterije ali konstantnega vira, prilagojenega na 3.3V. Ima zmožnost preklopa na način globokega spanja, kjer je poraba energije minimalna in s tem omogoča delovanje v daljšem razponu časa, kadar ni potrebno konstantno javljanje podatkov.

Programira se ga lahko z raznimi SDK-ji (software development kit), ki podpirajo različne funkcije ter so zaradi tega bolj in manj atraktivni za uporabo v sistemu nadzora doma. Najbolj popularni SDK-ji so:

- NodeMCU – zasnovan na Lua jeziku, tovarniško naložen na večini starejših,
- Arduino – najbolj razširjen in dokumentiran, enostaven za uporabo (uporablja C++),
- MicroPython – prirejena verzija Pythona,

- ESP8266 BASIC – odprtnokodna verzija BASIC jezika, prirejenega za ESP.

Nalaganje oz. »flashanje« SKD-jev je enostavno, potreben je dober razmislek, ki ga imamo namen uporabiti, saj to močno vpliva na izvedbo. Za ta projekt sem izbral Arduino, ker mi je znan jezik C++, zanj je tudi največ primerov izvedbe ter dokumentacije o pogostih težavah.

Slika 2: ESP-8266 ob kovancu za 1 €

Za ESP-8266 sem se odločil, ker je odlična naprava z majhno velikostjo in prav tako nizko ceno, kar pomeni, da jih lahko kupimo več in postavimo na več točk v stanovanju ter s tem upravljamo več naprav kot le eno.

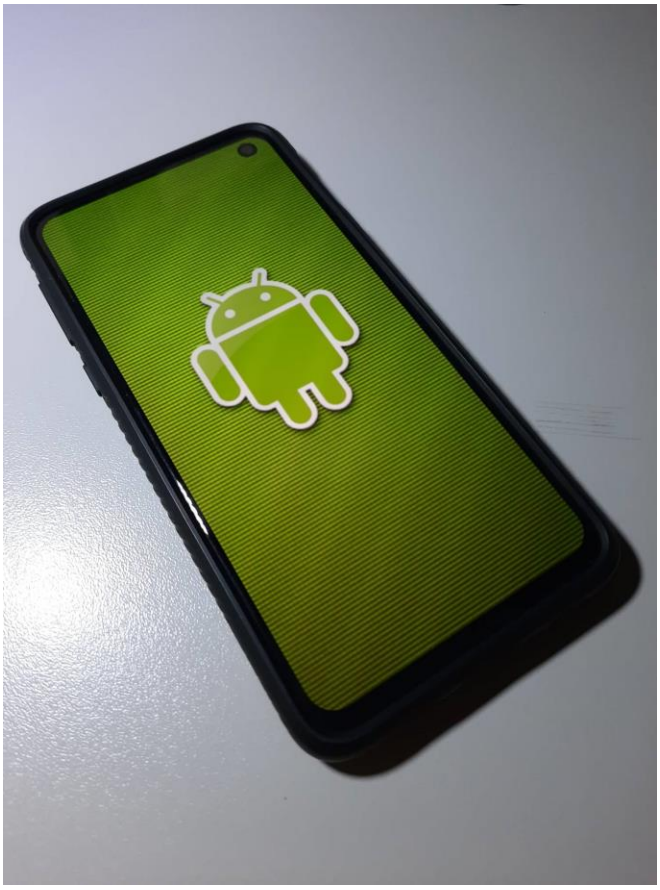
Ker lahko vsaki dodelimo statičen IPv4 naslov, je njihovo število praktično neomejeno, a obstajajo tudi manjše verzije, ki bi bile primerne za vgradnjo v vtičnice ali LED panele.

2.2.3 Mobilni telefon

Glavna atrakcija pametnega doma je, da lahko stvari in naprave upravljamo na daljavo. Če obstaja daljinski upravljalnik za televizor, zakaj ne bi obstajal daljinski upravljalnik za druge naprave, kot so luči in garažna vrata? Za to lahko uporabimo napravo, ki je večini dobro znana, saj se le redkokdaj ločimo od nje - mobilni telefon.

Ker uporabljam telefon s operacijskim sistemom Android, sem se odločil, da bom aplikacijo razvil zanj. Za razvoj aplikacije za ta OS je tudi več dokumentacije.

Pri izdelavi aplikacije se je potrebno odločiti, za katero verzijo Androida bomo le-to izdelali, saj vpliva na izbiro funkcij, ki jih lahko dodamo. Hkrati vpliva na število naprav, na katerih bo aplikacija pravilno delovala. Ker uporabljam telefon, ki ima verzijo 10, sem lahko uporabil vse funkcije ter možnosti pri razvoju.



Slika 3: Android pametni telefon

2.3 Programska oprema

Pri izdelavi projekta sem se soočal z mnogimi odločitvami glede strojne in programske opreme. Strojno opremo sem izbral glede na popularnost in dokumentacijo, izbiranje programske opreme pa je bilo težje, saj je ta morala točno ustrezati mojemu projektu ter zahtevam.

Poleg tega je morala biti enostavna ter kompatibilna s prav tako strojno opremo kot drugo programsko opremo. Med drugim sem moral izbrati IDE-je (integrirano razvojno okolje), protokole (za pošiljanje podatkov), programski jezik za testiranje in operacijske sisteme.

2.3.1 Razvojna okolja

Razvojno okolje je računalniška aplikacija, ki omogoča uporabo orodij za razvoj programov in aplikacij. Obstaja več vrst za več namenov in platform.

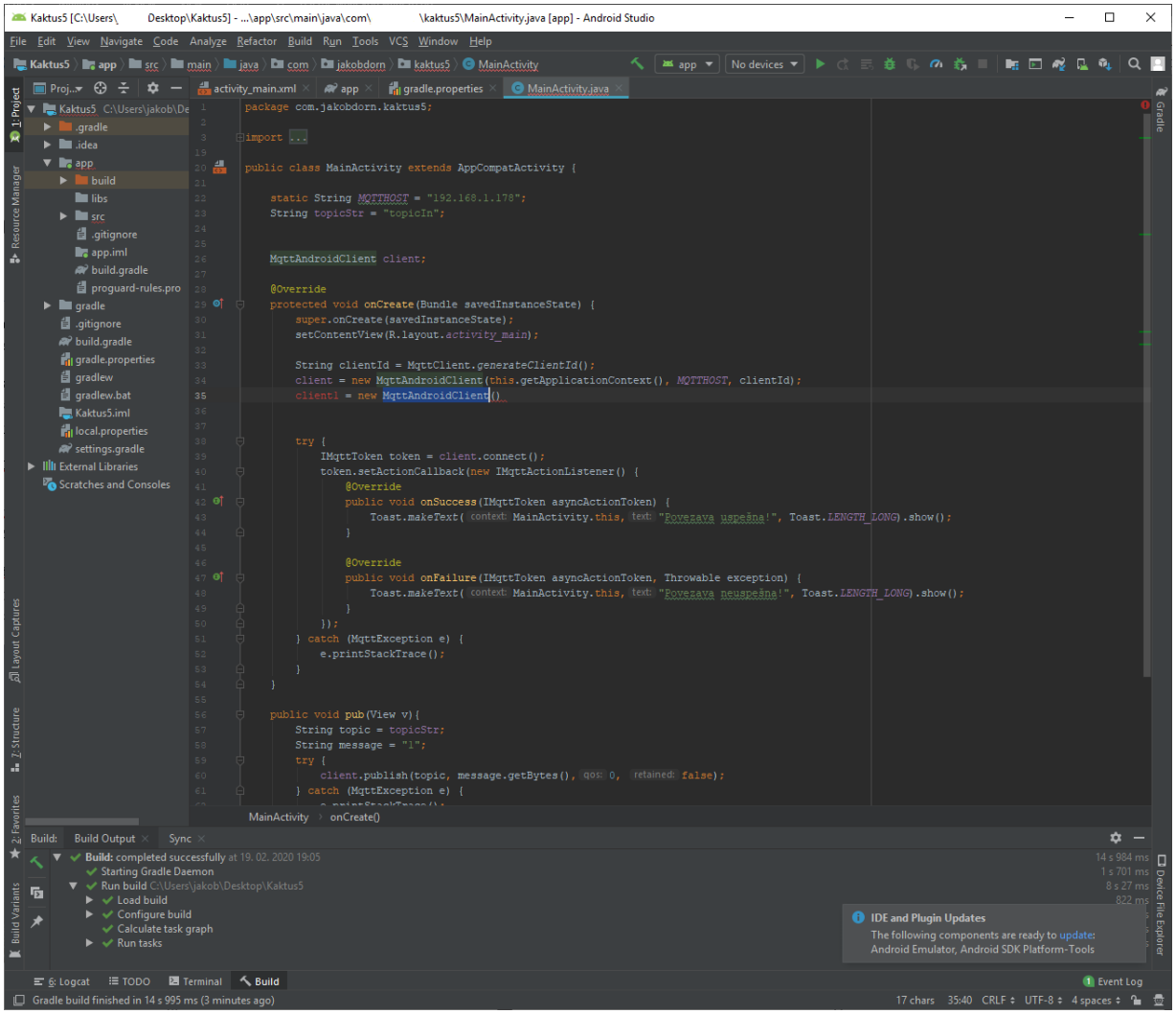
Ker sem programiral aplikacijo za Android telefon, sem izbral Android Studio, za programiranje ESP-ja sem izbral Arduino IDE, na Raspberry Pi pa sem za programiranje v jeziku Python uporabil vgrajeni Python IDLE.

2.3.1.1 Android studio

Android studio je uradni IDE za razvijanje aplikacij za Android. Prirejen je specifično za ustvarjanje programov in aplikacij za mobilne telefone in tablice, drugih naprav načeloma ne podpira.

V Android studiu lahko izbiramo med dvema jezikoma. Prvi je Kotlin, novejša verzija Jave, ki ima nekaj dodatnih funkcij ter podobno, a ne enako sintakso. Ta se je prevzel kot privzeti jezik šele sredi leta 2019. Zaradi poznega prevzema ni bil moja prva izbira, saj zanj ni veliko podpore, vodičev in obstoječih projektov, kjer bi si lahko pogledal že delujoče aplikacije in programe.

Uporablja se lahko tudi C++, originalni pa je Java, star jezik, ki se je uporabljal že pred Androidom za vrsto drugih namenov. Za uporabo Jave sem se odločil, ker dobro poznam jezik C#, ki je Javi nekoliko podoben, prav tako pa v šoli uporabljamo Javo za razvoj aplikacij v IDE-ju Android studio. Zanj je tudi največ dokumentacije, rešenih težav, odgovorjenih vprašanj, vodičev ter končanih projektov.



Slika 4: Android Studio IDE

IDE ni prijazen za začetnike, saj ponuja veliko funkcij in s tem tudi veliko možnosti za napake in frustracijo ob delu. Ker sem relativno izkušen programer, mi to ni delalo prevelikih težav, edina je bila soočanje z novim programskim jezikom (Java), njegovo sintakso in razvijanje aplikacije, ki bo prijazna tudi uporabniku, ne le ustvarjalcu.

Program lahko preizkusimo na dva načina:

- Uporabimo emulator, ki nam na računalniku zažene Android sistem, ta je skoraj identičen telefonu, a deluje počasneje. Edina omejitev je ta, da moraš uporabljati procesor znamke Intel, ker le te podpirajo takšno emulacijo.

- Uporabimo fizično Android napravo, povezano na računalnik preko USB kabla. To opcijo sem izbral, ker imam procesor znamke AMD.

Reševanje težav v kodi je presenetljivo enostavno. Ko se aplikacija naloži na telefon, ostane ta povezan na računalnik. Ob morebitni napaki ali sesutju programa ta vrne vse podatke na računalnik v posebno konzolo, kjer se prikazujejo. Tam lahko vidiš, kje je prišlo do napake in jo tako tudi odpraviš.

2.3.1.2 Arduino

Arduino IDE je uradno razvojno okolje za prototipne plošče Arduino in njene različice. Napisano je v jezikih C in C++, v njem pa se programira v C++ ali opcijsko drugem jeziku, v kolikor se odločimo za to.

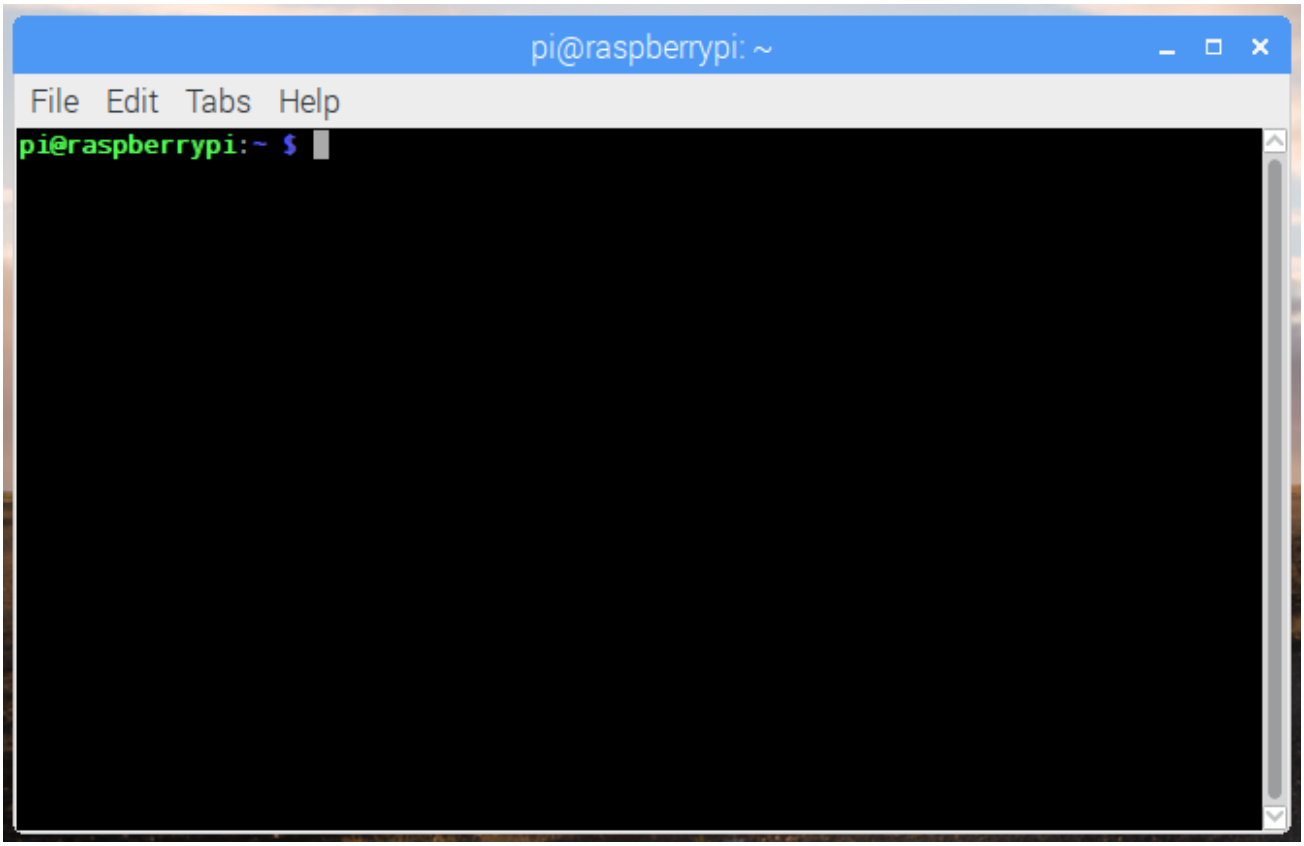
Slika 5: Arduino IDE

V ta IDE je možna vključitev eksternih knjižnic, ki razširijo funkcionalnost. Za svoje namene sem dodal knjižnico PubSubClient, ki omogoča uporabo protokola MQTT za pošiljanje in prejemanje podatkov od drugih naprav.

Dodan je tudi vmesnik za pisanje programa na ESP, saj to ni mogoče na tovarniški verziji IDE-ja.

2.3.1.3 Raspberry Pi Terminal

Raspbian ima kot večina OS-ov tudi svojo konzolo, preko katere lahko poženemo ukaze. To tehnično gledano ni razvojno okolje, a vseeno mi je prišlo prav za implementacijo MQTT protokola in testiranje izvedbe.

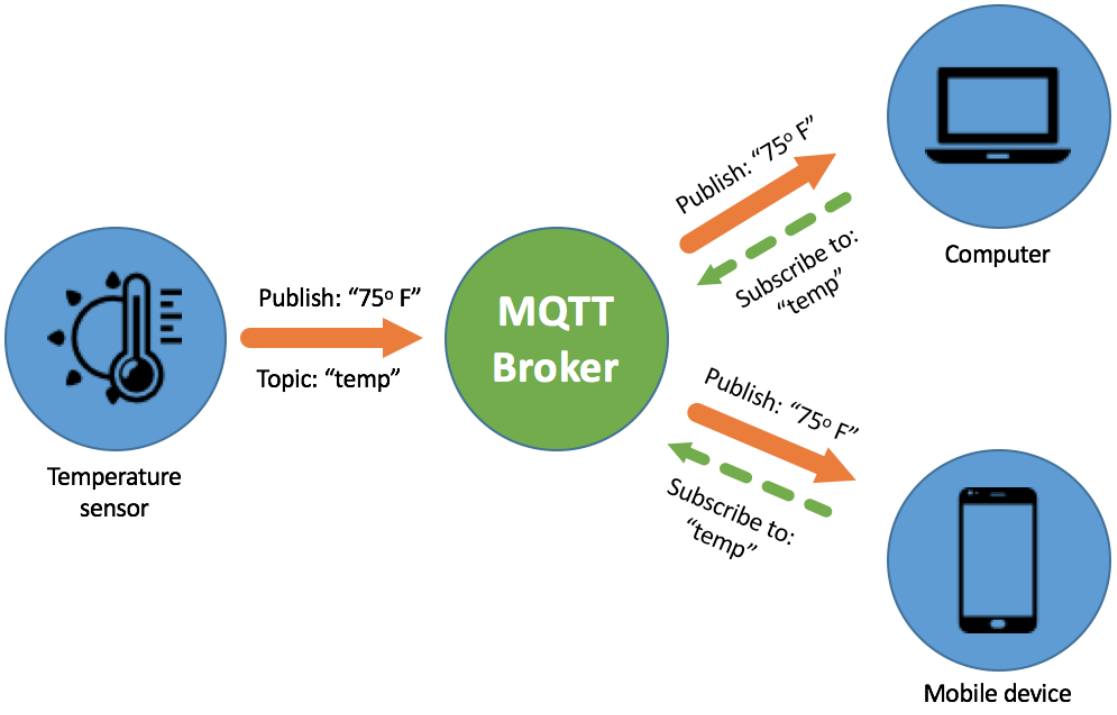


Slika 6: Raspbian konzola

2.3.2 Protokol MQTT

MQTT (MQ Telemetry Transport) je protokol za pošiljanje in prejemanje podatkov po internetni/intranetni povezavi. Sprva je bil razvit za komunikacijo po puščavi, kjer so črpali nafto. Tam so bile povezave ustvarjene s sateliti, kar je pomenilo, da je bilo pošiljanje podatkov drago. Večina protokolov se zaradi svoje relativne počasnosti, požrešnosti in velikosti paketov ni dobro obnesla. Nato sta Andy Stanford in Arlen Nipper razvila MQTT, »lažjo«, učinkovitejšo ter manjšo rešitev. Ta se je sprva uporabljala za svoj prvotni namen, dandanes pa so jo prevzeli tisti, ki se ukvarjajo z IoT ter avtomacijo doma.

MQTT je sestavljen iz dveh delov: posrednika in večjega števila strank. Posrednik je strežnik, ki prejme vsa sporočila vseh strank in jih posreduje naprej na pravo destinacijo. Stranka je vsaka naprava, ki je sposobna poslati ali prejeti MQTT sporočilo; za to je potrebna primerna knjižnica in SDK.



Slika 7: Primer MQTT povezave

Vir: https://www.innorobix.com/wp-content/uploads/2018/10/MQTT_1.png.

Vsi podatki se pošiljajo oz. posredujejo v hierarhiji »tem«. Ko ima naprava nov podatek, ki ga želi poslati, ga pošlje na IP naslov posrednika in doda podatek o tem, kateri temi je namenjen. Nato posrednik posreduje informacijo vsem strankam, ki so naročene na to temo. Pošiljatelj tako ne potrebuje podatka o številu ali naslovu strank, naročenih na temo, stranke, naročene na specifično temo, pa ne potrebujejo naslova pošiljatelja, ki objavlja te podatke. To močno poenostavi omrežje in povezave, saj so vse naprave povezane v eni točki - v posredniku.

Najmanjše MQTT sporočilo je lahko veliko samo dva bita, kar je izjemno malo. Kontrolno sporočilo lahko pošlje do 256 MB podatkov, v kolikor je to potrebno.

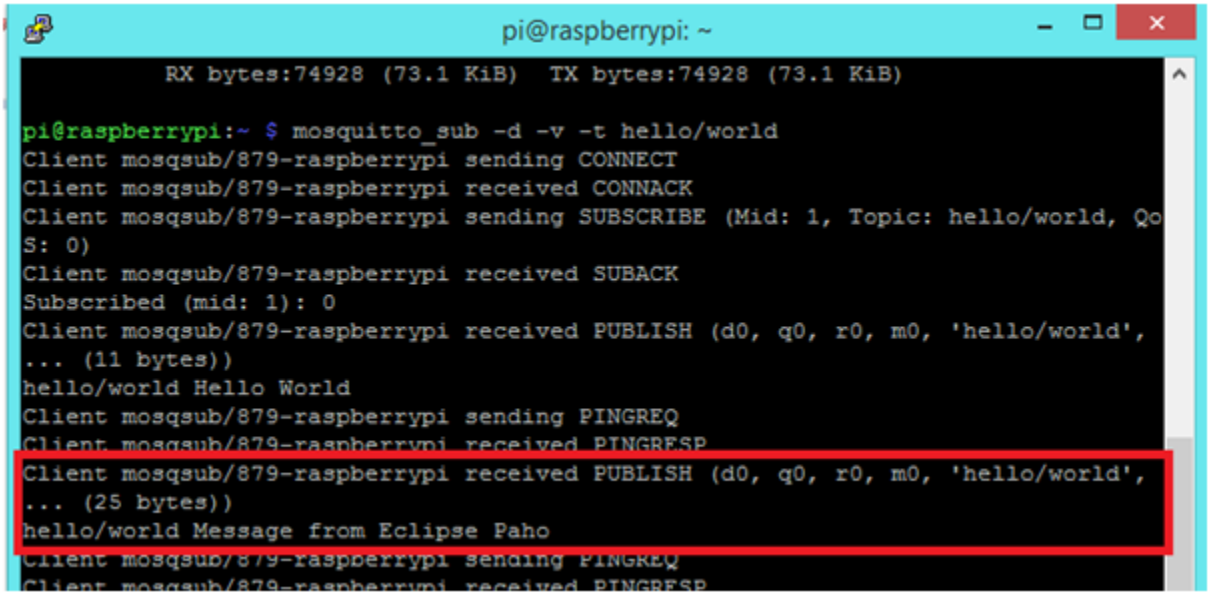
MQTT se zanaša na TCP protokol za pošiljanje podatkov. Obstaja tudi alternativna verzija MQTT-SN, ki je prirejena za uporabo preko drugih protokolov, kot sta UDP ter Bluetooth.

Edina težava pri tem protokolu je varnost, saj podatki niso kriptirani. Za enkripcijo lahko poskrbi TCP, ki lahko zaščiti podatke pred prestrežbo in nedovoljenim branjem. V kolikor se to ne nastavi, se podatki pošiljajo v obliki navadnega besedila in so berljivi vsem.

2.3.3 Mosquitto posrednik

Za MQTT ni uradnega posrednika. Vsak od neuradnih ima prednosti in slabosti; odločil sem se za Eclipse Mosquitto, ker je odprtokodni in podpira vse verzije MQTT-ja do 5.0.

Mosquitto sem naložil na RPi in mu dodelil statičen IP naslov za lažjo povezavo.



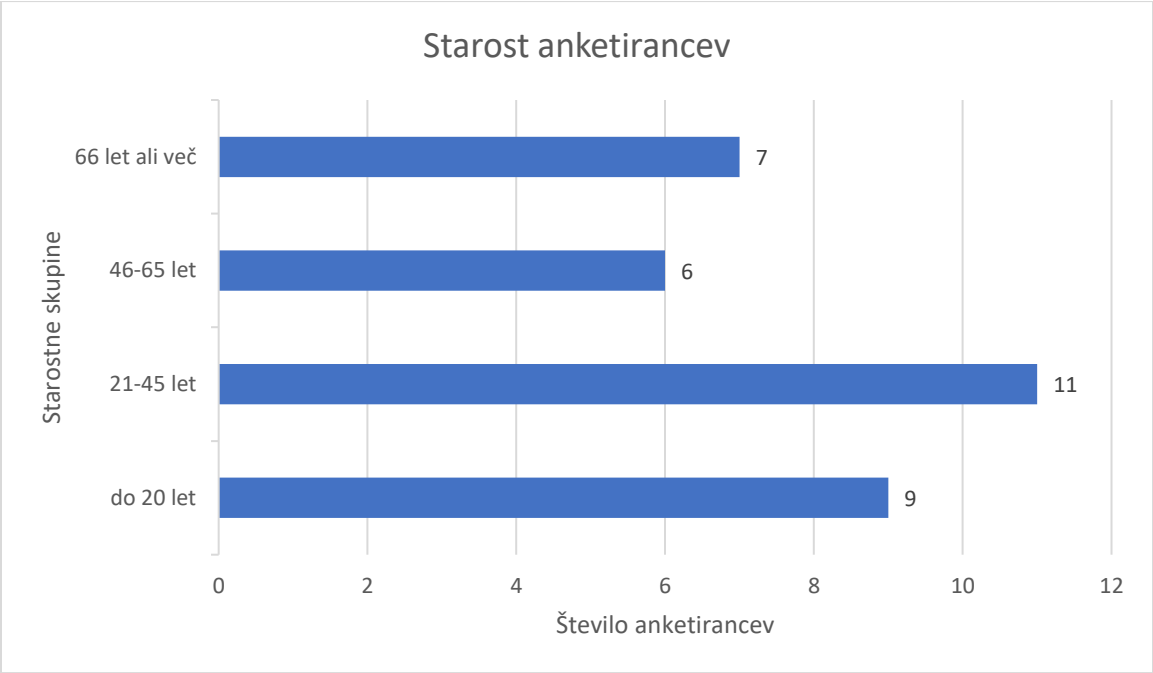
Slika 8: Primer izpisa poslanih podatkov

V kolikor se preko Mosquitta registriramo kot stranka, lahko tudi sami preizkusimo delovanje in vidimo poslane podatke.

3 Praktični del

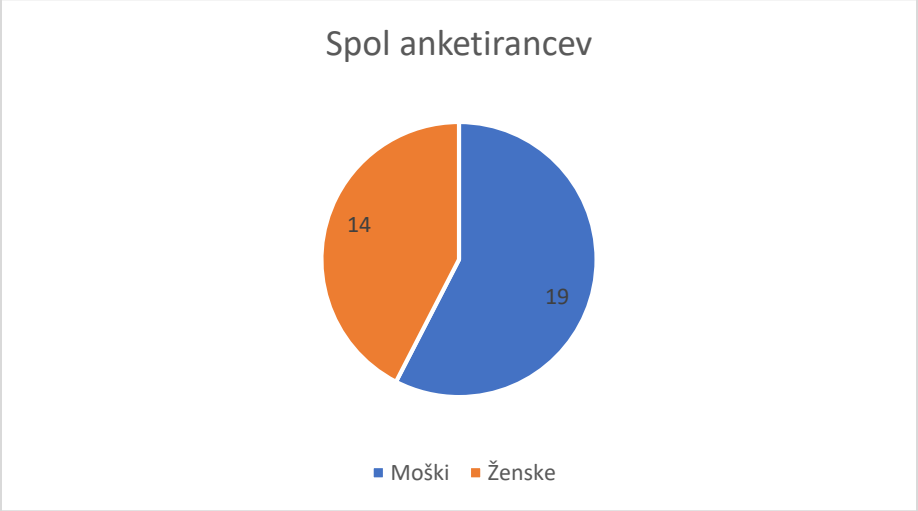
3.1 Anketiranje

Cilj anketiranja je bil, da ugotovim, kako se različne starostne skupine odzivajo na sistem, preko katerega lahko glasovno upravljaš dom. Anketirancev je bilo 33. Ker sem anketo razdelil, sem lahko anketiral ljudi različnih starostnih skupin: od najstnikov do upokojencev. To mi je omogočilo širok pogled, s tem pa sem zbral zanimive odgovore.



Graf 1: Starost anketirancev

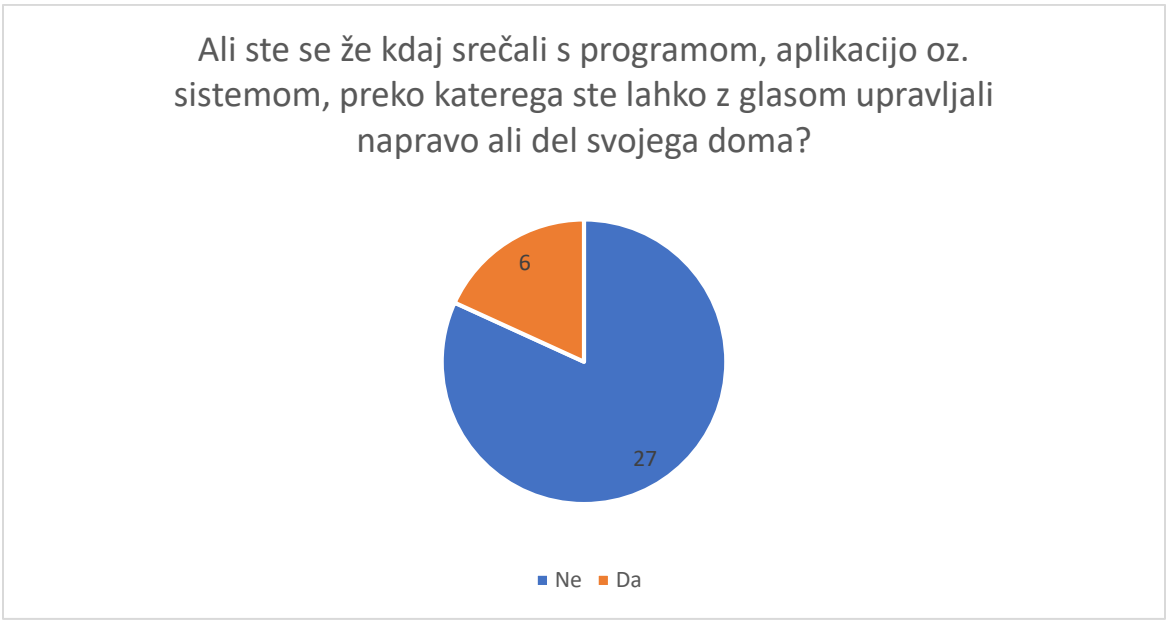
Največ anketirancev je bilo starih 21–45 let; ti so bili tudi najbolj izpostavljeni tehnološkemu napredku skozi čas in bili zadnja generacija, v kateri se je večina brez težav prilagodila novi tehnologiji. Najmanj je bilo starih 46–65, ki so vseeno bolj verzirani v uporabi tehnologije kot anketiranci iz skupine 66+, a vseeno manj kot tisti v skupinah mlajših.



Graf 2: Spol anketirancev

Spol anketirancev je bil prav tako dobro porazdeljen. Več je bilo moških, kar naj ne bi vplivalo na rezultate, saj se vsi na novosti odzivamo podobno ne glede na spol.

Pri prvem vprašanju *Ali ste se že kdaj srečali s programom, aplikacijo oz. sistemom, preko katerega ste lahko z glasom upravljali napravo ali del svojega doma?* sem nameraval določiti, koliko ljudi že dejansko ima izkušnje s takšno tehnologijo ali pa se je z njo vsaj srečalo.



Graf 3: Ali ste se že kdaj srečali s programom, aplikacijo oz. sistemom, preko katerega ste lahko z glasom upravljali napravo ali del svojega doma?

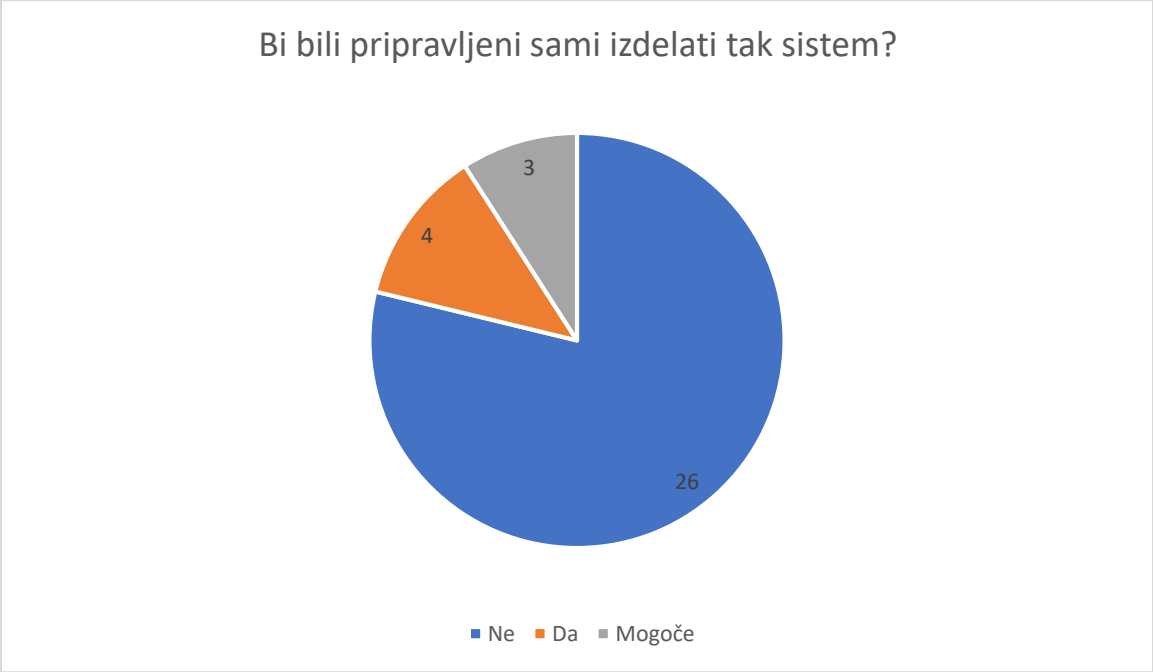
Večina se kot pričakovano s takšnim sistemom še ni srečalo oz. ga ni uporabilo niti enkrat. To sem pričakoval, ker je takšnih sistemov malo, tisti, ki so, so večinoma prodajani in vgrajeni na drugih trgih (ZDA, VB).

Sledilo je vprašanje *Bi v svoj dom vgradili tak sistem?,* ki me je zanimalo zaradi slutnje, da večina takšni tehnologiji ne bi zaupala. Nekateri imajo težave že pri uporabi mobilnih telefonov, kaj šele pri takšnem sistemu. Dodatno oviro predstavlja to, da je program za polno funkcionalnost potrebno uporabljati v angleščini.

Graf 4: Bi v svoj dom vgradili tak sistem?

Rezultati so me presenetili, saj bi več kot četrtina anketirancev bila pripravljena uporabiti tak sistem v svojem domu, le slaba polovica ne bi. Slaba četrtina ostaja neodločena in je odgovorila z *Mogoče,* kar pomeni, da so pripravljeni preizkusiti in se nato odločiti, saj verjetno nimajo predstave, kako bi to izgledalo in vplivalo na njihov vsakdan in rutino.

Zadnje vprašanje je bilo namenjeno za bolj tehnično usmerjene osebe, čeprav so odgovorili vsi. Zanimalo me je namreč, kakšen delež anketirancev bi bil pripravljen takšen sistem za glasovni nadzor doma izdelati sam. Pričakoval sem, da bo velika večina odgovorila ne.

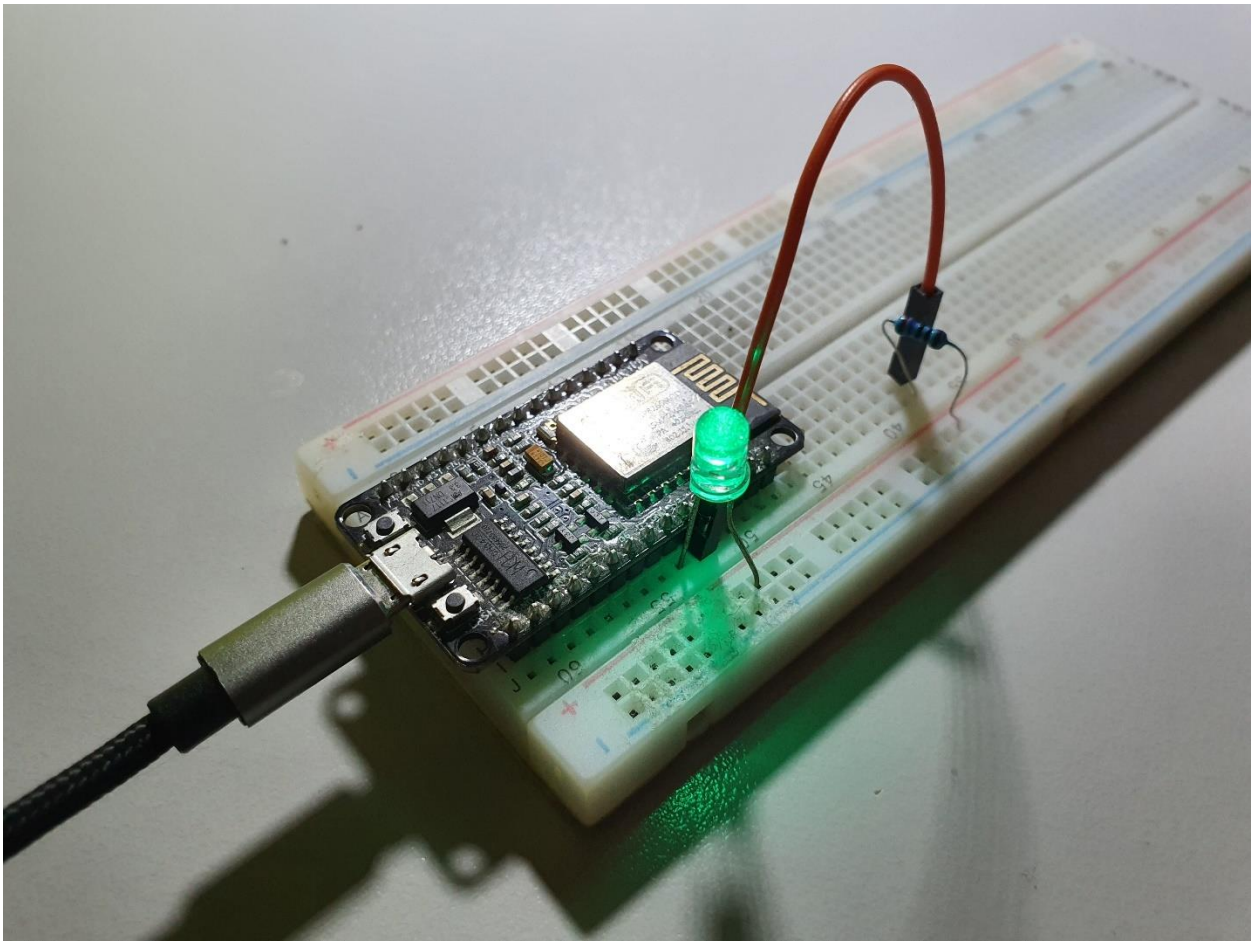


Graf 5: Bi bili pripravljeni sami izdelati tak sistem?

Moja pričakovanja so bila tudi potrjena ob pregledu rezultatov, saj so ti pokazali, da večina ljudi takšnega sistema ne bi zgradila sama. To je razumljiv odgovor, saj je za takšno delo potrebno tehnično znanje, ki ga večina ne premore. Morda bi preizkus takšnega sistema nekaterim spremenil mnenje in bi ga vsaj poskusili sestaviti sami.

3.2 Preizkus strojne opreme

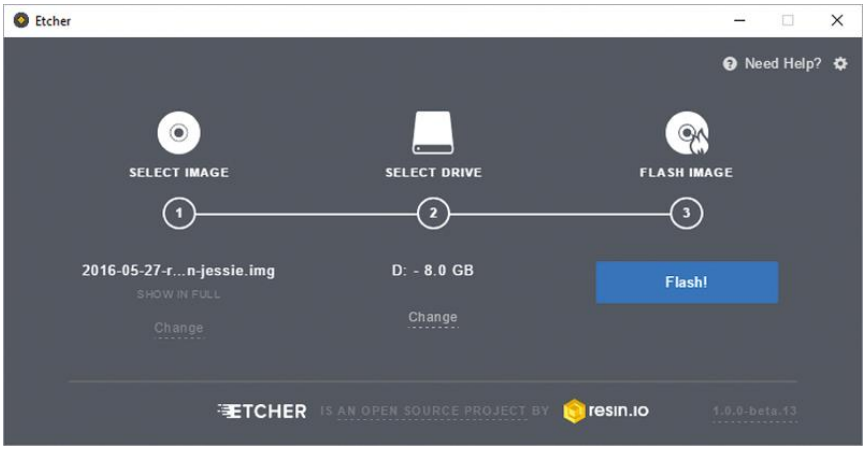
Strojno opremo sem že imel pridobljeno, saj sem se že ukvarjal z izgradnjo podobnega sistema. Edina težava je bila ta, da je bilo vse dalj časa izpostavljeno soncu ter nisem bil prepričan, če deluje in če ima ESP naložen pravi firmware, RPi pa pravi OS. Odločil sem se, da bom najprej preizkusil ESP. To sem storil tako, da sem ga priklopil na računalnik ter nanj naložil program, ki naj bi prižgal in ugasnil LED.



Slika 9: ESP LED Test

Sprva sem imel težave pri povezavi; te sem rešil z zamenjavo kabla. Nato se je pojavila še težava, ker se program ni uspešno naložil na ESP. S pomočjo dokumentacije ter forumov sem ugotovil, da imam na Arduino IDE napačno nastavljen model ESP-ja. Ko sem to popravil, je oboje delovalo in LED je utripala.

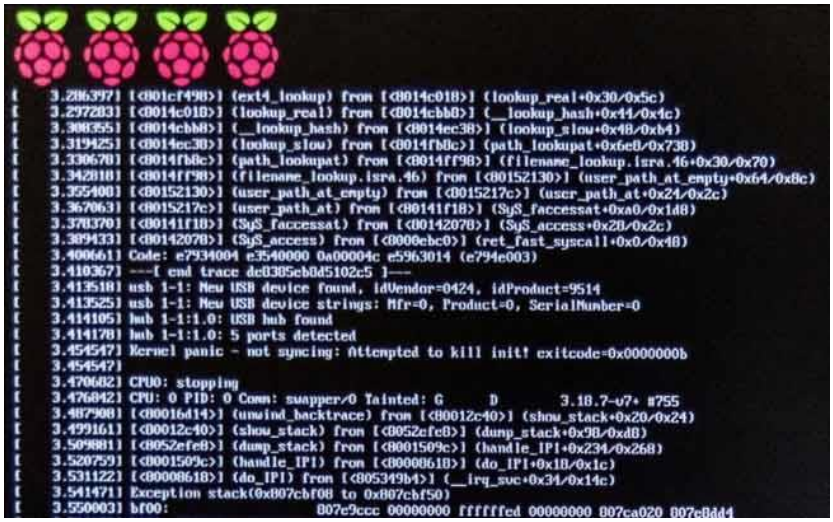
Nato je prišel na vrsto za preizkus RPi. Ker sem ga pred tem projektom že uporabljal, sem vedel, da bom moral ponovno naložiti Raspbian, ker sem želel prazen OS brez navlake. To sem storil tako, da sem prenesel image file Raspbiana ter ga namestil s pomočjo programa balenaEtcher.



Slika 10: Etcher program

Vir: https://images.ctfassets.net..

Ob vklopu RPi sem moral biti zelo pozoren, saj odvzem napajanja med uporabo SD kartice skoraj zagotovo pomeni korupcijo podatkov ter OS-a in v najhujših primerih tudi fizično uničenje SD kartice (to sem že izkusil pri prejšnjem projektu). Ob zagonu nas pričaka ekran, kjer se preverjajo vse nastavitve ter funkcije za pravilno delovanje.



Slika 11: Zagon RPi

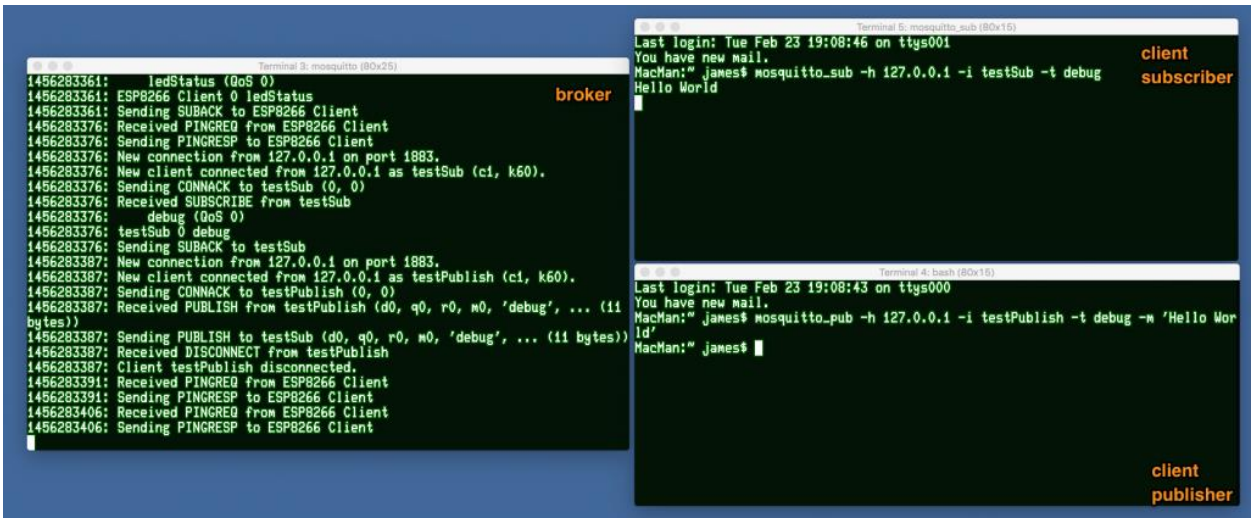
3.3 Nalaganje programov

Raspbian ima sicer že vrsto tovarniško naloženih programov (v kolikor se ob prenosu zanje odločimo), nima pa vseh, ki sem jih potreboval. Najbolj pomemben in edini, ki ga je RPi resnično potreboval, je Mosquitto.

Tega sem naložil tako, da sem odprl konzolo ter vanjo napisal sledečo kodo:

```
pi@raspberry:~ $ sudo apt update
pi@raspberry:~ $ sudo apt install -y mosquitto mosquitto-clients
```

Po tem je bil Mosquitto uspešno naložen, moral sem ga le še preizkusiti. To sem storil tako, da sem odprl dve konzolski okni. V prvem sem se naročil na temo »Test«, v drugem pa sem poslal sporočilo »Hello World« v to temo.



Slika 12: Preizkus Mosquitta

Vir: https://www.baldengineer.com/mqtt-tutorial.html.

Ker je sporočilo uspešno prišlo do drugega okna, sem lahko sklepal, da Mosquitto pravilno deluje. Nadaljnjega dela preko RPi nisem imel, uporabljal sem ga le še za pregledovanje prometa.

3.4 Priprava programa za ESP-8266

Ker sem imel Mosquitto že delujoč, sem se odločil, da preizkusim še drugi del strojne opreme – ESP-8266. Na tega sem naložil program, ki naj bi se ob zagonu naročil na temo ledStatus. Ko je po tej temi prišlo sporočilo »1«, je program določen pin prižgal ter s tem prižgal tudi LED, kar je pomenilo, da ta deluje. Obratno je veljalo za primer poslane »0«.

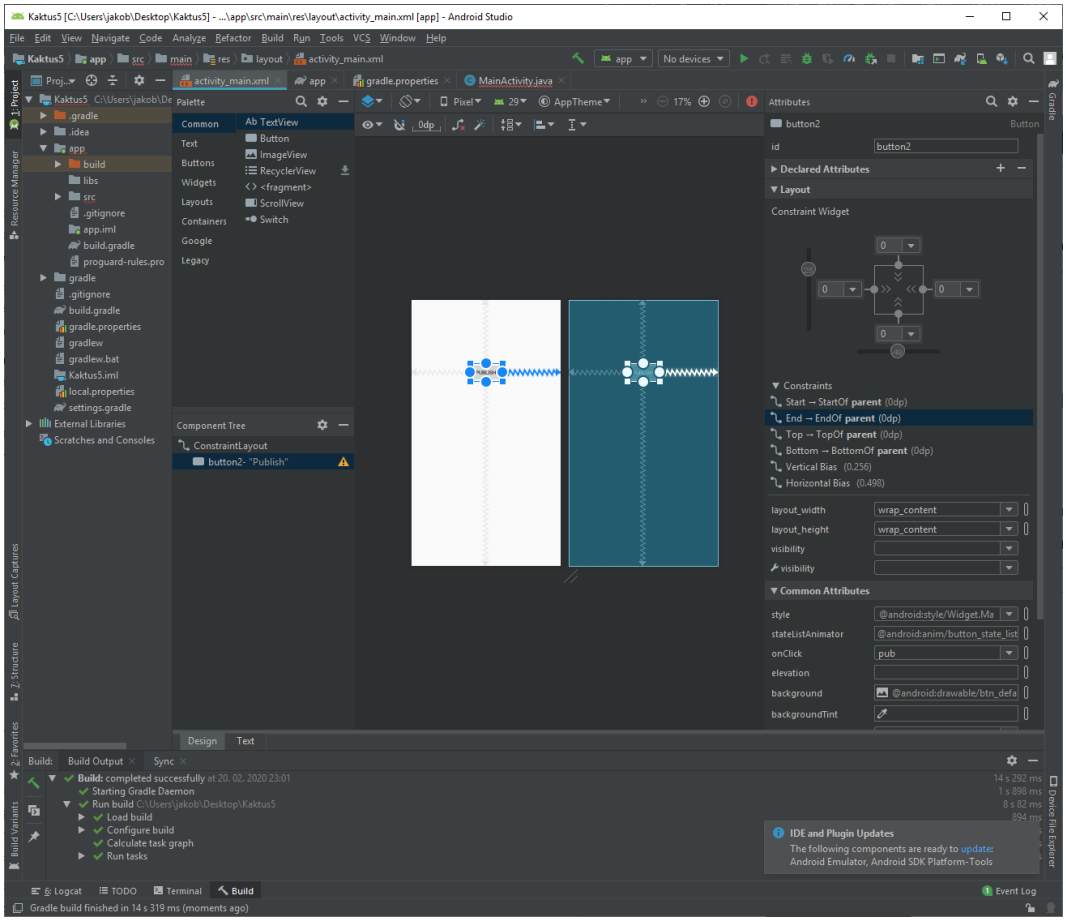
Slika 13: Del programa za ESP

Po tem, ko se je program uspešno naložil, je ta tudi pravilno deloval in prižigal LED glede na prejeto sporočilo. Ker še nisem imel delujoče aplikacije za mobilni telefon, sem to preizkusil preko konzole na RPi, kjer sem opazoval podatke ter poslal sporočilo ESP-ju.

3.5 Priprava programa za telefon

Ker je celoten smisel te naloge glasovno upravljanje, je aplikacija za mobilni telefon morala biti izdelana kvalitetno ter zanesljivo in enostavno. Za razvoj te aplikacije sem uporabil IDE android studio. Pomagal sem si z nekaj vodiči ter blogi.

Odločil sem se, da bom zaradi lažjega razvijanja aplikacijo sestavil sprva v dveh delih in ju nato združil. Prvi del je bil pošiljanje MQTT sporočila na ESP direktno iz telefona. To sem dosegel s pomočjo Paho android service. To je knjižnica, ki omogoča pošiljanje MQTT sporočil. Najprej sem moral to knjižnico uvoziti, kar mi je povzročalo težave, ker se je zanašala na starejši element iz knjižnice, ki se naj ne bi več vključeval v nove projekte. To sem popravil tako, da sem uvozil še »legacy« podporo iz knjižnice AndroidX. Nato je vse delovalo in ob pritisku gumba sem lahko poslal sporočilo na določen IP naslov.

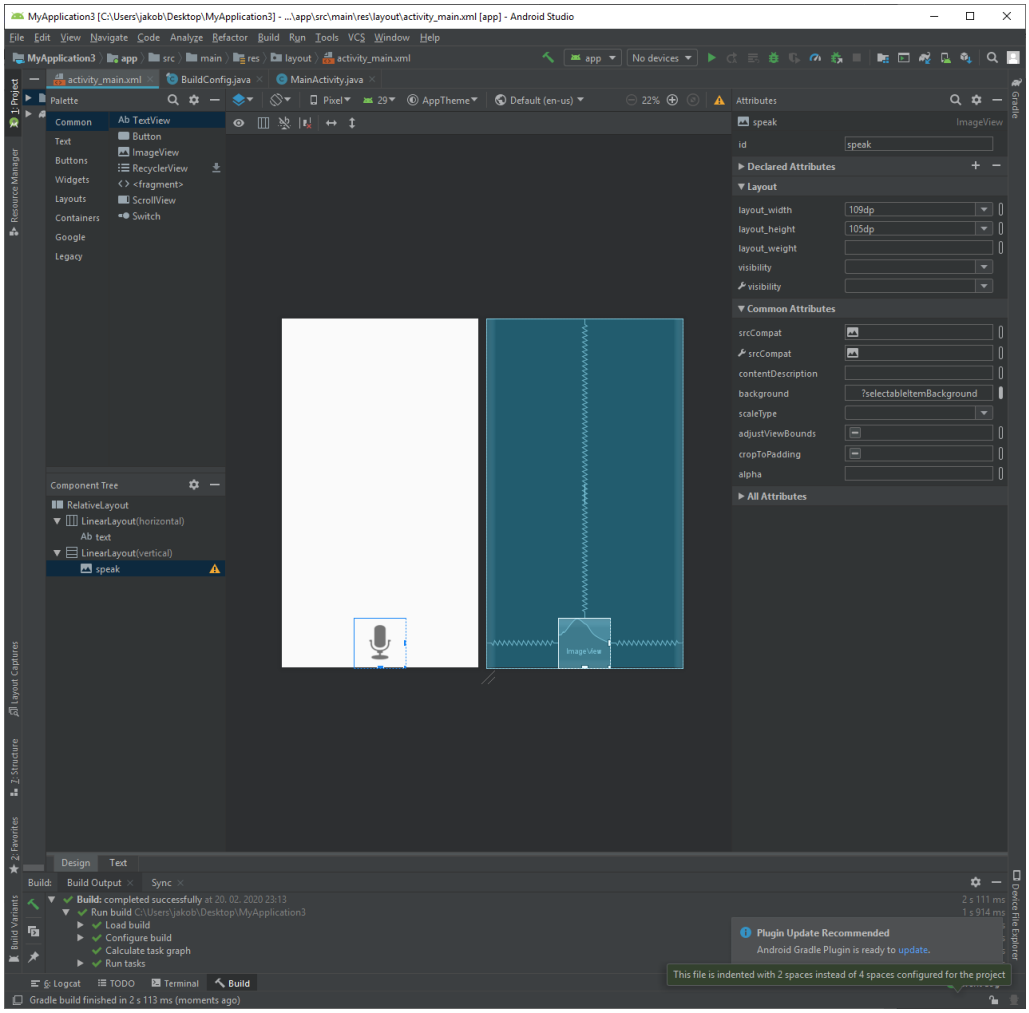


Slika 14: Izdelava aplikacije, 1. del

Drugi del aplikacije je bil prepoznava glasu in pretvorba le-tega v besedilo. K temu problemu bi lahko pristopil z več strani. Prvi bi bil, da bi izdelal svojo umetno inteligenco, ki bi bila zmožna natančno prepoznati besede ter ji razbrati in pretvoriti besedilo. Za to bi potreboval veliko več znanja na tem področju, a kljub temu ne bi izdelal nič zanesljivega.

Rešitev, za katero sem se odločil sam, je bila enostavnejša. Google omogoča dostop do njihovih aplikacij za prepoznavo glasu. Ker so znani kot eni izmed boljših na tem področju, njihovi programi pa se iz leta v leto izboljšujejo, sem to uporabil tudi sam.

Za gumb sem dodal enostavno sliko mikrofona, sprva pa sem besedilo le izpisal na ekranu in z njem nisem storil ničesar.



Slika 15: Izdelava aplikacije – 2. del

Nato sem oba dela programa le še združil ter dodal nekaj pogojnih stavkov, ki omogočijo, da ob besedah »turn on the light« program pošlje na ESP število 1 (kar pomeni prižig), ob besedah »turn off the light » pa pošlje število 0 (kar pomeni ugasnitev).

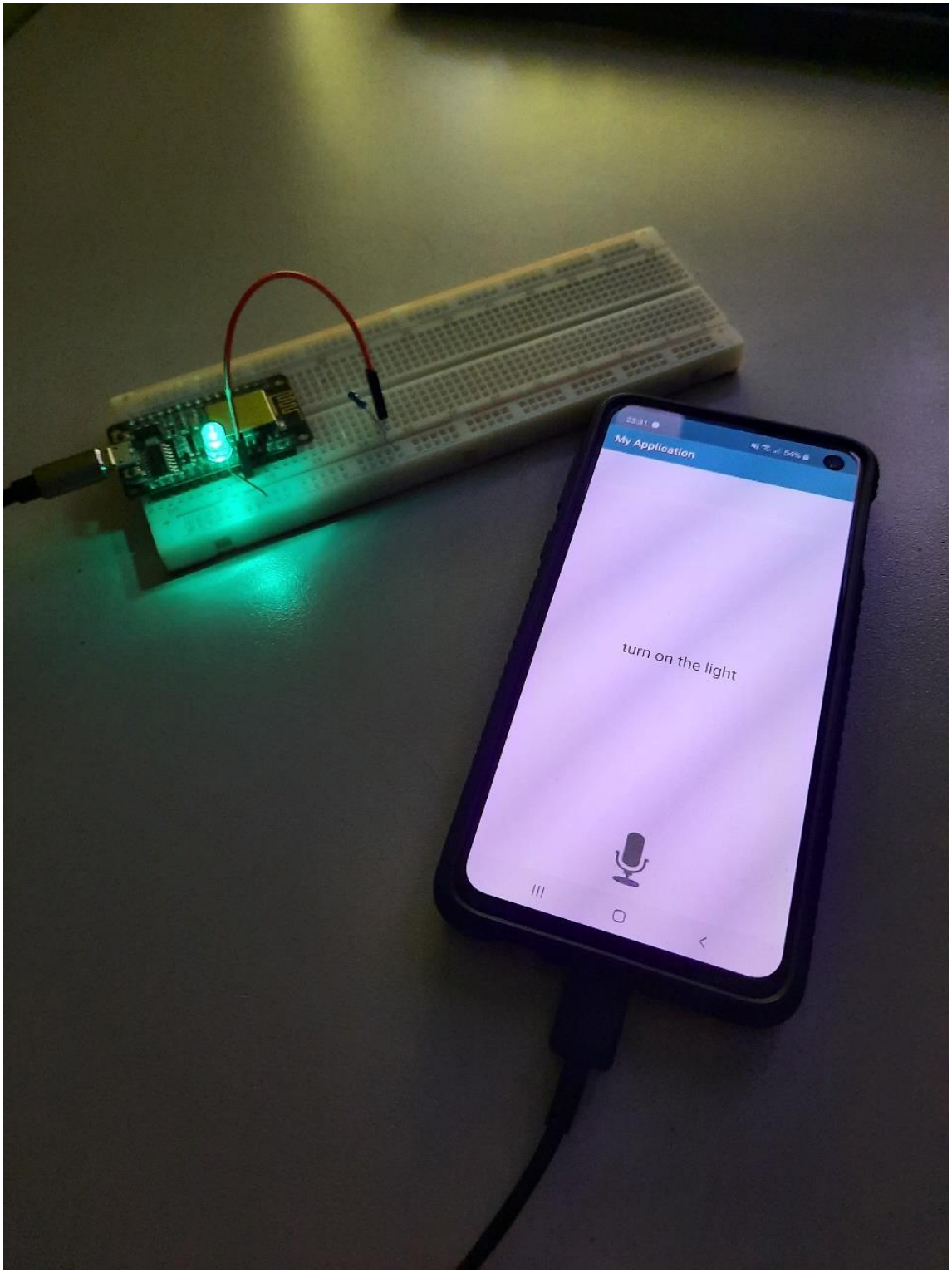
3.5.1 Uporaba angleščine za ukaze

Seveda bi bilo bolj logično, da bi tak program uporabljali v slovenščini, saj je naš materni jezik. Pri uporabi prepoznave glasa sem naletel na težavo pri drugih jezikih, saj je najbolje podprta angleščina. Ta neuradno velja za uradni jezik programiranja, kar vpliva na prepoznavanje in pretvarjanje izgovorjenih besed v besedilo.

V slovenščini bi tak program sicer bil izvedljiv, ampak bi bil veliko manj natančen in bolj nagnjen k napačni razpoznavi besed.

3.6 Preizkus sistema

Edina naloga, ki je še ostala, je bil preizkus sistema za glasovno upravljanje doma, v tem primeru glasovno upravljanje luči. Program sem naložil na telefon, ESP sem prižgal in na RPi pognal Mosquitto.



Slika 16: Preizkus aplikacije

4 Razprava

Področje pametnega upravljanja doma se bo še naprej hitro razvijalo.

V raziskovalni nalogi sem poglobil znanje o protokolu MQTT ter medsebojno povezanost naprav. Nadgradil sem znanje programskih jezikov Java ter C++. Kar sem želel ustvariti, sem uspel, a menim, da je še velik potencial za razširjenje ter poglobitev sistema v nekaj, kar bi uporabniku olajšalo vsakdan.

4.1 Hipoteze

Hipoteze, postavljene na začetku raziskovalne naloge, sem temeljito raziskal in preveril.

***Hipoteza 1**: Izdelava pametnega sistema za upravljanje doma je težka, saj se malokdo ukvarja s tem.*

To hipotezo sem ovrgel, saj sem odkril veliko novih forumov, blogov ter spletnih strani, ki se s to temo ukvarjajo in tudi aktivno pišejo vodiče, članke ter prispevke.

***Hipoteza 2**: Podlago za znanje imam solidno, a se bom moral naučiti več o protokolih, strojni opremi in delovanju takšnega sistema.*

To hipotezo sem potrdil, saj sem področje že približno poznal, a ne dovolj za izdelavo sistema, kot sem si ga zadal. Poglobiti sem moral še znanje izdelave aplikacije za mobilni telefon.

***Hipoteza 3**: Svoj sistem bom zlahka razširil ter dodal nove funkcije poleg osnovnih.*

Hipotezo sem ovrgel, saj je razširitev takšnega sistema precej bolj zahtevna, kot sem predvideval. Morda bom v prihodnosti mislil drugače, a trenutno to presega moje znanje.

***Hipoteza 4**: Večina anketirancev bi takšen sistem uporabljalo in vključilo v svoj vsakdan.*

Hipotezo sem ovrgel, saj je skoraj polovica anketirancev odgovorila, da ga ne bi uporabljala vsak dan. S pomočjo ozaveščanja javnosti ter izobrazbe bi se to lahko spremenilo**.**

5 Zaključek in smernice za nadaljnje delo

Med ustvarjanjem izdelka in pisanjem raziskovalne naloge sem pridobil veliko novega znanja na tem področju, spoznal nekaj novih tehnologij ter poglobil obstoječe znanje. Naučil sem se razvijanja aplikacij za Android in se pri delu tudi zabaval ter užival v iskanju rešitev in postavljanju novih ciljev. Izkušnja izdelave raziskovalne naloge je bila zelo zanimiva, po drugi strani pa je bilo tudi precej zahtevno.

Svojo glavno idejo prižiganja luči oz. katerekoli naprave s svojim glasom sem uresničil, hkrati pa so se mi posvetile nove ideje in razširitve za sistem, ki ga bom razvijal tudi po končani nalogi, saj se mi to področje zdi zanimivo in v njem vidim potencial. V prihodnje si želim povezati več senzorjev, ki bi poročali o trenutnem stanju v hiši ter omogočili odziv na dogodke.

Delo s to strojno in programsko opremo bi priporočal vsakomur, ki se mu zdi področje avtomacije doma zanimivo, saj menim, da je ta sistem odličen temelj za nadaljnje delo, pomembno pa je tudi to, da je vse cenovno dostopno. Čeprav se mi je še pred nekaj leti takšen sistem zdel neznansko kompleksen in nemogoč za lastno izdelavo, sem sedaj spoznal, da se področje razvija hitreje, kot sem pričakoval. Nekega dne bomo verjetno vsi imeli hišo ali stanovanje opremljeno s takšnim sistemom; zakaj ga ne bi zgradili sami?

6 Bibliografija

Anon., 2011. *Mosquitto.* [Elektronski]
Dosegljivo na: https://mosquitto.org/blog/2011/11/android-mqtt-example-project/
[Poskus dostopa 10. 1. 2020].

Anon., 2015. *Arduino PubSubClient.* [Elektronski]
Dosegljivo na: https://www.hivemq.com/blog/mqtt-client-library-encyclopedia-arduino-pubsubclient/
[Poskus dostopa 10. 1. 2020].

Anon., 2017. *MQTT.* [Elektronski]
Dosegljivo na: https://github.com/mqtt/mqtt.github.io/wiki
[Poskus dostopa 5. 1. 2020].

Anon., 2019. *Android 10.* [Elektronski]
Dosegljivo na: https://www.android.com/android-10/
[Poskus dostopa 4. 1. 2020].

Anon., 2019. *Android crashes.* [Elektronski]
Dosegljivo na: https://developer.android.com/topic/performance/vitals/crash
[Poskus dostopa 20. 12. 2019].

Anon., 2019. *android.com.* [Elektronski]
Dosegljivo na: https://developer.android.com/jetpack/androidx/migrate
[Poskus dostopa 14. 1. 2020].

Anon., 2019. *AndroidX Overview.* [Elektronski]
Dosegljivo na: https://developer.android.com/jetpack/androidx
[Poskus dostopa 14. 1. 2020].

Anon., 2020. *Android Studio release notes.* [Elektronski]
Dosegljivo na: https://developer.android.com/studio/releases/index.html
[Poskus dostopa 5. 1. 2020].

Anon., 2020. *ESP-8266.* [Elektronski]
Dosegljivo na: https://en.wikipedia.org/wiki/ESP8266
[Poskus dostopa 10. 1. 2020].

Anon., 2020. *Home automation.* [Elektronski]
Dosegljivo na: https://en.wikipedia.org/wiki/Home_automation
[Poskus dostopa 10. 1. 2020].

Lewis, J., 2019. [Elektronski]
Dosegljivo na: https://www.baldengineer.com/mqtt-tutorial.html
[Poskus dostopa 14. 12. 2019].

Light, R., 2016. [Elektronski]
Dosegljivo na: https://mosquitto.org/man/mosquitto-8.html
[Poskus dostopa 20. 12. 2019].

Novisu, T., 2016. *playing with MQTT by Android.* [Elektronski]
Dosegljivo na: https://www.youtube.com/watch?v=BAkGm02WBc0
[Poskus dostopa 4. 2. 2020].

Raff, S., 2017. [Elektronski]
Dosegljivo na: https://github.com/mqtt/mqtt.github.io/wiki
[Poskus dostopa 20. 12. 2019].

Sciker, 2017. *How to Connect with MQTT on Android.* [Elektronski]
Dosegljivo na: https://www.youtube.com/watch?v=I-XyqsxaGQc
[Poskus dostopa 14. 1. 2020].

Stanford-Clark, A., 2013. *mqtt.org.* [Elektronski]
Dosegljivo na: http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf
[Poskus dostopa 4. 2. 2020].

Thakur, A., 2019. *How to integrate Android Speech To Text.* [Elektronski]
Dosegljivo na: https://www.tutorialspoint.com/how-to-integrate-android-speech-to-text
[Poskus dostopa 10. 1. 2020].

7 Priloge

7.1 Anketni vprašalnik

Pozdravljeni, sem Jakob Dorn, dijak končnega letnika SŠ za kemijo, elektrotehniko in računalništvo, smer tehnik računalništva. Letos sem se odločil za izdelavo raziskovalne naloge na temo upravljanja doma in naprav v njem z lastnim glasom. V okviru te raziskovalne naloge bi rad raziskal tudi ozaveščenost javnosti o tej tehnologiji, njeno razširjenost in koliko ljudi jo je pripravljeno uporabiti in vključiti v del svojega življenja. Zato vas prosim, da odgovorite na naslednja vprašanja.

1. Ali ste se že kdaj srečali s programom, aplikacijo oz. sistemom, preko katerega ste lahko z glasom upravljali napravo ali del svojega doma?
    a. Da
    b. Ne
2. Bi v svoj dom vgradili tak sistem?
    a. Da
    b. Ne
    c. Mogoče
3. Bi bili pripravljeni sami izdelati tak sistem?
    a. Da
    b. Ne
    c. Mogoče
4. Spol:
    a. Moški
    b. Ženski
5. Starostna skupina:
    a. Do 20 let
    b. 21–45 let
    c. 46–65 let
    d. 66 let ali več

IZJAVA*

Mentor Boštjan Resinovič v skladu z 20. členom Pravilnika o organizaciji mladinske raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi z naslovom Glasovno upravljanje doma z RPi, katere avtor je Jakob Dorn:

- 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, 4.6.2020

žig šole

Podpis mentorja

Podpis odgovorne osebe

*

POJASNILO

V skladu z 20. 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.