Automatizacija pametne kuće uz primjenu Arduina i Linux usmjerivača

Glavinić, Endi

Undergraduate thesis / Završni rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:137:559961

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-23**

Repository / Repozitorij:

Digital Repository Juraj Dobrila University of Pula

Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

ENDI GLAVINIĆ

AUTOMATIZACIJA PAMETNE KUĆE UZ PRIMJENU ARDUINA I LINUX USMJERIVAČA

Završni rad

Pula, veljača, 2018.

Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

ENDI GLAVINIĆ

AUTOMATIZACIJA PAMETNE KUĆE UZ PRIMJENU ARDUINA I LINUX USMJERIVAČA

Završni rad

JMBAG: 0303033068, izvanredni student
Studijski smjer: Informatika
Predmet: Softversko inženjerstvo
Znanstveno područje: Društvene znanosti
Znanstveno polje: Informacijske i komunikacijske znanosti
Znanstvena grana: Informacijski sustavi i informatologija
Mentor: doc. dr. sc. Tihomir Orehovački

Pula, veljača, 2018.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani Endi Glavinić, kandidat za sveučilišnog prvostupnika informatike ovime izjavljujem da je ovaj Završni rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima, stečenom znanju tokom studiranja na ovoj Ustanovi, te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Ovom prilikom htio bih se zahvaliti svojoj obitelji, koja je bila najveća potpora tokom studiranja.

Student

_____

U Puli, 19. veljače, 2018. godine

IZJAVA

o korištenju autorskog djela

Ja, Endi Glavinić dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom AUTOMATIZACIJA PAMETNE KUĆE UZ PRIMJENU ARDUINA I LINUX USMJERIVAČA koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____ (datum)

Potpis

_____

Sadržaj

1. Uvod 2
2. Što je kućna automatika? 3
3. Funkcioniranje sustava 4
4. Protokol 5
5. Sustav automatizacije MojDom 7
6. MojDom Hardver 8
6.1. Odabir hardvera 8
6.2. Arduino 9
6.3. Arduino Nano 10
6.4. OpenWRT ..... **Error! Bookmark not defined.**
7. Podešavanje hardvera 14
7.1. Usmjerivač i instalacija OpenWrt-a 14
7.2. Komunikacija preko serijskog porta 14
7.3. Komunikacija preko SSH protokola 15
8. Softver 17
8.1. Odabir softvera 17
8.2. Instalacija esencijalnog softvera na OpenWrt 18
8.3. Arduino softver 20
8.4. MojDom softver 22
8.5. Osnovne skripte za komunikaciju 22
8.6. Web sučelje 25
8.7. Aplikacija za pametne telefone 27
9. Zaključak 31
Bibiliografija ..... **Error! Bookmark not defined.**
Popis slika 36
Popis tablica 37
Sažetak 38
Abstract 38

1. Uvod

U bliskoj budućnosti ljudi će živjeti na način koji smo navikli vidjeti u futurističkim filmovima sa kraja prošlog stoljeća uz iznimku da nas neće posluživati roboti već kuća sama. Kućanski uređaji će komunicirati između sebe. Ako ogladnimo „javiti“ ćemo sustavu koji će provjeriti što ima u frižideru ili ostavi te predložiti što bi mogli pojesti ili eventualno na osnovi namirnica te online dostupnih recepata, predložiti jelo koje bi mogli skuhati. Naravno, pećnica, štednjak ili mikrovalna automatski će podesiti temperaturu i vrijeme pripremanja hrane prema receptu kako bi jelo bilo što ukusnije. „Aparat za kavu će pripremiti kavu prema osobnim preferencijama pojedinih ukućana“ (CastelarArticles.com, n.d.) a „hladnjak će javiti da nema više mlijeka prije nego ostanemo bez njega. Možemo lako zamisliti da nećemo imati potrebu sastavljati listu za kupovinu, već će kuća na osnovi naših navika izraditi istu ili čak naručiti iz online trgovine“ (Srdić, 2016).

Mnogi današnji uređaju postaju „pametni“ ili u najmanju ruku više automatizirani, sustavi grijanja i klimatizacije integrirani na takav način da se slanjem poruke (ili spajanjem na web sučelje kuće) može podesiti termostat prije dolaska kući, kamere koje vam dojavljuju na mobitel tko je ispred kuće, ostavljajući vam mogućnost da tu osobu pustite u kuću, iako niste u blizini.. „Većini to zvuči vremenski daleko, međutim, sustavi pametnih kuća postoje desecima godina. Kod nas su, prema riječima nekoliko integratora, vrhunac doživjeli pred krizu, 2007 godine. Zbog same cjenovne nepristupačnosti, uglavnom su se ugrađivali u luksuzne vile“ (Srdić, 2016). Cjenovna nepristupačnost postojećih komercijalnih rješenja je ujedno problem koji će se nastojati riješiti ovim radom. Koristeći se različitim metodama analizirati će se postojeće stanje, izvesti studija slučaja, kompilirati, statistički obraditi te sintetizirati činjenice s ciljem izrade konačnog projekta koristeći resurse sa što manjim opterećenjem budžeta. Kao što proizlazi iz cilja, svrha projekta je omogućiti otvoreni pristup i cjenovno pristupačnu izradu kućne automatizacije onima koji imaju potrebna znanja da ista primjene u kućnoj (hobi) radinosti.

2. Što je kućna automatika?

U osnovi, kućna automatika je proizvod ili usluga koja uključuje akciju ili obavještavanje u kućnom okruženju, a gdje se događaj zbiva bez ljudske intervencije.

Iako je pristup inovativan on koristi postojeću tehnologiju. Pametni uređaju su zapravo isti oni konvencionalni uređaji sa integriranim računalima koji im daju više funkcionalnosti. Inteligentna ili pametna kuća je opremljena s centralnim sustavom upravljanja, koja osim udobnosti, također služi za optimizaciju potrošnje energije u domu. Na primjer, sustav može prilagoditi temperaturu u skladu s uputama, razinu rasvjete, uključiti i isključiti električne potrošače kao što su stroj za pranje rublja za vrijeme niže tarife električne energije, ventilaciju, upravljati sa roletama ili protupožarnom zaštitom. Također, ovisno o kvaliteti zraka, moguće je upravljati otvaranjem prozora ili uključivanjem ventilacije. Automatsko podizanje i spuštanje roleta te paljenje i gašenje svijetla simulira prisutnost u objektu te se koristi da bi ostavilo dojam prisutnost za vrijeme dok duže vremena nismo prisutni. Takav način moguće je aktivirati npr. jednostavnim zaključavanjem kuće.

Ukoliko trebamo „pametnu utičnicu“, „pametni senzor“, električnu bravu upravljanu putem otiska prsta, možemo to relativno jeftino nabaviti. Problem nastaje jer prilikom instalacije sustava kućne automatike za kuću površine 100m^2 troškovi prelaze nekoliko desetaka tisuća kuna a kod implementacije sustava pametnih kuća i preko sto tisuća kuna. Osim navedenog rastu i troškovi održavanja jer zbog zastarjelog obrazovnog sustava, električari nisu obučeni za rad sa takvim sustavima pa specijalizirani servisi zaračunavaju vrlo visoke naknade

Pametne kuće su „in“, štede vrijeme i energiju. Pripreme vam ugodnu temperaturu tik prije nego što ćete doći, postave rasvjetu ili pak uključe pećnicu. „Savjet za zelenu gradnju u Hrvatskoj barata podacima da emisija CO2 plinova koja se pripisuje građevinskome sektoru iznosi od 35 do čak 45 posto, a uštede koje se mogu postići postavljanjem sustava pametne kuće mogu postići uštedu energije od 40 do 60 posto.“ (Srdić, 2016)

„U ranim sedamdesetima u SAD je bio hit Mr.Coffe, aparat za kavu povezan sa budilicom. Prilikom aktiviranja alarma, vlasnicima je skuhao kavu te se pamti kao jedan od ranijih primjera kućne automatike. Danas, nakon nekoliko desetljeća, samo u kućama hobi majstora možemo pronaći aparat za kavu koji digitalno mjeri temperaturu vode, pritisak, vrijeme potrebno za idealnu količinu kave te po završetku šalje poruku na neki pametni uređaj“ (Riley, 2012, pp. 3-4)“. Koji je razlog tome? Jedan od glavnih razloga mogao bi bit nepostojanje unificiranog protokola kućne automatike. Većina proizvođača svoje protokole štiti pravima, pa za korištenje istih treba kupiti prava što u početku poskupljuje razvoj. Ako neki proizvođač namjerava proizvoditi veliki broj uređaja često se dogodi da mu varijabilni trošak kupnje prava premašuje fiksni trošak proizvodnje vlastitog protokola pa se odlučuje za potonje.

3. Funkcioniranje sustava

Sustav kućne automatizacije ima zadatak automatizirati procese u objektu koji bi inače ručno trebali kontrolirati čovjek. Neki sustavi uključuju sustav grijanja, klimatizaciju, sigurnosne sustave, rasvjetu, sustave prolaza vozila, parlafon i druge sustave koji se mogu integrirati u sustav kućne automatizacije.

Svaki sustav kućne automatizacije ima vlastite ulazne i izlazne uređaje i sustave.

Glavni dio sustava je kontroler koji upravlja podsustavima i ima korisničko sučelje za upravljanje putem pametnog telefona ili drugog uređaja. Regulator također ima unaprijed instalirane definirane scenarije pomoću kojih se sustav upravlja bez utjecaja čovjeka.

- Kontroler ima vlastite ulazne i izlazne uređaje koji djeluju na principu jednosmjerne ili povratne veze. To znači da kontrolor preuzima izmjerene vrijednosti okoliša na temelju kojih pruža određeni scenarij.
- Povratne informacije su u obliku izvršne komande za proces (e) (npr. Uključivanje grijanja ili klimatizacije).

- Ulazni uređaji sustava su različiti mjerni uređaji kao što su termostati, svjetlosni senzori, senzori vlažnosti tla (sustavi navodnjavanja), senzori pokreta i drugi.
- Izlazni uređaji sustava su ventili, kućna rasvjeta, multimedijski uređaji, klima uređaji, strojevi za pranje, pećnica, alarmni sustavi, sustavi otvaranja (garažna vrata, vrata, rolete) i drugo.

Pravi izbor kompaktnih uređaja ključni je čimbenik u smislu funkcionalnosti i stabilnosti sustava.

4. Protokol

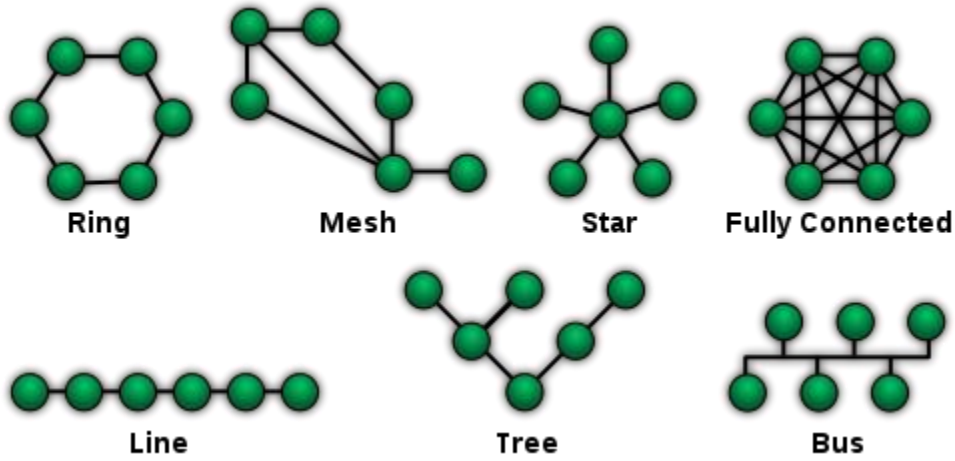
Kada govorimo o protokolu komunikacije između uređaja valja razdvojiti načine komunikacije. Glede na način povezivanja, to su (TSTBiometrics.com, n.d.):

- Žična (X10, HomePlug Power Alliance, CEBus, Insteon..)
- Bežična komunikacija (ZigBee, Z-Wave, ZeroG, EnOcean, TCP/IP..)

Kao dugogodišnjeg lidera na području upravljana i automatizacije kuća, uzimamo poduzeće X10 koje se među prvima bavilo proizvodnjom uređaja i razvijalo vlastiti protokol. Na tržištu su se etablirali kao proizvođač relativno povoljnih rješenja. „Protokol X10 koristi postojeću strujnu infrastrukturu po kojoj pulsno šalje signale od bazne stanice ili računala sa X10 sučeljem prema krajnjim aktuatorima. Najveći nedostaci X10 su degradacija signala, kontrolni zbroj (eng. Checksum), povratne potvrde signala, robusnost hardvera te primarna namjena samo za kontrolu releja“ (Riley, 2012, p. 4).

Kod bežične komunikacije također postoji više načina komunikacije, te više topologija. Topologija definira način na koji uređaji među sobom komuniciraju. „Najčešće se koristi topologija tipa zvijezda jer koristi središnju upravljačku jedinicu koja komunicira sa svim dostupnim uređajima (klijentima) i preuzima odgovornost za donošenje odluka. Uloga klijenata je slanje podataka preuzetih sa senzora natrag upravljačkoj jedinci, koja nakon analize primljenih podataka šalje klijentu naredbu za poduzimanje određenih radnji. Još jedna topologija koju treba razmotriti je mrežna topologija koja nema upravljačke jedinice

i koristi konstelaciju kontrolnih jedinica otprilike jednake inteligencije i sposobnosti međusobno povezane. Svaka jedinica šalje informacije na mrežu koju dijele sve jedinice. Svaka jedinica je neovisna i donosi vlastite odluke na temelju zajedničkih informacija. Izbor topologije sustava regulira izbor komunikacijskih sučelja kao što su ZigBee, Z-Wave, ZeroG, EnOcean, TCP/IP, RF, Bluetooth, itd.“ (Rastogi & Sharma, 2014) Dijagram najčešćih sabirničkih topologija možemo vidjeti na slici 1.



Slika 1 - Slikovni prikaz mrežnih komunikacija Izvor: (Wikipedia, n.d.)

Sve češće čujemo pojam IoT i možda je to smjer u kojem će se kućna automatizacija razvijati. Naime, većina IoT uređaja, opremljena je ethernet portom ili pak Wi-Fi karticom te se preko usmjerivača povezuju na poslužiteljsku aplikaciju za upravljanje koja može biti na lokalnom serveru no, također, može se nalaziti u nekom cloud sustavu. Pojeftinjenje mikrokontrolera i razvoj integriranih operacijskih sustava doveli su ubrzanog razvoja grane pa u skoroj budućnosti možemo očekivati veću primjenu automatizacije u kućanstvu te sustava pametnih kuća.

Općenito u automatizaciji, najčešći elementi za upravljanje automatikom jesu PLC (programmable logic controller), odnosno industrijska računala koja imaju procesor, memoriju te industrijske ulaze i izlaze koje, za razliku od kućnog računala, koriste umjesto tipkovnice, miša i ekrana. Takva računala još su uvijek relativno skupa te ih je općenito teže programirati jer za programiranje ne koriste standardne programske jezike već „ladder logiku“ koja predstavlja shematski dijagram relejne logike. Takva notacija je odabrana kako bi se smanjio zahtjev za obukom postojećih tehničara. Suvremeni PLC-i

se, osim na navedeni način, mogu programirati logikom stanja (state logic) i u programskim jezicima C ili BASIC. Prvotni PLC-i su koristili programiranje liste instrukcija baziranih na logici stoga“ (Murugan, 2008). Protokoli komunikacije u automatizaciji PLC-om, najčešće su Modbus, Profibus (Siemens), Fieldbus, EtherNet/IP, DeviceNet..

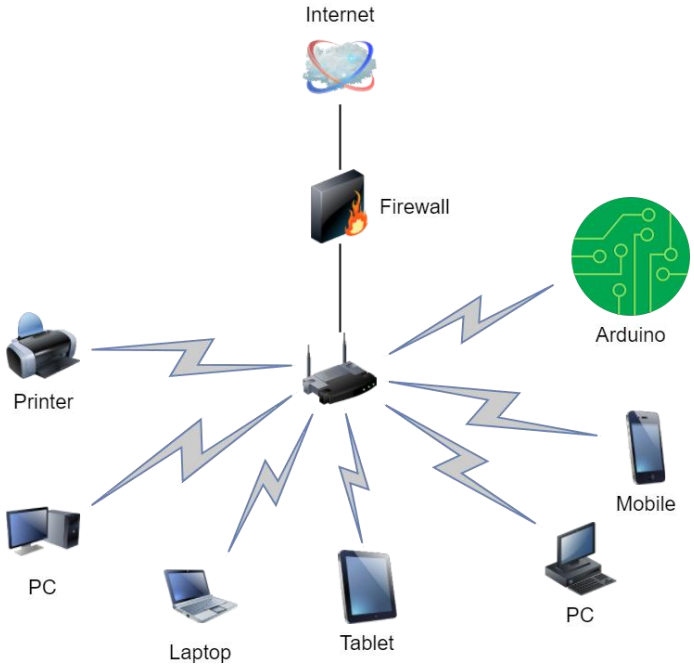
Kontroleri koji se baziraju na mikrokontrolerima poput Microchipovog PIC-a, Atmelovog AVR-a jeftini su, no zahtijevaju znanje nižih programskih jezika, strukture čipa te vanjskog sučelja za programiranje pa nisu praktični. Komunikacija između njih je najčešće serijska: UART, USART ili I^2C.

U skorije vrijeme dizajnirani su Arduino sustavi, Njihov otvoreni kod pridonosi popularnosti, pa postoje razni brendovi zadnjih nekoliko godina koji se zasnivaju na Arduino dizajnu i filozofiji, vrlo su rasprostranjeni i cjenovno vrlo prihvatljivi (oko 20€). Također, vrlo su pogodni za korištenje u sustavima kućne automatike pa ćemo se sa njima baviti u ovom radu.

5. Sustav automatizacije MojDom

Za potrebe ovog rada dizajniran je sustav kao na slici 2.

Požarni zid je na Mikrotikovom usmjerivaču i propušta portove 443 i 22 prema internom Asus (OpenWrt) usmjerivaču. Važno je napomenuti da mu je osigurana fiksna IP adresa zbog registracije SSL certifikata. Interni usmjerivač, osim što vrši svoju primarnu funkcije (usmjeravanje prometa, mrežno preklapanje, Wi-Fi dostupna točka, NAS) služi i kao poslužitelj za MojDom aplikacije. Preko USB kabla spojen je sa Arduinom koji upravlja relejima, senzorima i trijacima (izvršni elementi spojeni preko digitalnih i PWM izlaza Arduina). Svaki Arduino (za primjer ovog rada korišten samo jedan) pamti svoja stanja te preko povratne veze u realnom vremenu ista dostavlja serveru kako ne bi u isto vrijeme, sa dva ulaza dobio različite upute.



Slika 2 - Dizajn sustava (izrada autora)

6. MojDom Hardver

6.1. Odabir hardvera

Kod odabira hardvera osnovni uvjet je bio open source[1], jednostavna mogućnost instaliranja softvera te veći broj podržanog softvera. Kako je zamišljen server koji treba imati mogućnosti posluživanja PHP i HTML, softverski preduvjet je bio jednostavna instalacija OpenWrt-a. Tu je u obzir došlo nekoliko uređaja no odluka je pala na Asusov RT-N14U usmjerivač sa 600 MHz procesorom, 16 MB flash memorije te 64 MB radne memorije. Usmjerivač je kao takav spreman za spajanje na Internet što je također

[1] Open Source (hrv. Otvoreni kod) se odnosi na ono što se može dijeliti i mijenjati jer je javno dostupno (opensource.com, n.d.)

preduvjet da bi MojDom bio dostupan izvan lokalne mreže. Također, RT-N14U sadrži USB port, Wi-Fi sa 802.11b, 802.11g, 802.11n te serijski port[2] (koji, doduše, nije eksponiran na kućištu te zahtijeva lemljenje konektora). Kod odabira elemenata elektronike za upravljanje relejima u užem izboru su bili PIC16F877 mikrokontroler te Arduino Nano. Za konačan odabir Arduina zaslužna je njegova jednostavnost korištenja, programiranja a i sadrži sve potrebne elemente kako bi se ga jednostavno integriralo unutar bilo kojeg projekta (napaja se putem mikro usb kabela, nije mu potrebna periferija, vanjski sat i sl.) Za izvršne elemente odabran je relejni niz koji je naručen putem stranice dx.com iz razloga što je niz od 8 releja bio jeftiniji od pojedinog releja.

6.2. Arduino

„Arduino je vodeći svjetski ekosustav otvorenog izvornog hardvera i softvera. Tvrtka nudi niz softverskih alata, hardverskih platformi i dokumentacije koja gotovo svakome može omogućiti da bude kreativan s tehnologijom.

Arduino je popularan alat za razvoj internetskih proizvoda, kao i jedan od najuspješnijih alata za STEM / STEAM obrazovanje. Stotine tisuća dizajnera, inženjera, studenata, programera i kreatora širom svijeta koriste Arduino za inovaciju u glazbi, igrama, igračkama, pametnim kućama, poljoprivredi, autonomnim vozilima i još mnogo toga.

Prema Arduino započinje kao istraživački projekt Massima Banzija, Davida Cuartiellesa, Toma Igoe, Gianluce Martina i Davida Mellisa na Institutu Interakcijskog Dezajna Ivrea (IDII) početkom 2000-ih, kao jezik za učenje kodiranja unutar konteksta vizualne umjetnosti razvijen od strane Caseya Reasa i Bena Frya, te naposlijetku projekt teme Hernanda Barragana o ploči ožičenja.

Arduino ploča uvedena je 2005. godine kako bi pomogla studentima dizajna - koji nisu imali prethodno iskustvo u programiranju elektronike ili mikrokontrolera - stvoriti

[2] Serijski komunikacijski interfejs, koristi RS-232 priključak (Evans, et al., 1976, p. 36)

prototipove koji povezuju fizički svijet sa digitalnim. Od tada je postao najpopularniji alat za prototipove elektronike koji koriste inženjeri, pa čak i velike korporacije.

Arduino je prvi široko rasprostranjeni projekt Open Source Hardvera realiziran za izgradnju zajednice koja bi mogla pomoći u širenju korištenja alata od kojih će imati koristi i koja je stvarana od doprinosa stotina ljudi koji su pomogli u ispravljanju koda, pisanju primjera, stvaranju vodiča, podršci ostalim korisnicima.

Od osnutka Arduino projekta, uvedene su mnoge nove razvojne zajednice i softverske knjižnice, proširujući raspon mogućnosti dostupnih zajednici. Danas, više od desetljeća kasnije, Arduino nastavlja pružati hardver i softver otvorenog koda kako bi nove ideje oživjeli.

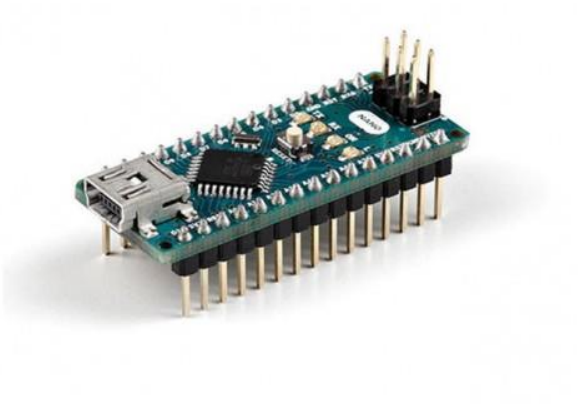
Otvorenost i jednostavnost korištenja projekta doveli su do masovnog usvajanja mikroprocesorske elektronike temeljene na projektima, pa je Arduino bio katalizator u stvaranju Makers pokreta[3]. Ujedno je postao broj jedan izbor proizvođačima elektronike, posebno za razvoj rješenja za IoT, za koji se predviđa da će do 2021. godine postati tržište od 6 trilijuna dolara.“ (Arduino, n.d.)

Arduino kao razvojna okruženja razlikuju se po mogućnostima, memoriji, broju ulaza i izlaza i sl. Najčešće su korišteni: Arduino Uno, Arduino Leonardo, Arduino Micro, Arduino Nano, Arduino Mega, Arduino Zero, Arduino Due, Arduino Due, Arduino Ethernet te Arduino Diecimila i Arduino Duemilanove koji se više ne proizvode. Za potrebe ovog rada, koristiti će se Arduino Nano, relativno jeftini kontroler sa pregršt ulazno – izlaznih jedinica te USB portom.

6.3. Arduino Nano

Za tehničku implementaciju rješenja MojDom koristiti će se Arduino Nano (slika 2), ulazni model, dostupan po cijeni 20€ na web trgovini Arduina (https://store.arduino.cc)

[3] Makers pokret - trend u kojem pojedinci ili grupe pojedinaca stvaraju i plasiraju proizvode koji se rekreiraju i montiraju koristeći neiskorištene, odbačene ili slomljene elektroničke, plastične, silicijske ili gotovo bilo koje sirovine i / ili proizvoda s računala povezanog uređaja. (Technopedia.com, n.d.)

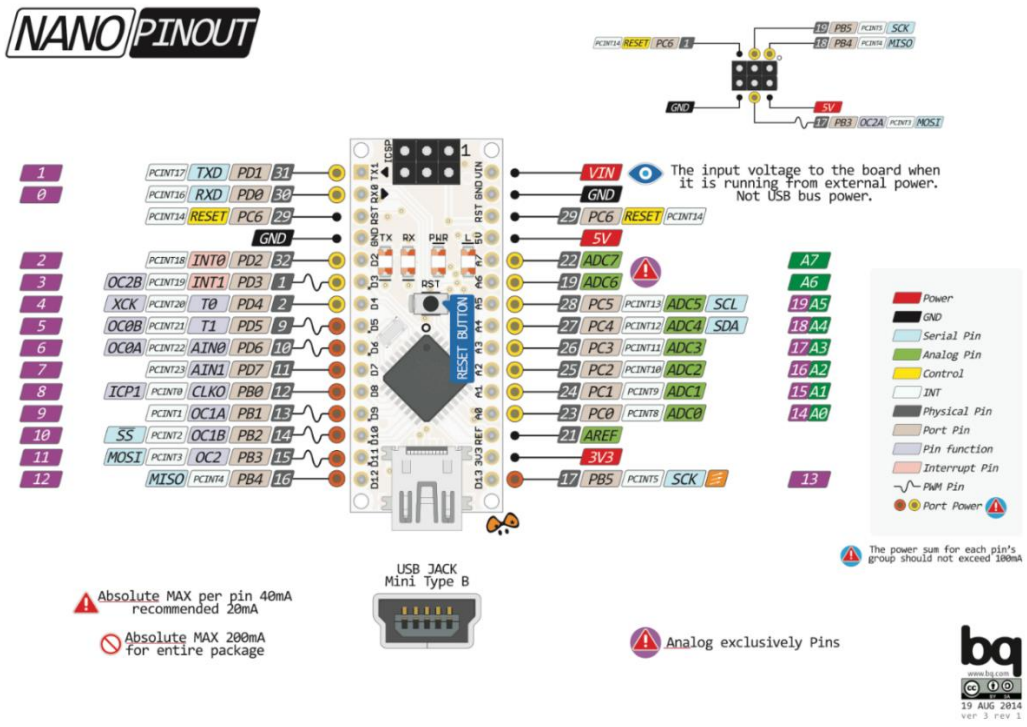


Slika 3- Arduino Nano (ArduinoCC, n.d.)

Arduino Nano razvojno okruženje bazirano je na ATmega328 mikrokontroleru. Samo okruženje nema DC priključak pa se može napajati ili preko pina (5V) ili preko microusb konektora koji ujedno služi kao port za programiranje (preko ugrađenog bootloadera – sustava koji učitava i pokreće zadatke i procese prilikom startanja određenog sustava) te kao komunikacijski port (serijska RS232 komunikacija, ugrađen MAX232 čip ili slično). Također, putem pina 0 i 1 dostupna je UART TTL komunikacija. Sam čip podržava i I^2C te SPI protokole komunikacije. Kapacitet memorije navedenog mikrokontrolera je 32KB flash, 1KB EEPROM te 2KB radne memorije pri brzini od 16Mhz. U samo 18x45mm pločica sadrži 6 PWM te 22 digitalna I/O pina.

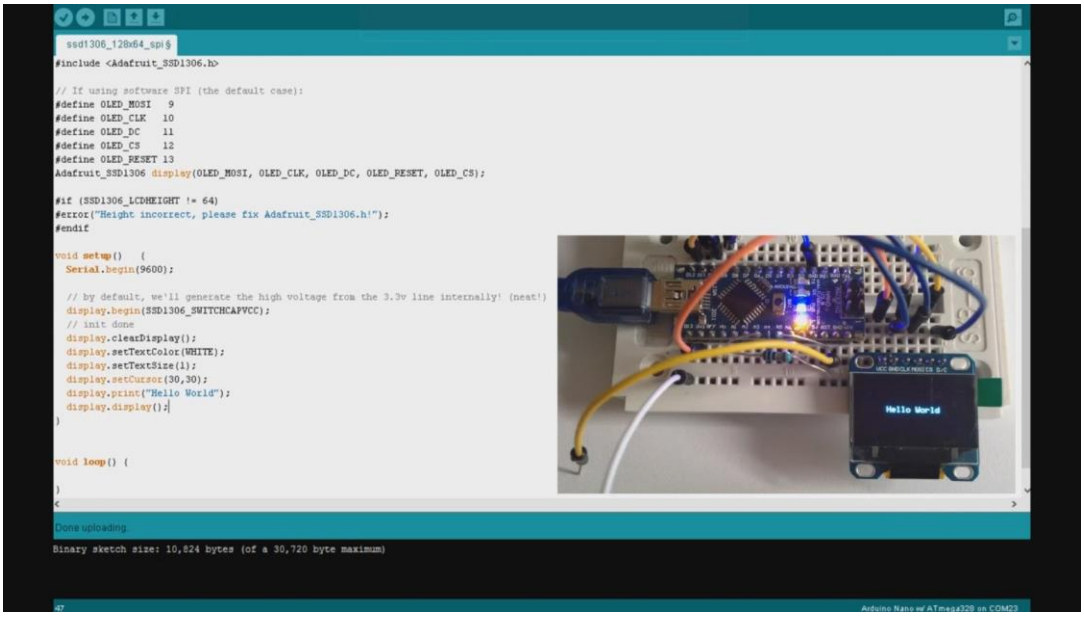
Za razliku od navedenih PLC-ova, Arduino se programira u klasičnom razvojnom okruženju, u C/C++ jeziku. Bootloader omogućava jednostavno kopiranje izvršne datoteke u razvojno okruženje preko microUSB kabla.

Na slici 4. može se vidjeti shematski prikaz sustava Arduino Nano sa numeracijom i definicijama pinova.



Slika 4- Raspored pinova na Arduino Nano (www.bq.com, 2014)

Na slici 5. prikazan je izgled razvojnog okruženja sa primjerom programa „Hello World“, napisanog u programskom jeziku C++, te Arduino Nano sa pokrenutim programom.



Slika 5 - Arduino razvojno okruženje sa kôdom za "Hello World" (Planetleak, 2014)

6.4. OpenWrt

Usmjerivače, odnosno preklopnike često se poima samo kao uređaje koji služe za spajanje računala/mobilnih uređaja na Internet te spajanje telefona na VoIP ulaze. Napredniji korisnici iste znaju koristiti za NAS pohranu ili mrežno tiskanje. „Usmjerivači se ponašaju poput poštanskog sandučića koji korisnicima unutar mreže dostavlja podatke sa vanjskih mreža. Jedna od većih razlika je što je kod većine korisnika IP adresa promjenjiva dok je kućna ili poslovna adresa korisnika uglavnom nepromjenjiva. „ (Cisco, n.d.)

Sa strane hardvera ruteri su računala koja sadrže potrebne mrežne portove, komunikacijske portove za servis a mogu sadržavati i VoIP portove, USB portove i sl.

Sa strane softvera ruteri uglavnom pokreću vlasnički softver proizvođača koji je prilagođen svakom modelu. Često dva različita modela (skuplji i jeftiniji) imaju jednaki hardver a razlikuju se samo u OS kojeg pokreću. Kako je ruter u osnovi računalo, postoji nekoliko proizvođača softvera te nekoliko Open Source inicijativa koji proizvode OS za rutere. Najpoznatiji od njih je OpenWrt, Linux distribucija namijenjena za ugrađene uređaje (eng. Embedded device). OpenWrt je zamišljen kao razvojno okruženje za programiranje aplikacija za rutere, bez potrebe za pisanjem firmvera. Kao Linux distribucija, nativno podržava bash skripte i LUA programski jezik te skidanje paketa sa raznim skriptama i programima.

7. Podešavanje hardvera

7.1. Usmjerivač i instalacija OpenWrt-a

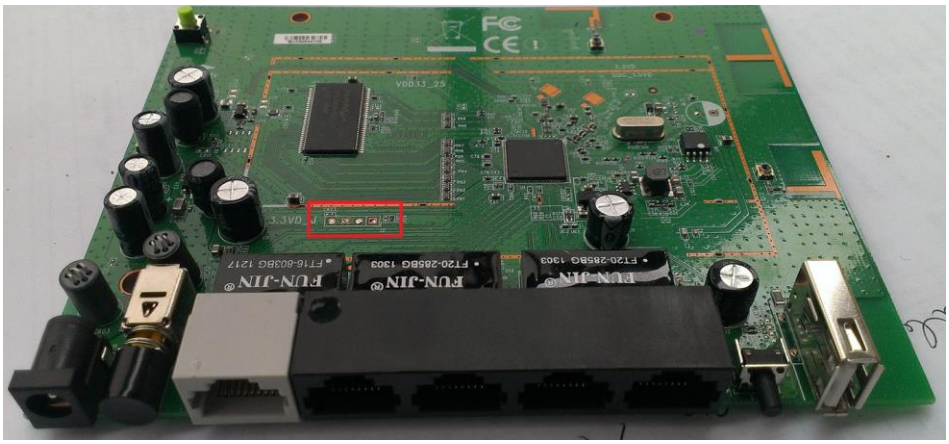
Kao interni usmjerivač te MojDom server, odabran je Asus RT-N14U sa taktom 600MHz, 16Mb flash memorije te 64Mb radne memorije, jednim WAN portom (za vezu prema dolaznom usmjerivaču), 4 LAN porta za interne konekcija te USB-om. Prilikom odabira OpenWrt-a kao OS bitno je potražiti podržane uređaje kako bi se izbjeglo pisanja drivera i prilagođavanja usmjerivaču. To je bio još jedan kriterij pri odabiru ovog usmjerivača. Instalacija OpenWrt-a je vrlo jednostavna: u administratorskom sučelju usmjerivača odabere se zapisivanje novog verzije firmvera te se učita prethodno skinuta *.bin datoteka sa stranica https://wiki.openwrt.org/toh/asus/rt-n14u. Nakon učitavanja usmjerivač se ponovno pokrene i sustav je instaliran. Komunikaciju sa konzolom OpenWrt-a možemo ostvariti na 2 načina:

- Preko serijskog porta
- Preko SSH protokola (putty ili sl.)

7.2. Komunikacija preko serijskog porta

Komunikacija preko serijskog porta (Evans, et al., 1976, p. 36) zanimljiva je iz razloga što su vidljivi svi signali od samog pokretanja usmjerivača pa je odlična za otkrivanje grešaka u fazi razvoja. Međutim, može se primijetiti kako usmjerivač nema serijskog porta na koji

bi se spojili. Zapravo, da bi se uspostavilo serijsku komunikaciju, mora rastaviti usmjerivač te zalemiti konektor (eng. pin header) za serijsku komunikaciju.



Slika 6 - Tiskana pločica usmjerivača ASUS RT-N14U, crveno je označene mjesto za pin header (Izrada autora)

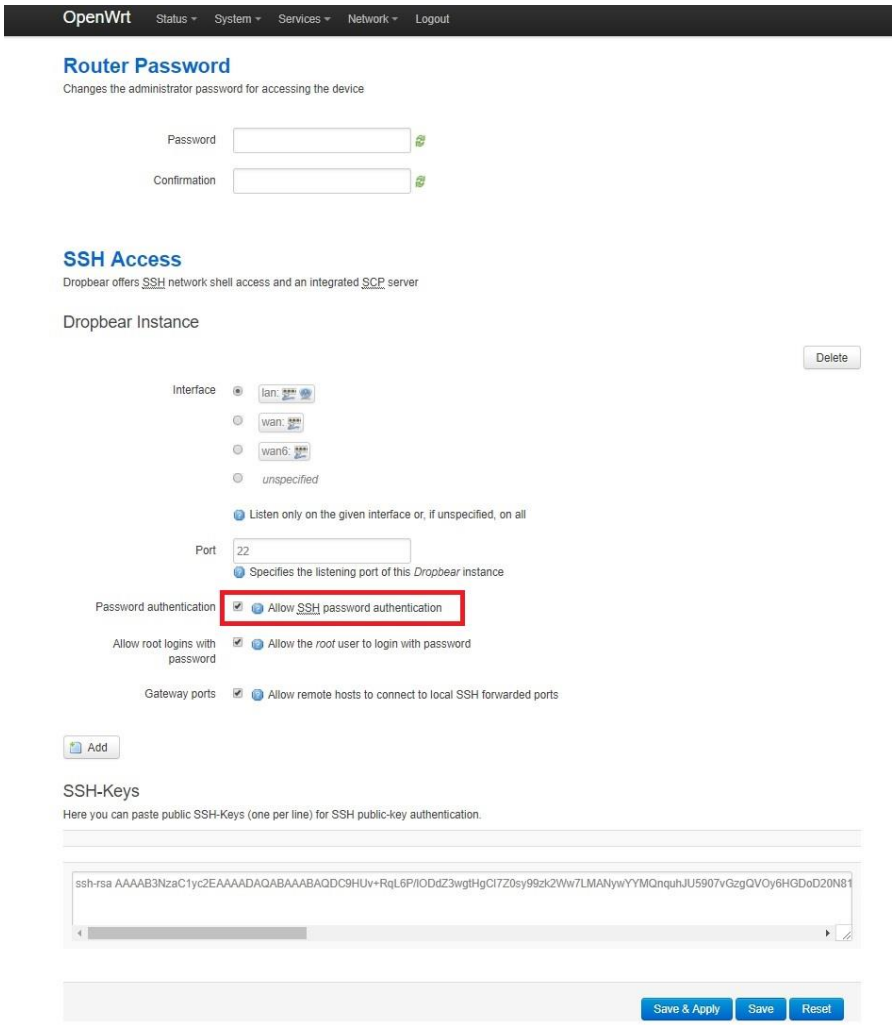
Na slici 5, unutar crvenog pravokutnika vide se 4 točkice na koje treba zalemiti konektor. Redom, one su: TX, GND, RX, Vcc (~3,3V). Sa 3,3V, signal je nižeg napona od napona porta na računalu pa će se za spajanje, umjesto sa serijskog porta računala ili pak USB-SERIAL kabela koristiti USB-TTL pretvornik.

Nakon instalacije drivera na usmjerivač, potrebno je podesiti port kako bi računalo i OpenWrt komunicirali po istim parametrima. Inicijalna brzina (baudrate) je 57600. Nakon spajanja i učitavanja sustava, spremni smo za komunikaciju sa sustavom.

7.3. Komunikacija preko SSH protokola

Prednost komunikacije preko SSH je mogućnost udaljenog upravljanja sa usmjerivačem. Za razliku od serijske komunikacije ovdje su potrebni korisničko ime i lozinka. Također, SSH konzoli ne može se pristupiti dok se ne učita SCP server. Za početak, treba postaviti fiksnu IPv4 ili IPv6 adresu kako se uvijek moglo spajati na na isti IP.

Inicijalno, OpenWrt ne omogućava spajanje preko SSH pa je to potrebno omogućiti kroz serijsku konzolu ili web sučelje. Potrebno je definirati korisničko ime i lozinku te uključiti opciju SSH.



Slika 7 - Omogućavanje SSH pristupa (Izrada autora)

Za spajanje na usmjerivač preko konzole može se koristiti program Putty (https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html). Nakon pokretanja programa upiše se IP adresa usmjerivača, označi SSH protokol te, ukoliko nije mijenjan zadani port, upiše port 22. Nakon otvaranja konekcije potrebno se prijaviti gore postavljenim imenom i lozinkom.

8. Softver

8.1. Odabir softvera

Prilikom odabira softvera vodilo se računa o više pretpostavki: za server je odabran OpenWrt[4], kao okruženje za programiranje Arduina odabrano je standardno Arduino sučelje i programski jezik C a za sučelje PHP, HTML, JQuery te React Native.

Razlozi odabira navedenih alate različiti su. OpenWrt je odlična alternativa statičnom firmveru pa je odlično rješenje za samogradnju, odnosno, za prenosivost sa uređaja na uređaj. OpenWrt se na ruter instalira najčešće kroz postojeće sučelje kao nadogradnja a besplatno se može prenijeti sa mrežnih stranica.

Najčešće korišteni programski jezici za programiranje Arduina su C,C++ i Python no kako je sam program za upravljanje relejima preuzet sa Interneta a napisan je u C++, u ovom radu, neće biti osvrta na ostale jezike.

Kod odabira PHP-a kao skriptnog jezika za interakciju između servera i klijenta te servera i Arduina presudila je jednostavnost i brzina te mogućnost pokretanja nativnih naredbi programske ljuske OpenWrt-a. PHP je programski jezik koji se izvodi na serverskoj strani, a služi za kreiranje interaktivnih mrežnih stranica. Prema (php.net, n.d.) PHP je kratica od imena „Hypertext Preprocessor“. Vrlo je popularan zbog toga jer se može direktno upisivati kraj HTML koda, besplatan je i otvorenog je koda te ima podršku za rad s raznim bazama podataka kao što su MySQL te je ujedno u ta kombinacija najrasprostranjeniji način izrade interaktivnih mrežnih stranica. Struktura i sintaksa samog koda je vrlo slična C programskom jeziku ili JavaScript.

HTML, CSS i JQuery su korišteni za web sučelje za upravljanje zbog jednostavnosti i rasprostranjenosti

[4] OpenWrt Project je linux sustav za ugrađene uređaje (Brown, 2016)

8.2. Instalacija esencijalnog softvera na OpenWrt

Na OpenWrt-u je softver moguće instalirati na više načina poput instalaciju kroz web sučelje ili preko terminala. Za instalaciju preko terminala koristimo predinstalirani upravljač paketa OPKG. Komanda za instalaciju je: **opkg -i *nazivpaketa***

Za lakšu instalaciju i izbjegavanje problema sa različitim verzijama, priložen je izbor paketa autora:

base-files, bash, busybox, ca-certificates, coreutils, coreutils-stty, curl, ddns-scripts, ddns-scripts_no-ip_com, diffutils, dnsmasq, dropbear, firewall, fstools, hostapd-common, ip6tables, iptables, iw, jshn, jsonfilter, kernel, kmod-cfg80211, kmod-crypto-aes, kmod-crypto-arc4, kmod-crypto-core, kmod-eeprom-93cx6, kmod-gpio-button-hotplug, kmod-ip6tables, kmod-ipt-conntrack, kmod-ipt-core, kmod-ipt-nat, kmod-ipv6, kmod-leds-gpio, kmod-ledtrig-usbdev, kmod-lib-crc-ccitt, kmod-lib-crc-itu-t, kmod-mac80211, kmod-nf-conntrack, kmod-nf-conntrack6, kmod-nf-ipt, kmod-nf-ipt6, kmod-nf-nat, kmod-nf-nathelper, kmod-nls-base, kmod-ppp, kmod-pppoe, kmod-pppox, kmod-rt2800-lib, kmod-rt2800-mmio, kmod-rt2800-pci, kmod-rt2800-soc, kmod-rt2x00-lib, kmod-rt2x00-mmio, kmod-rt2x00-pci, kmod-slhc, kmod-usb-acm, kmod-usb-core, kmod-usb-ohci, kmod-usb-serial, kmod-usb-serial-ftdi, kmod-usb2, libblobmsg-json, libc, libcurl, libgcc, libip4tc, libip6tc, libiwinfo, libiwinfo-lua, libjson-c, libjson-script, liblua, libncurses, libnl-tiny, libopenssl, libpcre, libpolarssl, libpthread, librt, libubox, libubus, libubus-lua, libuci, libuci-lua, libusb-1.0, libustream-polarssl, libxml2, libxtables, lua, lua-examples, luasocket, luci, luci-app-commands, luci-app-ddns, luci-app-firewall, luci-app-ntpc, luci-base, luci-lib-ip,

Neki od osnovnih paketa su ugrađeni u sam OpenWrt a neki od navedenih treba instalirati putem navedene komande. Neki od programa koji će zasigurno trebati za pokretanje servera za kućno automatiku su uhttpd (web server), php5* (php interpreter), luci-ssl, openssl-util, libopenssl (alati za certifikate, kako bi stranica bila potpisana), kmod-usb-serial, kmod-usb-serial-ftdi (driveri za USB-TTL kabel), ntpclient (automatsko sinkroniziranje vremena).

Serverski program za upravljanje kućnom automatikom treba kopirati u datoteku koju uhttpd eksponira kao server document root.

Za uspostavljanje uspješne komunikacije između rutera i Arduina, potrebno je podesiti baud na slijedeći način:

stty -F /dev/ttyS0 57600 clocal cread cs8 -cstopb –parenb

Naravno, da bi navedena komanda radila, trebaju biti instalirani pakete coreutils i coreutils-stty. Ukoliko se ukaže potreba za drugim brzinom prijenosa podataka ili je serijska komunikacija na Arduinu podešena na drugi takt, jednostavno se zamijeni 57600 sa željenim taktom (npr. 9600). Kako bi prilikom svakog ponovnog pokretanja usmjerivača postavke bile učitanje, navedenu komandu treba unijeti u „cron jobs“:

mkdir /etc/crontabs
#možemo primijetiti kako je /var/spool/cron/crontabs uobičajena mapa za crontabs
#no kako se sve u direktoriju /var/ briše prilikom ponovnog pokretanja mi smo napravili /etc/crontabs
touch /etc/crontabs/root
ln -sf /etc/crontabs/root /etc/crontab
touch /etc/init.d/SerijskiCron
#ovu skriptu generiramo te kopiramo slijedeće linije
#!/bin/sh
start crond
/usr/sbin/crond -c /etc/crontabs
stty -F /dev/ttyS0 57600 clocal cread cs8 -cstopb –parenb
#Navedenom komandom SerijskiCron postaje izvršna datoteka
chmod 755 /etc/init.d/SerijskiCron

#isključi se cron i ponovno pokrene za test:

killall crond;

/etc/init.d/SerijskiCron

8.3. Arduino softver

Arduino softver se bazira na KTA-223v12 (OceanControls.com, n.d.) koji ima licencu Creative Commons Attribution Share-Alike (cc-by-sa) licencom. KTA-223 je relejni kontroler koji se bazira na Arduino Duemilenove. Zbog veličine i cijene, za ovaj projekt je odabran Arduino nano. Izvorno je KTA-233 zamišljen kao USB/RS-485, za primjenu na Arduino Deumilanove ili Uno kontroleru, gdje se, ukoliko se koristi više kontrolera, prvoga spaja na USB a ostalima se prosljeđuju poruke preko RS-485. Kako u konkretnom primjeru ne postoji potreba za dislociranim kontrolerima svi Arduino nano se spajaju na USB hub koji je spojen u USB port usmjerivača. Arduino nano ima jedan USB port koji je u osnovi USB - serial konverter a OpenWrt ih prepoznaje respektivno kako /dev/ttyUSB0, /dev/ttyUSB1…

Nakon programiranja Arduina te spajanja istog u USB port usmjerivača potrebno je uspostaviti testnu komunikaciju. Najlakše je spojiti se preko puttyja na serijsku konzolu ili SSH te pokrenuti programe poput picocom ili screen (instalacija pomoću OPKG na gore navedeni način) te poslati upit na Arduino: @00 RS 0 (status svih releja, svih Arduina). Ukoliko je odgovor #00 xx, gdje je xx decimalno stanje releja (binarna reprezentacija 8 releja za prvi i zadnji uključeni relej je 10000001, dakle (2^(1-1) + 2^(8-1) = 129). Za prvi i zadnji uključeni relej, odgovor bi bio #00 129

Iz dokumentacije dostupne na mrežnim stranicama (OceanControls.com, n.d.), možemo pronaći sve moguće komande:

Slova	Komanda	Parametri
ON	Uključi relej	1-8: Uključi releje 1-8 individualno 0: Uključi sve releje odjednom
OF	Isključi relej	1-8: Isključi releje 1-8 individualno 0: Isključi sve releje odjednom
WR	Pošalji komandu svim relejima	Parametar je broj koji pokazuje koji releji će biti uključeni a koji isključeni

<table>
<tr><td>TR</td><td>Uključi releje na određeno vrijeme</td><td>2 parametra, Prvi 1-8: broj releja, Drugi 001-255: Vrijeme u sekundama koliko će relej biti uključen.</td></tr>
<tr><td>KA</td><td>Zadrži uključeno</td><td>Kao i TR samo što se koristi ako je bitno da je relej pod nadzorom. Ukoliko je potrebno da nešto bude duže uključeno, šalje se signal svakih Tmax – 10s, brine da ako nestane komunikacije nakon određenog vremena, uređaj bude isključen</td></tr>
<tr><td>RS</td><td>Status releja</td><td>1-8: Prikazuje status releja 1-8 individualno
0: Prikazuje status svih releja odjednom</td></tr>
<tr><td>IS</td><td>Status ulaza</td><td>1-4: Prikazuje status ulaza 1-4 individualno
0: Prikazuje status svih ulaza</td></tr>
<tr><td>AI</td><td>Čitaj digitalne ulaze</td><td>1-3: Prikazuje status analognih ulaza 1-3 individualno
0: Prikazuje status svih analognih ulaza</td></tr>
<tr><td>SS</td><td>Status sustava</td><td>Mora biti 0, Vraća status svih releja, analognih i digitalnih ulaza</td></tr>
<tr><td>SA</td><td>Postavi adresu</td><td>01-99: Sprema adresu uređaja u memoriju</td></tr>
<tr><td>SB</td><td>Postavi baud rate</td><td>1-10: Postavlja Baud Rate
<table>
<tr><td>1: 1200 baud</td><td>6: 19200 baud</td></tr>
<tr><td>2: 2400 baud</td><td>7: 28800 baud</td></tr>
<tr><td>3: 4800 baud</td><td>8: 38400 baud</td></tr>
<tr><td>4: 9600 baud (uobičajeno)</td><td>9: 57600 baud</td></tr>
<tr><td>5: 14400 baud</td><td>10: 115200 baud</td></tr>
</table></td></tr>
</table>

Tablica 1 - Komande za komunikaciju sa Arduinom (OceanControls.com, n.d.)

8.4. MojDom softver

Koncept upravljanja zasniva se na master-slave komunikaciji, bez baze podataka gdje svaki Arduino element pamti svoja stanja. Master je zapravo middleware u obliku kojeg obavlja PHP skripta. Neka drugi program preko te skripte čita i postavlja vrijednosti na Arduinu. Za ovaj primjer napravljena su dva sučelja: web sučelje i dinamičko mobilno sučelje. Osnovne skripte za komunikaciju

Komunikacija između OpenWrt-a i Arduina izvršava se vanjskim pozivom na writeStatus.php i readStatus.php koji su svojevrsni middleware u ovom upravljačkom sustavu.

Na slici 8 vidimo kôd writeStatus.php skripte koja služi za slanje statusa svakog pojedinog pina klijentu (Arduino). Kako je ova verzija softvera programiranja isključivo za testiranje mogućnosti skripta komunicira samo sa jednim uređajem (devTTY0). To se može promijeniti na način da se varijabla $port prosljeđuje putem $_GET naredbe.

Kako bi se skripta writeStatus.php mogla koristiti u svrhe otklanjanja grešaka, omogućen je pristup preko $_POST i $_GET. U produkcijskoj varijanti trebalo bi izbrisati dio sa $_POST kako ne bi bilo omogućen direktan pristup iz preglednika. Da bi skripta uredno funkcionirala, aplikacija za web sučelje ili mobilna aplikacija šalje pin (u izvršnoj okolini to je relej) i status releja (odnosno uređaja kojim upravljamo). Nakon što primitka varijable, generiramo string za slanje na port: $signal = "@00 " . $status . " " . $pin . "\r";

```
<?php
// Dozvoli pristup skripti izvan localhosta
header('Access-Control-Allow-Origin: *');
// Postavi timeout na 500 ms
$timeout = microtime(true) + 0.5;
$tagid = "";
// Postavi opcije za komunikaciju sa uređajem
exec("/bin/stty -F /dev/ttyS0 9600 sane raw cs8 hupcl cread clocal -echo -onlcr ");
while ($tagid[0] != "#") {
   $f = file_put_contents('/dev/ttyUSB0', "@00 RS 0\r");
   $timeout = microtime(true) + 0.5;
   $handle = fopen('/dev/ttyUSB0', 'r'); // Otvori uređaj za čitanje
   $tagid = fgets($handle); // Čitaj podatke iz uređaja
}
print $tagid; // Prikaži podatke
fclose($handle); // zatvori datoteku uređaja
?>
```

Slika 8 - writeStatus.php, skripta za čitanje iz klijenta i slanje u displej (Izrada autora)

Kao i kod writeStatus.php, tako i kod readStatus.php (slika 8) komuniciramo samo sa jednim portom što možemo promijeniti na način:

$port = /dev/ttyUSB;

$nport = shell_exec („cd /dev | ls -1 ttyUSB | wc -l“);

for($i=0, $i<$nport, i++){

$port += $i;

}

, pa koristiti $port umjesto /dev/ttyS0. U gornjem primjeru shel_exec je naredba za izvršavanje sistemskih naredbi, a kombinacija naredbi unutar navodnika zapravo daje broj datoteka koje sadrže ttyUSB. OpenWrt za svaki novi uređaj eventualno spojen preko USB huba kreira novi ttyUSB port inkrementalno mu dodjeljujući brojeve.

```
<?php
// Dozvoli pristup skripti izvan localhosta
header('Access-Control-Allow-Origin: *');
$port ='/dev/ttyUSB0';
function openSerial($command)
{
   $openSerialOK = false;
   try {
      $fp = fopen($port, 'r+');
      $openSerialOK = true;
   }
   catch(Exception $e) {
      echo 'Message: ' . $e->getMessage();
   }
   if ($openSerialOK) {
      fwrite($fp, $command);
      fclose($fp);
   }
}
if (isset($_GET['pin'])) {
   $pin = $_GET['pin'];
   $pin++;
   $status = $_GET['status'];
   if ($status == "false") {
      $status = "of";
   }
   else $status = "on";
   $signal = "@00 " . $status . " " . $pin . "\r";
   openSerial($signal);
   print $signal;
}
if (isset($_POST['pin'])) {
   $pin = $_POST['pin'];
   $pin++;
   $status = $_POST['status'];
   if ($status == "false") {
      $status = "of";
   }
   else {
      $status = "on";
   }
   $signal = "@00 " . $status . " " . $pin . "\r";
   echo $signal;
   openSerial("$signal");
}
?>
```

Slika 9 - readStatus.php, učitavanje statusa (Izrada autora)

8.5. Web sučelje

Web sučelje je napisano u HTML5, JQuery, css te je jedan od primjere implementacije izvršne kontrole. Koristeći AJAX i JQuery pristupa se readStatus i writeStatus skriptama te se izvršavaju zadaće.

„HTML5 je jezik za organiziranje i prikazivanje sadržaja na WWW-u nastao kao rezultat rada organizacije W3C, a sa željom da se poboljša i olakša organizacija i prikazivanje sadržaja prisutnih na WWW-u danas i u budućnosti te da se omogući pristup istima na jednak način s različitih preglednika, odnosno uređaja. Nastao je kao nasljednik HTML 4.01 i XHTML 1.1 standarda“ (Webtech, n.d.)

„jQuery je brza, mala ali funkcijama bogata JavaScript biblioteka. To čini stvari poput manipulacije dokumentima, rukovanja događajima, animacije i Ajaxa mnogo jednostavnijim pomoću jednostavnog API-ja koji funkcionira u mnoštvu preglednika. S kombinacijom svestranosti i proširivosti, jQuery je promijenio način na koji milijuni ljudi pišu JavaScript.“ (Foundation, 2018)

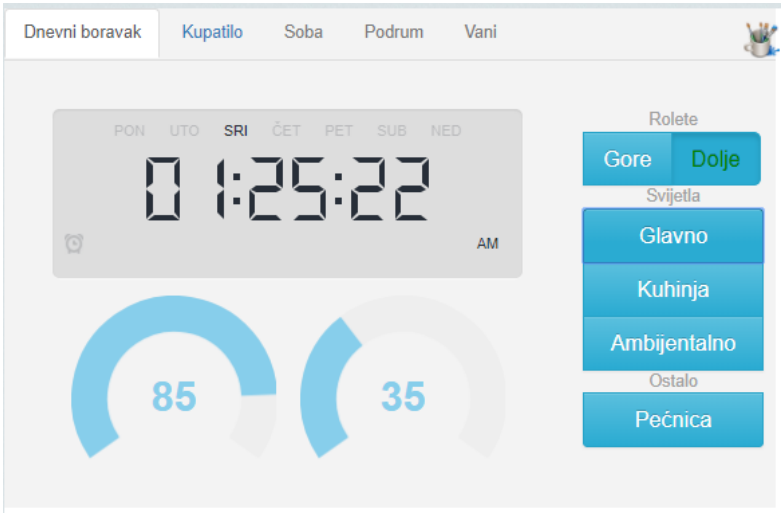
„Dok se HTML koristi za definiranje strukture i semantike vašeg sadržaja, CSS se upotrebljava za stilizaciju i postavljanje. Na primjer, možete upotrijebiti CSS da biste izmijenili font, boju, veličinu i razmak sadržaja, podijelili ga u više stupaca ili dodali animacije i druge ukrasne značajke.“ (Patonnier, 2014)

Za oblikovanje forme koristimo Bootstrap (CSS i jQuery knjižnicu), za podešavanje intenziteta svijetlosti jQuery.knob.js (Terrien, 2015)

Za prikaz sata koristimo knjižnicu moment.js (Wood, 2015)

Klikom na neki od gumbova pozivamo skriptu writeStatus.php i šaljemo naredbu za uključivanje, odnosno isključivanje, ovisno o prethodnom stanju. Kada je riječ o intenzitetu svjetla i boja valja spomenuti kako još nisu jer fizički dio sustava koji uključuje PWM dimmere nije dovršen.

Na slici 10. prikazan je izgled web sučelja, čija je funkcija omogućavanje upravljanja preko Internet preglednika. Prednost takvog upravljanja, u odnosu na upravljanje putem standardnih prekidača i tipkala je u tome što se putem aplikacije, koja se može pokrenuti na tabletu, računalu, televizoru ili mobitelu, može upravljati sa svim trošilima u kući sa jedne pozicije.



Slika 8 - Izgled web sučelja napisan u HTML5, JQuery, Bootstrap, CSS (Izrada autora)

Klikom na bilo koji od gumbova desno, šalje se signal za paljenje/gašenje pojedinog trošila. Kod roleta, za razliku od ostalih uređaja ima nekoliko koraka koji se poduzimaju. Uobičajeno stanje roleta je da se lijevi i desni gumb (gore/dolje) nalaze u neutralnom stanju. Klikom na, npr. „Dolje“, pošalje se signal za spuštanje roleta te tekst gumba pozeleni (kao na slici 10), tada se rolete spuštaju do dolje ukoliko ne prekinemo spuštanje na jedan od dva načina:

1. Ponovnim klikom na „Dolje“ šalje se signal za isključenje releja koji spušta rolete, te se rolete zaustavljaju a tekst gumba pobijeli
2. Klikom na „Gore“ prvo se šalje se signal za isključenje releja koji spušta rolete, tekst gumba „Dolje“ pobijeli. Zatim se šalje signal za isključenje releja koji spušta rolete a tekst gumba „Gore“ pozeleni

Na istoj slici vide se okrugle komande koje služe za podešavane intenziteta svijetla, ukoliko kliknemo bilo gdje na njihove rubove, ili koristimo opciju povuci i spusti (eng. drag and drop) šalje se signal konačne vrijednosti te Arduino preko PWM portova podešava intenzitet svjetlosti. Klikom na sredinu tih komandi, otvara se dijalog za odabir boje a nakon odabira, traka intenziteta poprimi tu boju te se Arduinu šalje signal za promjenu boje. Klikom na izbornik moguće je odabrati drugu prostoriju te kontrolirati trošila u istoj. Osim na događaj klika („on click“), komponente reagiraju i na događaj dodira („touchend) kako bi aplikacija radila na zaslonima osjetljivima na dodir (računalima, raznim mobilnim uređajima..)

8.6. Aplikacija za pametne telefone

Aplikacija za pametne telefone napisana je kao primjer korištenja React Native (dalje: RN) frameworka, novog JavaScript frameworka dizajniranog od strane Facebooka (zapravo se u nekim elementima razlikuje od JS) a koji ima mogućnost generiranja izvršnih datoteka za oba sustava: android i ios. Za razliku od nekih konkurentnih frameworka, poput Cordove ili Ionica, „sa React Native, ne gradi se 'mobilna web aplikacija', 'HTML5 aplikaciju', ili 'hibridnu aplikaciju'. Stvara se prava mobilna aplikacija koja se ne razlikuje od aplikacije napisane u Objective-C ili Javi. Aplikacija u React Native ima iste fundamentalne UI[5] blokove poput iOS i Android aplikacija.“ (Facebook, 2017).

Zbog međuzavisnosti pojedinih programa sa drugačijim verzijama može se naići na problem prilikom instalacije React Native, naime, nekoliko međuovisnih dijelova sudjeluje u izgradnji programa, pa se često jedan poziva na određenu verziju drugoga što se znadovesti do poteškoća. U praksi, ako instaliramo Create React Native App:

```
$ npm i -g create-react-native-app
$ create-react-native-app MojDom
$ cd MojDom
$ npm start
```

[5] Korisničko sučelje (eng. User Interface)

, čime omogućavamo pokretanje demo aplikacije, sve će funkcionirati ispravno. Ako pogledamo zapis povijesti preuzimanja, vidimo da se pokreće React, ReactNative i još mnogi drugi. Kada bi preuzeli noviju verziju npr. ReactNative:

```
$ npm install -g react-native-git-upgrade
$ react-native-git-upgrade
```

postoji mogućnost ovisno o prije preuzetom repozitoriju da program neće funkcionirati zbog međuzavisnosti o točno definiranim verzijama (najčešće u odnosu na NPM, React i sl.).

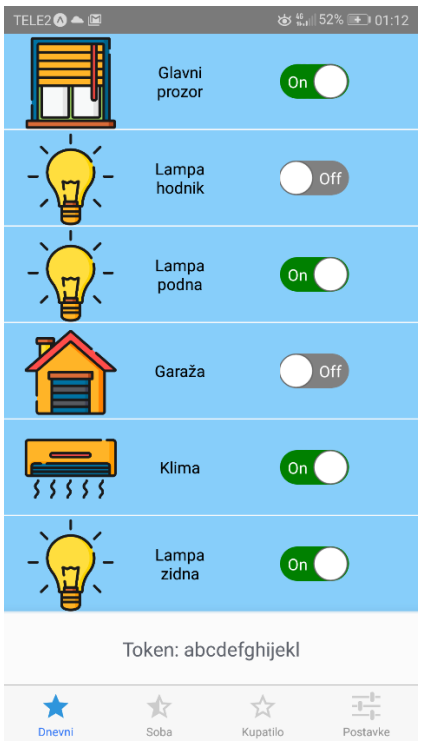
Aplikacija čiji je dio prikazan na slici 11 preuzima JSON datoteku sa servera te na osnovi nje dinamički kreira gumbove i druge elemente za kontrolu.

```
return (
  <View style={styles.containerDnevni}>
   <ScrollView style={styles.container} contentContainerStyle={styles.contentContainer}>
        {stanja.map((c, index) =>(

          <View key={"ls"+index} style={styles.containerDnevniLista}>
          <View  key={"lss"+index} style={{width: '35%', height: '100%', alignItems: 'center', justifyContent: 'center', backgroundColor: 'lightskyblue'}}>

            <Image  key={"img"+index}
              source= {(() => {
                        switch (c.vrsta) {
                          case "rolete":   return (require('../assets/images/rolete.png')) ;
                          case "lampa":   return (require('../assets/images/lampa.png')) ;
                          case "klima":   return (require('../assets/images/klima.png')) ;
                          case "garaza":   return (require('../assets/images/garaza.png')) ;
                          default:      console.log(c.vrsta);
                        }
                      })()}
              style={styles.welcomeImage}
            />
          </View>

          <View  key={"lsl"+index} style={{width: '15%', height: '100%', alignItems: 'center', justifyContent: 'center', backgroundColor: 'lightskyblue'}}>
            <Text style={{textAlign: 'center'}} key={"tro"+index}>{c.trosilo}</Text>
          </View>

          {this.renderIf(c.stanja == 2,
            <View style={[styles.container,{width: '20%', height: '100%', justifyContent: 'center', backgroundColor: 'lightskyblue', alignItems: 'center'}]}>
              <View style={{width: '40%', justifyContent: 'center'}}>
              <Slider
                trackStyle={customStyles2.track}
                thumbStyle={customStyles2.thumb}
                minimumTrackTintColor={'green'}
                maximumTrackTintColor={'red'}
                maximumValue = {2}
                step = {1}
                value={this.state.vrijednost}
                onSlidingStart={(beforeState) => {this.setState({ beforeState });
                      console.log(c.trosilo.replace("\n"," "), beforeState);
                  }
                }
                onSlidingComplete={(vrijednost) => {this.setState({ vrijednost });
                      console.log(c.trosilo.replace("\n"," "), vrijednost);
                      this.afterToggleState(vrijednost);
                  }
                }
              />
              </View>
```

Slika 9 - Primjer kôda u React Native (Izrada autora)

Iz te datoteke se učita ime prekidača, vrsta trošila te se ovisno o vrsti trošila generiraju unaprijed učitane slike, koristeći switch naredbu na vrsti trošila.

Na slici 12 vidimo različite mogućnosti „formatiranja“ izgleda, čitajući iz JSON skripte.



Slika 10 - Izgled aplikacije MojDom nakon učitane JSON datoteke (izrada autora)

Mobilna se aplikacija, prema svojoj namjeni i funkcionalnostima, donekle razlikuje od web aplikacije, naime, zamišljena je kao udaljen pristup sustavu MojDom i spaja se preko Interneta dok je web varijanta zamišljena za kontrolu unutar etherneta, stoga, nema svrhe da se zaustavljaju rolete u nekoj poziciji kad se nema uvid na poziciju roleta. Također, nije implementirana funkcionalnost prigušivanja svijetla i mijenjanja boje .

Ključna funkcija je slanje na server prilikom klika na On/Off „toggle“:

```
postToServer(pin, status) {
  testbody = "pin="+pin+"&"+"status="+status;
```

```
 const requestBody = testbody;
 //console.log(testbody);
 fetch("https://wifi.xxxxxxxxx.hr/writeStatus.php", {
    method: 'POST',
    headers: {
      'Content-Type': 'application/x-www-form-urlencoded;charset=UTF-8'
    },
    body: requestBody
  }).then(function(response) {
    return response.text().then(function(text) {
      console.log(text);
    });
  })
  .catch(e => console.log(e));
}
```

Da bi navedena funkcija radila, server mora imati instaliran valjani ssl certifikat zbog da bi se zaštitili od cross-site scriptinga[6]. Više o načinu preuzimanja datoteka, teksta i drugog, koristeći naredbu fetch, može se pronaći na stranicama (https://fetch.spec.whatwg.org/, n.d.)

9. Praktični dio rada

Praktični dio ovog rada, zbog opće dostupnosti, postavljen je na mrežnim stranicama githuba https://github.com/EndiGlavinic/MojDom (Glavinić, 2018) te sadrži izvorni kod programa, slike i dijagrame vezane za ovaj rad, te video prezentaciju sustava MojDom u primjeni.

[6] „Tip propusta u računalnoj sigurnosti, tipično se odnosi na web i mobilne aplikacije. XSS dozvoljava napadaču injekciju klijentske skripte (npr. JavaScript) u mrežnu stranicu koja je kasnije pregledna drugim korisnicima. Može se iskoristit kako bi se zaobišla sigurnost i samim time izgubila kontrola nad mrežnom stranicom, ili kako bi se ukrale informacije korisnika“ (Pascale, 2008)

10. Zaključak

Automatizacija i kontrola proizvodno-tehnoloških procesa je skupa, ogromna infrastruktura, stoga nije primjenjivana u kućanstvima. Minijaturizacijom i širom dostupnošću sve većeg broja hardverskih komponenti koje zadovoljavaju i industriju i ljude koji se time bave iz hobija, došlo je do razvoja kućne automatizacije. Na taj način moguće je dobiti ekvivalent industrijskom okruženju, uključujući senzore za temperaturu, svjetlost, vlagu, plin, pritisak, potrošnju struje, vode, plina, razne aktuatore, video nadzor, prepoznavanje lica, glasa ili otiska prsta puno jeftinije nego u industiji. Internet of Things (IoT) omogućava svakom uređaju da bude spojen na Internet, te samim time upravljiv. Omogućavanje praćenja potrošnje energenata i ostale senzorike poput temperature, vjetra i svjetla, pomaže kod efikasnijeg upravljanja kućom (npr. zatvaranja roleta zimi popodne kako bi zadržali toplinu u kući ili ujutro ljeti, kako bi spriječili ulazak topline čime smanjujemo potrebu za hlađenjem/grijanjem). Kućna automatizacija donosi mogućnost udaljenog upravljanja sustavom grijanja/hlađenja pa daljinski možemo postaviti temperaturu prilikom dolaska kako bi nas dočekao ugodan dom.

Ovaj rad prikazije istraživanje na polju informacijskih i komunikacijskih znanosti i elektrotehnike, a vezano uz udaljeno upravljanje kućnom automatikom. Iako na tržištu postoji veliki broj različitih sustava kućne automatizacije, ovim radom, analizirano je postojeće stanje, obrađeni vrste i protokoli komunikacije, te ostvareno rješenje problema visoke cijene dizajnom sustava koji je po svoj prirodi „Open Source“ i cjenovno vrlo pristupačan kako profesionalcima tako i hobi majstorima.

Kao jednu od boljki sustava kućne automatizacije, a zbog minijaturizacije komponenti izdvajamo sigurnost. Zbog manjih i slabijih komponenti nego kod velike braće u industriji često ne možemo integrirati sigurnost na onoj razini na kojoj bi željeli pa je zabilježen porast preuzimanja IoT uređaja za DDoS napade, prema Maineu (2017), čak 91% između III. i I.kvartala 2017. godine. Stoga je za ovaj rad odabrana nekriptirana komunikacija između OpenWrt-a i Arduina a od OpenWrt-a, koji je na unutrašnjoj mreži, postavljena je 256 bitna zaštita prema van.

Iako demonstrira osnovu kućnoj automatizaciji te udaljeni pristup istoj, zbog svoje strukture MojDom je lako nadogradiv. Kao jedan budući istraživački projekt može se identificirati implementacija daljinskog otvaranja vrata, integracije sustava za kontrolu pristupa (ulazna vrata, dvorišna i garažna vrata sa povratnom vezom), sustava video nadzora, upravljanja multimedijalnim sustavom, klima uređajem (osim uključivanja/isključivanja, također kontrola postavki), sustava zalijevanja vrta/travnjaka sa povratnom vezom i slično. Također, trebalo bi razviti sustava pametne kuće sa mogućnošću strojnog učenja i umjetne inteligencije ili iskoristiti postojeće rješenje na projektu MojDom.

Ograničenja ovog sustava, poput nestanka struje ili pada internetske veze, moguće je riješiti akumulatorima, agregatom te redundancijom napajanja i internetske veze.

Autor se nada će nekome ovaj tekst te prilozi na Github-u poslužiti kao motivacija u upotrebi rješenja otvorenog koda za implementaciju istog u svom domu.

Reference

ArduinoCC, n.d. *Arduino Nano.* [Mrežno] Dostupno na: https://store.arduino.cc/arduino-nano [Pristupljeno 02.02.2018.].

Arduino, n.d. [Mrežno] Dostupno na: https://www.arduino.cc/en/Main/AboutUs [Pristupljeno 16.02.2018.].

Brown, R., 2016. *Welcome to the OpenWrt Project.* [Mrežno] Dostupno na: https://openwrt.org [Pristupljeno 17.02.2018.].

CastelarArticles.com, n.d. *Smart Home Technology: Changing the Way Houses Operate.* [Mrežno] Dostupno na: http://articles.castelarhost.com/smart_home_technology.htm [Pristupljeno 29.01.2018.].

Cisco, n.d. *www.cisco.com.* [Mrežno] Dostupno na: https://www.cisco.com/c/en/us/solutions/small-business/resource-center/connect-employees-offices/networking-basics.html [Pristupljeno 15.01.2018.].

Evans, J. M. J. i dr., 1976. *Standards for computer aided manufacturing.* Washington D.C.: National Bureau of Standards.

Facebook, 2017. *React Native - Build native mobile apps using JavaScript and React.* [Mrežno] Dostupno na: https://facebook.github.io/react-native/ [Pristupljeno 30.01.2018.].

Foundation, T. j., 2018. *jQuery.* [Mrežno] Dostupno na: https://jquery.com/ [Pristupljeno 04.02.2018.].

Glavinić, E., 2018. *MojDom - software i hardware za upravljanje kućnom automatizacijom.* [Mrežno] Dostupno na: https://github.com/EndiGlavinic/MojDom [Pristupljeno 18.02.2018.].

Maine, M., 2017. DDoS attacks have doubled in six months, up 91% on first quarter [Mrežno] Dostupno na: https://www.scmagazineuk.com/ddos-attacks-have-doubled-in-six-months-up-91-on-first-quarter/article/709147/ [Pristupljeno: 16.02.2018.]

Murugan, V., 2008. *Introductory PLC Programming.* [Mrežno] Dostupno na: https://en.wikibooks.org/wiki/Introductory_PLC_Programming [Pristupljeno 18.02.2018.].

OceanControls.com, n.d. *KTA 223 Manual v12.* [Mrežno] Dostupno na: https://oceancontrols.com.au/files/datasheet/ocean/KTA-223_Manual_v12.pdf [Pristupljeno 15.02.2018.].

OceanControls.com, n.d. *Relayduino USB/RS-485 IO Module 12 VDC.* [Mrežno] Dostupno na: http://oceancontrols.com.au/KTA-223.html [Pristupljeno 17.02.2018.].

opensource.com, n.d. *What is open source?.* [Mrežno] Dostupno na: https://opensource.com/resources/what-open-source [Pristupljeno 2018 2 18].

Pascale, K., 2008. *Cross-site Scripting (XSS).* [Mrežno] Dostupno na: https://www.owasp.org/index.php/Cross-site_Scripting_(XSS) [Pristupljeno 15.02.2018.].

Patonnier, J., 2014. *Learn to syle HTML using CSS.* [Mrežno] Dostupno na: https://developer.mozilla.org/en-US/docs/Learn/CSS [Pristupljeno 04.02.2018.].

php.net, n.d. *What is PHP?.* [Mrežno] Dostupno na: http://php.net/manual/en/intro-whatis.php [Pristupljeno 17.02.2018.].

Planetleak, 2014. *Arduino SSD1306 128x64 OLED LCD SPI - Hello World.* [Mrežno] Dostupno na: https://www.youtube.com/watch?v=shIeYpZAEd0 [Pristupljeno 15.01.2018.].

Rastogi, T. & Sharma, R. R., 2014. *Home automation system design: the basics.* [Mrežno] Dostupno na: https://www.embedded.com/design/connectivity/4431025/Home-automation-system-design--the-basics [Pristupljeno 17.02.2018.].

Riley, M., 2012. *Programming Your Home - Automate with Arduino, Android and Your Computer.* Dallas, Texas - Raleigh, North Carolina: The Pragmatic Bookshelf .

Srdić, S., 2016. *Pametne kuće u Hrvatskoj - mit ili stvarnost.* [Mrežno] Dostupno na: http://planb.hr/staging/pametne-kuce-hrvatskoj-mit-stvarnost/ [Pristupljeno 29.01.2018.].

Technopedia.com, n.d. [Mrežno] Dostupno na: https://www.techopedia.com/definition/28408/maker-movement [Pristupljeno 18.02.2018.].

Terrien, A., 2015. *jQuery Knob.* [Mrežno] Dostupno na: http://anthonyterrien.com/knob/ [Pristupljeno 04.02.2018.].

TSTBiometrics.com, n.d. *Home Control and Automation Explained.* [Mrežno] Dostupno na: http://www.tst-biometrics.com/[Pristupljeno 20.01.2018.].

Webtech, n.d. *HTML5.* [Mrežno] Dostupno na: http://www.webtech.com.hr/html5.php [Pristupljeno 04.02.2018.].

Whatwg.org/, n.d. *Fetch- Living standard.* [Mrežno] Dostupno na: https://fetch.spec.whatwg.org/ [Pristupljeno 15.02.2018.].

Wikipedia, n.d. *Network topology.* [Mrežno] Dostupno na: https://en.wikipedia.org/wiki/Network_topology [Pristupljeno 10 2 2018].

Wood, T., 2015. *momentjs.com.* [Mrežno] Dostupno na: http://momentjs.com/ [Pristupljeno 04.02.2018.].

www.bq.com, 2014. *Nano Pinout.* [Mrežno] Dostupno na: www.bq.com [Pristupljeno 15.02.2018.].

Popis slika

Slika 1 - Slikovni prikaz mrežnih komunikacija Izvor: (Wikipedia, n.d.) 6
Slika 2 - Dizajn sustava (izrada autora) 8
Slika 3- Arduino Nano (ArduinoCC, n.d.) 11
Slika 4- Raspored pinova na Arduino Nano (www.bq.com, 2014) 12
Slika 5 - Arduino razvojno okruženje sa kôdom za "Hello World" (Planetleak, 2014) 12
Slika 6 - Tiskana pločica usmjerivača ASUS RT-N14U, crveno je označene mjesto za pin header (Izrada autora) 15
Slika 7 - Omogućavanje SSH pristupa (Izrada autora) 16
Slika 10 - Izgled web sučelja napisan u HTML5, JQuery, Bootstrap, CSS (Izrada autora) 26
Slika 11 - Primjer kôda u React Native (Izrada autora) 28
Slika 12 - Izgled aplikacije MojDom nakon učitane JSON datoteke (izrada autora) 29

Popis tablica

Tablica 1 - Komande za komunikaciju sa Arudinom (izvor: Oceans Controls)..... 21

Sažetak

Ovaj se rad bavi primjenom Arduina i usmjerivača u automatizaciji kuće. U uvodu se objašnjava pojam pametne kuće, automatizacije kuće i benefita istih. Nakon toga predstavlja se korišteni hardver i softver te se čitaocu pojašnjava kako podesiti sustavske postavke za repliciranje sustava. Uz korištenje usmjerivača i Arduina u automatizaciji, navode se korišteni programski jezici te njihovi benefiti. Na kraju se definiraju prednosti i nedostaci sustava te ukratko opisuju moguća poboljšanja.

Ključne riječi: kućna automatizacija, pametne kuće, Arduino, OpenWrt, usmjerivač, ReactNative, PHP

Abstract

This thesis examines the use of Arduino and router in home automation. Introduction explains terms like smart home, home automation and benefits of using it. After introduction we present used used hardware and software and we explain how to setup system parametres for replication. Beside used hardware later in the text is explanation of used computer languages and technique. Finnaly there are described benefits and disadvantages of the system and possible improvements.

Keywords: home automation, smart home, Arduino, OpenWrt, router, ReactNative, PHP