

Bežični nadzor i upravljanje industrijskih procesa

Leš, Marko

Master's thesis / Diplomski rad

2015

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:235:056347>

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

Download date / Datum preuzimanja: **2024-12-04**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Marko Leš

Zagreb, 2015.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentor:

Prof. dr. sc. Željko Šitum

Student:

Marko Leš

Zagreb, 2015.

IZJAVA

Izjavljujem da sam ovaj rad izradio samostalno služeći se znanjima stečenim tijekom studija, u slobodno vrijeme i navedenom literaturom.

ZAHVALE

Zahvaljujem svojem mentoru Prof. dr. sc. Željku Šitumu na ukazanom povjerenju prihvaćanjem mentorstva za ovaj rad te za vođenje kroz izradu s korisnim savjetima.

Zahvaljujem tvrtki ELEKTRON ERMA-STRMEC d.o.o. na ustupljenoj opremi, dijelovima i prostoru za izradu diplomskog rada.

Zahvaljujem obitelji na strpljenju i neprestanoj potpori pruženoj kroz studij.

Zagreb, svibanj 2015.

Marko Leš



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo
materijala i mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur.broj:	

DIPLOMSKI ZADATAK

Student: **MARKO LEŠ**

Mat. br.: 0035177826

Naslov rada na hrvatskom jeziku: **BEŽIČNI NADZOR I UPRAVLJANJE INDUSTRIJSKIH PROCESA**

Naslov rada na engleskom jeziku: **WIRELESS MONITORING AND CONTROL OF INDUSTRIAL PROCESSES**

Opis zadatka:

Suvremena industrijska rješenja automatiziranih procesa obuhvaćaju prikupljanje mjernih podataka s velikog broja različitih senzora te slanje odgovarajućih upravljačkih signala na različite izvršne članove u cilju pravilnog odvijanja procesa. Za regulaciju procesnih veličina u industrijskim postrojenjima vrlo često su ugrađeni različiti procesni i programski regulatori te pokazivači i signalizatori koji korisniku omogućuju lakši nadzor procesa. Pri tome se javljaju problemi istovremenog prikaza stanja više procesnih veličina, nadzor većeg broja uređaja s jednog ili više udaljenih mjesta, složen postupak očitavanja korisničkih podataka zbog različitih komunikacijskih protokola i dr.

U zadatku je potrebno:

1. razviti web aplikaciju koja korisniku omogućuje pristup s različitih uređaja (npr. mobitel, tablet, laptop) neovisno o operacijskom sustavu preko web preglednika,
2. izraditi vlastiti uređaj temeljen na jeftinom minijaturnom računalu Raspberry Pi koji omogućuje nadzor i upravljanje pojedinačnih podređenih uređaja (pokazivača i regulatora) putem bežične mreže pomoću web preglednika,
3. opisati proceduru postavljanja svih programa i parametara na uređaju, komunikacijske protokole i način rada uređaja,
4. ispitati rad uređaja na realnom industrijskom procesu.

Zadatak zadan:

12. ožujka 2015.

Rok predaje rada:

14. svibnja 2015.

Predviđeni datum obrane:

20., 21. i 22. svibnja 2015.

Zadatak zadao:

Prof. dr. sc. Željko Šitum

Predsjednik Povjerenstva:

Prof. dr. sc. Franjo Cajner

SADRŽAJ

SADRŽAJ	I
POPIS SLIKA	II
POPIS TABLICA.....	IV
SAŽETAK.....	V
SUMMARY	VI
1. UVOD.....	1
2. OPIS SUSTAVA	3
3. UREĐAJ ZA NADZOR I UPRAVLJANJE	6
3.1. Raspberry Pi.....	6
3.2. Komunikacijski protokoli i mreže u industriji	8
3.2.1. RS-485	8
3.2.2. Ethernet	8
3.2.3. Modbus	9
3.2.4. Wi-Fi	12
3.3. Model i električna shema uređaja	14
4. WEB APLIKACIJA ZA NADZOR I UPRAVLJANJE.....	18
4.1. Internet	19
4.2. Web alati	20
4.3. Opis web aplikacije.....	24
4.4. Skripta za prikupljanje podataka.....	34
5. POSTAVLJANJE WEB APLIKACIJE NA RASPBERRY PI.....	35
5.1. Instaliranje operativnog sustava.....	35
5.2. Podešavanje Wi-Fi pristupne točke.....	38
5.3. Instaliranje LAMP stack-a	42
5.3.1. Apache web server sa PHP-om.....	42
5.3.2. MySQL	43
5.3.3. FTP server.....	43
5.3.4. phpMyAdmin.....	45
5.4. Inicijalizacija Web aplikacije.....	46
6. ISPITIVANJE NA REALNOM INDUSTRIJSKOM PROCESU	47
7. ZAKLJUČAK.....	50
LITERATURA.....	51
PRILOZI.....	52

POPIS SLIKA

Slika 1.	Regulacija temperature ekstrudera sa 4 GAMA-3k	3
Slika 2.	Shema sustava	5
Slika 3.	Raspberry Pi 2 Model B	7
Slika 4.	Raspberry Pi Compute Module	7
Slika 5.	Wi-Fi logo korišten od Wi-Fi Alianse-a.....	12
Slika 6.	Primjer bežične mreže	13
Slika 7.	Uređaj za nadzor i upravljanje.....	14
Slika 8.	Edimax EW-7811Un	15
Slika 9.	Uređaj za nadzor i upravljanje – dijelovi	16
Slika 10.	Električna shema dodatne pločice	17
Slika 11.	Razlika između klasične i web aplikacije.....	18
Slika 12.	Odnos server – klijent.....	19
Slika 13.	Usporedba HTML-a, CSS-a i Javascript-a.....	20
Slika 14.	Prilagodljiva web stranica veličini ekrana uređaja.....	21
Slika 15.	Način rada servera sa PHP-om.....	22
Slika 16.	Ažuriranje sadržaja web stranice koristeći AJAX.....	23
Slika 17.	MVC - pozivanje funkcija.....	24
Slika 18.	MVC - struktura direktorija.....	25
Slika 19.	Kontroleri u direktoriju controller	26
Slika 20.	Relacije između tablica u MySQL bazi podataka	27
Slika 21.	Prijava u web aplikaciju	28
Slika 22.	Usporedba navigacije administratora i korisnika	29
Slika 23.	Početna stranica web aplikacije.....	29
Slika 24.	Popis svih konfiguriranih uređaja.....	30
Slika 25.	Postavljanje novog objekta.....	30
Slika 26.	Pregled parametara jednog objekta	31
Slika 27.	Pregled parametara objekta u grafu.....	32
Slika 28.	Obrazac za registraciju novog korisnika	33
Slika 29.	Win32 Disk Imager za instalaciju operacijskog sustava Raspbian	35
Slika 30.	Raspberry Pi konfiguracijski meni	36

Slika 31.	Putty –program za pristup uređaju preko SSH protokola.....	37
Slika 32.	Izgled phpMyAdmin sučelja za upravljanje MySQL-om	45
Slika 33.	WinSCP – prebacivanje datoteka na server preko mreže.....	46
Slika 34.	Mogućnosti PID regulatora GAMA-3k.....	47
Slika 35.	Ispitivanje uređaja na regulatoru GAMA-3k.....	48
Slika 36.	Graf temperature.....	49

POPIS TABLICA

Tablica 1. Usporedba Raspberry Pi pločica	6
Tablica 2. Adrese podataka u tablicama.....	9
Tablica 3. Funkcijski kodovi	10
Tablica 4. Modbus RTU poruka.....	11
Tablica 5. Modbus ASCII poruka	11
Tablica 6. Modbus TCP poruka	12
Tablica 7. Wi-Fi standardi.....	13

SAŽETAK

Zadaća ovog rada je ponuditi rješenje za bežični nadzor i upravljanje industrijskim procesima, odnosno regulatorima i pokazivačima koji podržavaju Modbus RTU protokol preko RS-485 i Modbus TCP preko Ethernet mreže. Napravljen je uređaj temeljen na minijaturnom i jeftinom računalu Raspberry Pi na kojem se nalazi web aplikacija, koja korisniku omogućuje bežično na zaslonu mobilnog uređaja, tablet ili stolnom računalu prikaz svih parametara, te grafički prikaz mjerenih varijabli. Opisani su svi dijelovi za predloženi uređaj, te sama struktura web aplikacije i grafičko sučelje. Na kraju je uređaj ispitan na realnom industrijskom procesu.

Ključne riječi: Raspberry Pi, web aplikacija, regulacija, nadzor, upravljanje, proces, Modbus

SUMMARY

This paper aims to provide a solution for wireless monitoring and control of industrial processes, i.e. regulators and indicators that support the Modbus RTU protocol via RS-485 and Modbus TCP over Ethernet networks. A device was made based on a miniature and inexpensive computer Raspberry Pi which has a web application that allows the user to wirelessly on the screen of the mobile device, tablet or desktop computer display of all parameters and graphic display of the measured variables. All parts of the proposed device, the structure of web application and graphical interface are described. At the end the device was tested on a real industrial process.

Key words: Raspberry Pi, web application, regulation, monitoring, control, process, Modbus

1. UVOD

U automatizaciji industrijskih procesa, suvremena rješenja obuhvaćaju prikupljanje mjernih podataka sa velikog broja različitih senzora, te slanje odgovarajućih upravljačkih signala na različite izvršne članove u cilju pravilnog odvijanja procesa. Za regulaciju procesnih veličina u industrijskim postrojenjima vrlo često se koriste različiti procesni i programski regulatori, te pokazivači i signalizatori koji korisniku omogućuju lakši nadzor procesa. U industrijskom postrojenju vrlo često su regulatori i pokazivači ugrađeni u upravljačke ormare koji se nalaze blizu nekog uređaja kojim se upravlja. Takvih uređaja može biti i više u industrijskom postrojenju, te mogu biti različiti. Za pravilno odvijanje nekog industrijskog procesa svi ovi uređaji moraju biti usklađeni preko nekog centralnog upravljačkog sustava.

Da bi to bilo moguće potrebno je sve regulatore i pokazivače u industrijskom postrojenju povezati međusobno na neku lokalnu mrežu (RS-485, Ethernet, CAN). Na toj mreži potrebno je imati i uređaj koji će tim svime upravljati, u većini slučajeva neko računalo. Takav sustav se naziva SCADA (eng. *Supervisory Control And Data Acquisition*), te predstavlja računalni sustav za nadzor, mjerenje i upravljanje industrijskim sustavima. U industrijskim postrojenjima mogu se nalaziti uređaji od različitih proizvođača, koji komuniciraju nekim od mnogobrojnih komunikacijskih protokola. Za nadzor i upravljanje svim uređajima s jednog mjesta potrebno je još kupiti neki gotovi ili napraviti svoj programski paket koji će omogućiti prikupljanje svih podataka s pojedinih uređaja, njihov prikaz, te zadavanje određene reference svakom od uređaja.

Tvrtka Elektron Erma – Strmec od samog početka teži proizvodnji kvalitetne opreme za mjerenje, regulaciju, procesno upravljanje te pružanje što je moguće bolje korisničke podrške. U svom proizvodnom programu nude različite senzore temperature, vlage, prisutnosti, razine, protoka, tlaka, dok kao izvršne članove nude različite ventile koji mogu biti sa elektromotornim ili pneumatskim pogonom, te kontinuirane tiristorske module za regulaciju izlazne struje. Za prikazivanje mjerenih vrijednosti nude pokazivače i signalizatore, te za regulaciju procesa različite industrijske, procesne i programske regulatore. Prikaz svih parametara uređaja te mjerenih veličina omogućen je pomoću jednostavnog pokazivača, na kojem je moguće samo jednu ili dvije vrijednosti istovremeno prikazivati. Za prikaz ostalih veličina koriste se tipke, te je potrebno detaljno uputstvo za prikaz ili promjenu svih

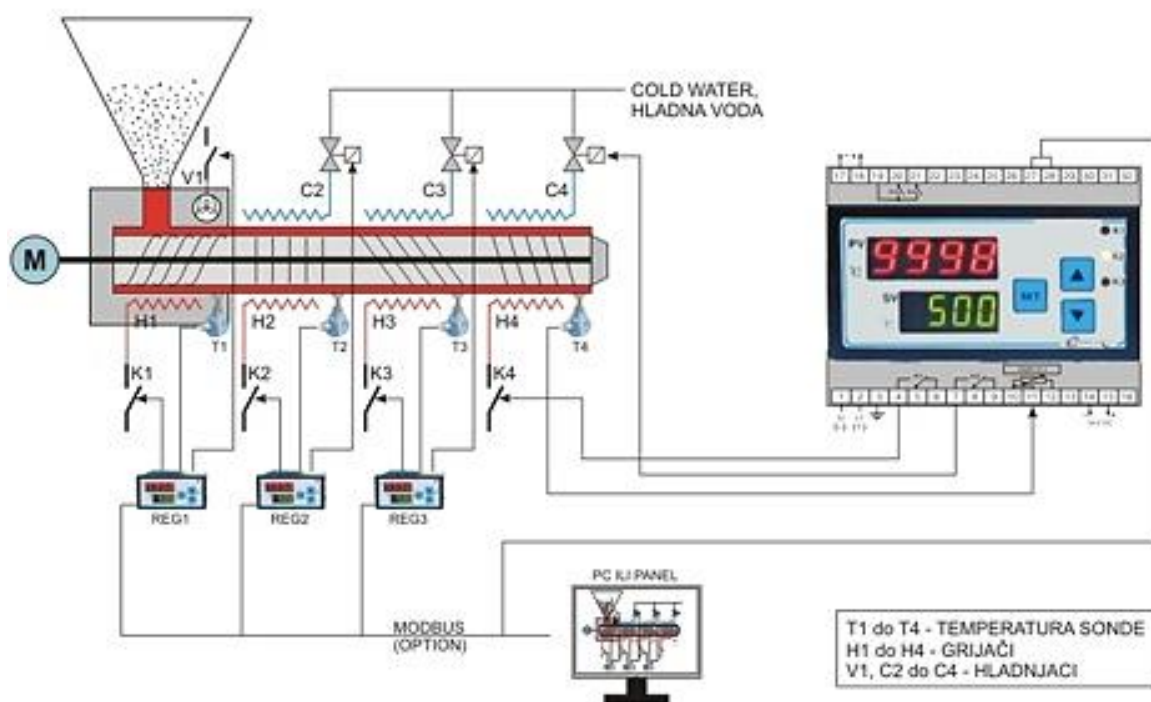
parametara. Skoro svi uređaji ima podršku za Modbus RTU protokol preko RS485 komunikacije, te većina ima podršku za puno brži Modbus TCP protokol preko Ethernet mreže. Logiranje je moguće na mikro SD karticu ili USB memoriju, koji se nalaze sa stražnje strane uređaja.

Problemi koji se ovdje pojavljuju su prikaz više veličina istovremeno, nadzor više uređaja s jednog ili više udaljenih mjesta, te kompliciran proces kod očitavanja logiranih veličina. Preko Modbus protokola omogućeno je da korisnik ima neki svoj SCADA sustav, pošto tvrtka nema za sve uređaje razvijen svoj SCADA sustav. Kod tog procesa potreban je podatak korisniku o adresama svih parametara, vrsti parametara, njihove maksimalne i minimalne vrijednosti, te preciznost, dok se za svaki uređaj te vrijednosti razlikuju. U posljednje vrijeme raste popularnost bežičnih mreža u industrijskoj automatizaciji, pošto oprema postaje jeftinija, a pouzdanost i sigurnost takvih mreža raste. Isto tako i cijena tablet računala postaje manja, te je želja korisniku nekog automatiziranog sustava omogućiti da pomoću svog mobilnog telefona ili tablet uređaja nadzire taj sustav ili s njim upravlja bežično u industrijskom postrojenju.

Kao rješenje ovih problema ideja je napraviti uređaj temeljen na jeftinom minijaturnom računalu Raspberry Pi koji bi omogućio nadzor i upravljanje ostalih uređaja (pokazivača i regulatora) na mreži preko zaslona mobilnog telefona, tablet uređaja ili računala preko bežične ili lokalne mreže bez instaliranja dodatnih programa, samo pomoću web preglednika. U ovom radu opisano je kako bi predloženi sustav funkcionirao, detaljnije je opisan fizički uređaj koji bi se izrađivao, web aplikacija koja se nalazi na tom uređaju, te kako se postavlja i podešava web aplikacija. Na kraju su prikazani rezultati ispitivanja na realnom industrijskom procesu, te zaključak.

2. OPIS SUSTAVA

U klasičnim industrijskim pogonima često više regulatora upravlja nekim kompleksnim procesom. Npr. kod regulacije temperature ekstrudera nalazi se više regulatora i svaki nadzire pojedini segment [Slika 1]. Regulatori su montirani na upravljačke ormare, a ako postoji više ekstrudera moguće je sve regulatore montirati u jedan upravljački ormar kako bi se s jednog mjesta upravljalo sa svime. Kako tvrtka Elektron Erma – Strmec nudi širok asortiman proizvoda moguće je da se u jednom industrijskom postrojenju nalazi više njihovih regulatora ili pokazivača na različitim uređajima. Ako se želi s jednog mjesta nadgledati sve uređaje, međusobno spojiti preko RS-485 ili Ethernet mreže, te postaviti centralni nadzorni uređaj ili računalo. Za nadzor potreban je i odgovarajući programski paket koji će omogućiti vizualizaciju, pošto tvrtka nema svoj program. Takav program bi trebalo izraditi, pri čemu je potrebno poznavati adrese parametara svih uređaja, koje se razlikuju ovisno o modelu. Program pisan u izvornom jeziku, mogao bi raditi samo na operativnom sustavu za koji je napravljen, dok bi za druge operativne sustave trebalo prilagođavati program, odnosno aplikaciju.



Slika 1. Regулacija temperature ekstrudera sa 4 GAMA-3k

Želja tvrtke Elektron Erma – Strmec je imati u svom asortimanu uređaj na koji bi se mogli spojiti svi njihovi uređaji, te omogućiti korisniku lakši nadzor i upravljanje bez instaliranja dodatnih programa. Kako su posljednjih godina mobilni telefoni i tablet računala evoluirala, postali su brži od nekih industrijskih računala koja i danas obavljaju svoju funkciju. Uređaj koji bi omogućio nadzor i upravljanje bez instaliranja dodatnih programa također bi trebao omogućiti pristup sa mobilnih telefona i tablet računala bežično.

Zahtjevi za takav uređaj su sljedeći:

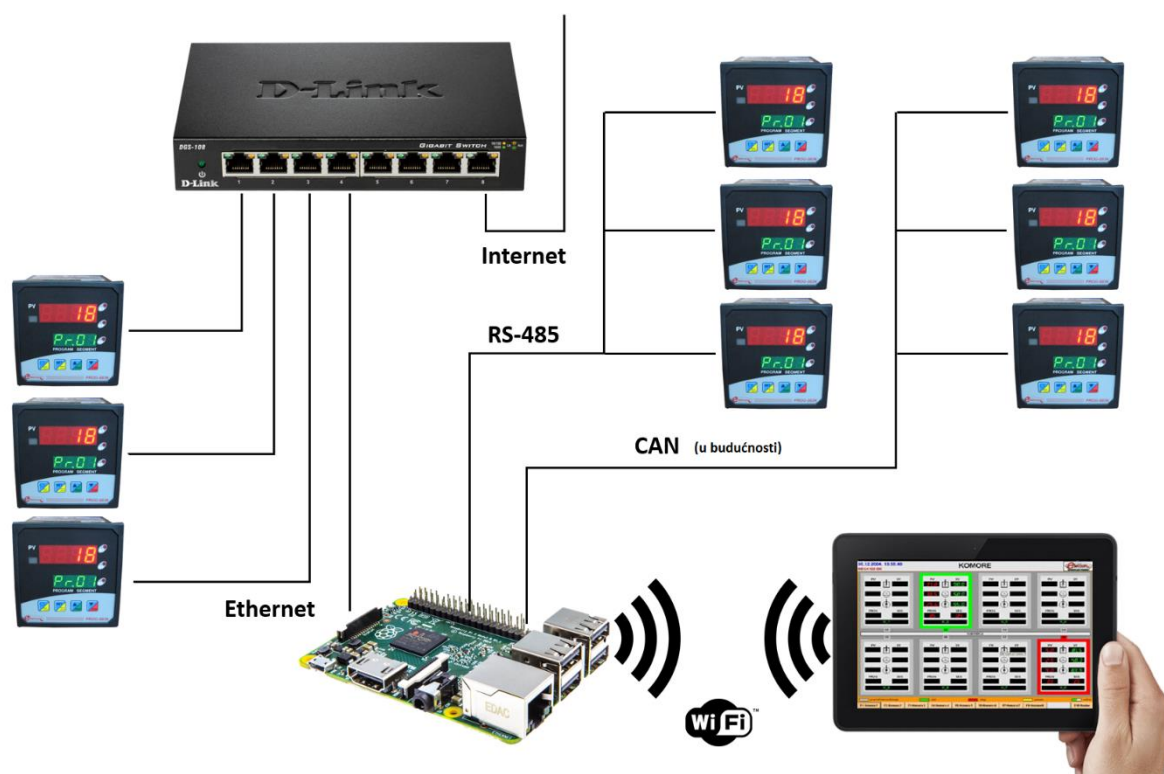
- Omogućiti nadzor i upravljanje bez instaliranja dodatnih programa
- Omogućiti pristup sa mobilnih telefona i tablet računala
- Omogućiti žičano i bežično povezivanje
- Podrška za Modbus RTU protokol preko RS-485 mreže
- Podrška za Modbus TCP protokol preko Ethernet mreže
- Buduća proširenja na CANopen
- Minijaturnih dimenzija
- Montaža u upravljački ormar
- Zaštita od elektromagnetskih smetnji
- Podrška spajanja do 32 regulatora ili pokazivača
- Buduća mogućnost upravljanja preko interneta

Kao odgovor na ove zahtjeve odlučeno je uređaj temeljiti na jeftinom minijaturnom računalu Raspberry Pi koji može pokrenuti operativni sustav Raspbian baziran na Linux-u, te će najnovija verzija omogućiti pokretanje operativnog sustava Windows 10, što je zanimljivo za buduću širu primjenu uređaja. Tijekom testiranja Raspberry Pi je pokazao veliku otpornost na elektromagnetske smetnje, koje su namjerno napravljene uključivanjem i isključivanjem releja u blizini procesora kako bi se testirala stabilnost sustava. Računalo na sebi ima Ethernet konektor, USB konektore, te pinove za moguća proširenja na kojima se nalaze SPI, I2C i UART komunikacije. Preko Ethernet konektora moguće je pločicu spojiti na lokalnu mrežu, te uz odgovarajući programsku podršku preko Modbus TCP protokola komunicirati sa regulatorima i pokazivačima. Na USB konektor može se spojiti jeftini Wi-Fi stick koji može raditi kao bežični klijent ili pristupna točka. Za spajanje na RS-485 mrežu te buduće na

CANopen potrebno je izraditi dodatnu pločicu koja bi sadržavala čipove za naponsku konverziju komunikacije. Kako je Raspberry Pi malenih dimenzija lako se može smjestiti u standardno kućište koje omogućava montažu na šinu unutar upravljačkog ormara.

Raspberry Pi radio bi kao web server, te bi se na njemu nalazila web aplikacija koja bi sadržavala i spremala sve informacije o sustavu i uređajima koji su spojeni na njega. Tako bi sve informacije bile spremljene na jedno centralno mjesto, te bi s jedne strane bili spojeni svi uređaji, a s druge bi se više korisnika moglo spojiti na web aplikaciju putem bežične mreže ili lokalne Ethernet mreže sa svojih mobilnih telefona, tablet ili stolnih računala putem web preglednika. Korisnici ne bi trebali ništa instalirati, nego samo poznavati lozinku za bežičnu mrežu i IP adresu uređaja. Kako se ne bi bilo tko mogao spojiti prijava u web aplikaciju bila bi preko korisničkih imena i lozinki koje bi identificirale svakog od korisnika.

Prednost ovakvog sustava prema klasičnom SCADA sustavu je pohrana svih informacija o uređajima na jednom mjestu i mogućnost spajanja više korisnika žičano ili bežično preko svojih uređaja.



Slika 2. Shema sustava

3. UREĐAJ ZA NADZOR I UPRAVLJANJE

Uređaj koji će omogućiti nadzor i upravljanje industrijskim procesima baziran je na Raspberry Pi-u. Ovdje su ukratko opisane neke karakteristike tog računala, komunikacijski protokoli i mreže, te model i električna shema uređaja.

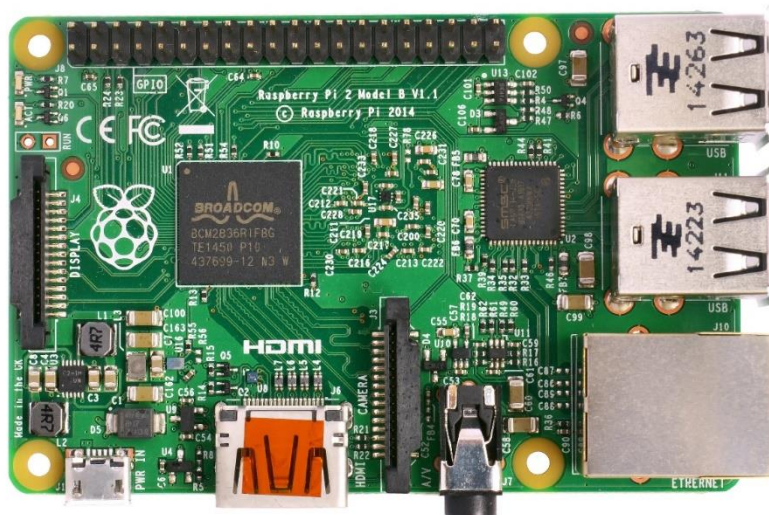
3.1. Raspberry Pi

Raspberry Pi je serija jeftinih računala veličine kreditne kartice, razvijena u UK od strane Raspberry Pi Foundation-a sa namjerom promocije učenja osnova računala u školama. Raspberry Pi je sposoban raditi sve što i svako obično stolno računalo, od pregledavanja Interneta i gledanja filmova visoke rezolucije, do pisanja tekstova, tablica i igranja igrice. Prije svega ima mogućnost interakcije sa okolnim svijetom, što ga čini idealnim za različite projekte. Postoje različite verzije [Tablica 1], a razlika je u brzini procesora, količini radne memorije i dostupnim pinovima procesora za spajanje druge periferije. Na pločicama se koriste Debian ili Arch Linux ARM distribucije operativnog sustava, koje se nalaze na SD memorijskoj kartici, dok je glavni programski jezik Python, sa podrškom i ostalih programskih jezika.

Tablica 1. Usporedba Raspberry Pi pločica

	Model B	Model B+	Model B (2. generacija)	Compute modul
Cijena	35\$			30\$
Procesor	700 MHz jednojezgreni ARM1176JZF-S		900 MHz četverojezgreni ARM Cortex-A7	700 MHz jednojezgreni ARM1176JZF-S
Radna memorija	512 MB (ranije 256 MB)	512 MB	1 GB	512 MB
Broj USB	2	4		-
Trajna memorija	SD memorijska kartica minimalno 2 GB	MicroSD memorijska kartica minimalno 2 GB		4 GB eMMC flash memorije na čipu
Broj pinova za proširenja	26 (17 GPIO)	40 (26 GPIO)		200 (46 GPIO)
Potrošnja	3.5 W	3 W	4 W	1 W
Dimenzija	85.6 x 56.5			67.6 x 30
Težina	45 g			7 g

Najnovija verzija pločice Raspberry Pi 2 model B [Slika 3] sadrži Broadcom BCM2836 sistem na čipu koji ima četverojezgreni ARM Cortex-A7 900 MHz procesor sa 1 GB radne memorije. Za sve pločice potrebno je napajanje sa stabilnim naponom 5V i strujom 1 A. Na pločici se nalaze četiri USB priključka, priključak za Ethernet, HDMI, 3.5 mm priključak za video i audio, te 40 pinova od kojih se neki mogu koristiti kao digitalni ulazi ili izlazi, a neki kao komunikacije I2C, SPI i UART.



Slika 3. Raspberry Pi 2 Model B

Za projekte kojima nisu potrebni svi konektori koje ima klasična pločica, postoji Compute modul [Slika 4] koji sadrži samo sistem na čipu (procesor i radna memorija) i eMMC flash memoriju sa SO-DIMM konektorom. Za takvu pločicu potrebno je izraditi vlastitu pločicu sa željenom periferijom, kao npr. više Ethernet konektora, čip za Wi-Fi, Bluetooth ili SATA 2 za spajanje tvrdih diskova, te stabilno napajanje za modul i periferiju.



Slika 4. Raspberry Pi Compute Module

3.2. Komunikacijski protokoli i mreže u industriji

Za povezivanje uređaja u industriji, danas postoje različita mrežna rješenja i komunikacijski protokoli. Neka od tih rješenja su brzi Ethernet koji povezuje računala i PLC-ove, DeviceNet koji se koristi u opasnim okolinama, Profibus, CANopen, te Modbus za povezivanje više uređaja preko RS-232, RS422 i RS-485 mreža. Ovdje će se ukratko opisati današnje dvije najpopularnije mrežne tehnologije i komunikacijski protokol koji će se implementirati na uređaj za nadzor i upravljanje, te Wi-Fi kao budućnost bežičnog povezivanja u industriji.

3.2.1. RS-485

RS-485 je serijska tehnologija prijenosa podataka, kod koje se koristi upletena parica. Brzine koje se mogu ostvariti su od 100 kbit/s do 35 Mbit/s ovisno o duljini linije, maksimalno do 1200 m. Svi uređaji spojeni su u sabirničku strukturu, a po segmentu je dozvoljeno do maksimalno 12 uređaja. U svakom trenutku dozvoljen je prijenos samo jedne poruke u jednom smjeru između sugovornika. RS-485 je half-duplex veza koja za prijenos podataka koristi dvije linije (A i B) kroz koje se istovremeno prenosi isti podatak, a informacija na strani prijemnika da li je podatak binarna jedinica ili nula, dobiva se razlikom napona između tih dviju linija. Zajednički napon je u opsegu -7 V do +12 V, a zbog diferenciranja signala i niskoomskog zaključanja, RS-485 je puno otporniji na utjecaj smetnji od RS-232.

3.2.2. Ethernet

Ethernet (IEE 802.3) je najčešće korištena sabirnička tehnologija za lokalne mreže (LAN). Protokol definira fizičke karakteristike priključka, električna svojstva signala i oblik paketa koji putuje Ethernet medijem. Brzine koje se mogu ostvariti su od 10 Mbit/s do 100 Mbit/s, dok najnoviji standardi imaju brzinu od 1 Gbit/s do 10 Gbit/s. Pošto se samo jedan paket može istovremeno emitirati, Ethernet koristi mehanizam CSMA/CD (eng. *carrier sense multiple access / collision detection*) koji sprečava da računala međusobno interferiraju. Za jednostavnu lokalnu mrežu potrebna su dva računala sa ugrađenim Ethernetom, te switch koji će upravljati protokom podataka, a za spajanje te lokalne mreže na internet potreban je modem, odnosno router.

3.2.3. Modbus

Modbus je serijski komunikacijski protokol izvorno zamišljen za upotrebu s programabilnim logičkim kontrolerima (PLC), no zbog svoje jednostavnosti i lake dostupnosti danas je to praktički industrijski standard promjenjiv u raznim elektroničkim uređajima. Uređaj koji zahtjeva informacije naziva se Modbus Master, a uređaj koji ih daje je Modbus Slave. U standardnoj Modbus mreži postoji jedan Master i 247 Slave-ova, sa jedinstvenim adresama od 1 do 247. Modbus je otvoreni protokol, što znači da ga proizvođači mogu implementirati u svoje uređaje besplatno. Tipično se koristi za primanje i slanje informacija sa uređaja prema glavnom kontroleru ili sustavu za prikupljanje podataka, kao npr. sustav mjerenja temperature i vlage šalje rezultate računalu. Često se koristi u sustavima nadzora i prikupljanja podataka (SCADA), kako bi se centralno računalo povezalo sa udaljenim terminalnim jedinicama (RTU). Za serijsku komunikaciju (RS-485) postoji Modbus RTU i Modbus ASCII, dok za Ethernet mrežu postoji Modbus TCP.

Kod serijskih komunikacija samo Master može inicijalizirati komunikaciju, dok kod Ethernet mreže, bilo koji uređaj može odaslati Modbus komandu, ali obično je to samo jedan Master. Modbus komanda sadrži adresu uređaja s kojim se želi komunicirati, te će samo taj uređaj odgovoriti, bez obzira što će ju svi uređaji primiti. To je iznimka kod slanja neke poruke svim uređajima pri čemu je vrijednost adrese 0. Sve Modbus komande imaju i provjeru informacija, kako bi se detektirale pogreške komunikacije.

Informacije u Slave uređajima su spremljene u četiri različite tablice [Tablica 2]. Dvije tablice imaju spremljene diskretne vrijednosti, a dvije numeričke. Svaka tablica ima 9999 vrijednosti u koje se mogu spremiti podaci. Na jednu adresu moguće je spremiti 2 bajta podataka, odnosno decimalni pozitivni broj u rasponu od 0 do 65536, dok je za veće ili decimalne brojeve potrebno koristiti više adresa.

Tablica 2. Adrese podataka u tablicama

Broj vrijednosti	Adresa podatka	Tip	Ime tablice
1 – 9999	0000 do 270E	Čitaj / piši	Diskretni izlazi
10001 – 19999	0000 do 270E	Samo čitaj	Diskretni ulazi
30001 – 39999	0000 do 270E	Samo čitaj	Analogni ulazni registri
40001 – 49999	0000 do 270E	Čitaj / piši	Analogni izlazni <i> Holding</i> registri

Kako bi uređaj koji prima zahtjev znao što treba napraviti i odgovoriti postoje standardni funkcijski kodovi [Tablica 3]. Oni najčešće korišteni (u tablici podebljani) koriste se za postavljanje ili čitanje stanja jednog ulaza ili bit-a (binarno), ili za registre (2 bajta).

Tablica 3. Funkcijski kodovi

Tip funkcije		Ime funkcije	Funkcijski kod	
Pristup podacima	Pristup bit-ovima	Fizički diskretni ulazi	Čitanje statusa ulaza	2
		Unutarnji bitovi	Čitanje statusa bita	1
			Postavljanje jednog bita	5
			Postavljanje više bitova	15
	16-bitni pristup	Fizički ulazni registri	Čitanje ulaznog registra	4
			Čitanje <i> Holding </i> registra	3
		Unutarnji registri	Postavljanje jednog registra	6
			Postavljanje više registara	16
			Čitanje/postavljanje registara	23
			Maskiranje registra	22
			Čitaj FIFO	24
	Pristup zapisima	Čitanje zapisa	20	
		Postavljanje zapisa	21	
	Dijagnostika	Čitanje statusa iznimaka	7	
Dijagnostika		8		
Čitanje broja COM događaja		11		
Čitanje COM događaja		12		
Čitanje adrese Slave-a		17		
Čitanje identifikatora uređaja		43		
Ostalo	Enkapsulacija	43		

Modbus RTU [Tablica 4] i ASCII [Tablica 5] oboje se koriste kod serijske komunikacije, odnosno kod RS-323 i RS-485 mreža. Poruka koja se prenosi u ASCII obliku lako je razumljiva ljudima, ali pošto je za prijenos iste poruke potrebno više bajtova, prednost ima Modbus RTU koji izravno prenosi heksadecimalne znamenke bez njihove konverzije u ASCII kod.

Tablica 4. Modbus RTU poruka

Naziv	Dužina (bitova)	Funkcija
Start	28	Početak poruke
Adresa	8	Adresa Slave uređaja
Funkcija	8	Želimo li čitati ili pisati
Podatak	n x 8	Informacija koja se šalje
CRC	16	Provjera
Kraj	28	Kraj poruke

Tablica 5. Modbus ASCII poruka

Naziv	Dužina (znakova)	Funkcija
Start	1	Početak poruke (:)
Adresa	2	Adresa Slave uređaja
Funkcija	2	Funkcijski kod
Podatak	n	Informacija koja se šalje
CRC	2	Provjera
Kraj	2	Kraj poruke (CR/LF)

Oblik zaglavlja Modbus TCP-a [Tablica 6] je dulji nego kod Modbus RTU i ASCII, ali pošto se bazira na Ethernet mrežama koje su puno brže od serijskih mreža omogućuje prijenos podataka puno većom brzinom. Jedina mana je što je potrebno dovesti poseban kabel od razdjelnika do svakog uređaja, za razliku od Modbus RTU i ASCII kod kojih su svi uređaji povezani sa dvije upletene žice.

Tablica 6. Modbus TCP poruka

Naziv	Dužina (bajtova)	Funkcija
Identifikator transakcije	2	Sinkronizacija server-klijent
Identifikator protokola	2	Nula za Modbus TCP
Dužina okvira	2	Broj preostalih bajtova
Adresa	1	Adresa Slave uređaja
Funkcija	1	Funkcijski kod
Podatak	n	Informacija koja se šalje

3.2.4. Wi-Fi

Wi-Fi (eng. *Wireless - Fidelity*) ili IEEE 802.11 je bežična mreža gdje se podaci između dva ili više uređaja prenose pomoću radijskih frekvencija i odgovarajućih antena. Najčešće se koristi u LAN mrežama, a brand je Wi-Fi Alianse-a [Slika 5] koji propisuje standarde i izdaje certifikate za sve Wi-Fi uređaje. Wi-Fi mreže rade pomoću jednostavne radio tehnologije gdje se signali pretvaraju u nule i jedinice, a šalju se na frekvencijama 2.4 GHz i 5 GHz.



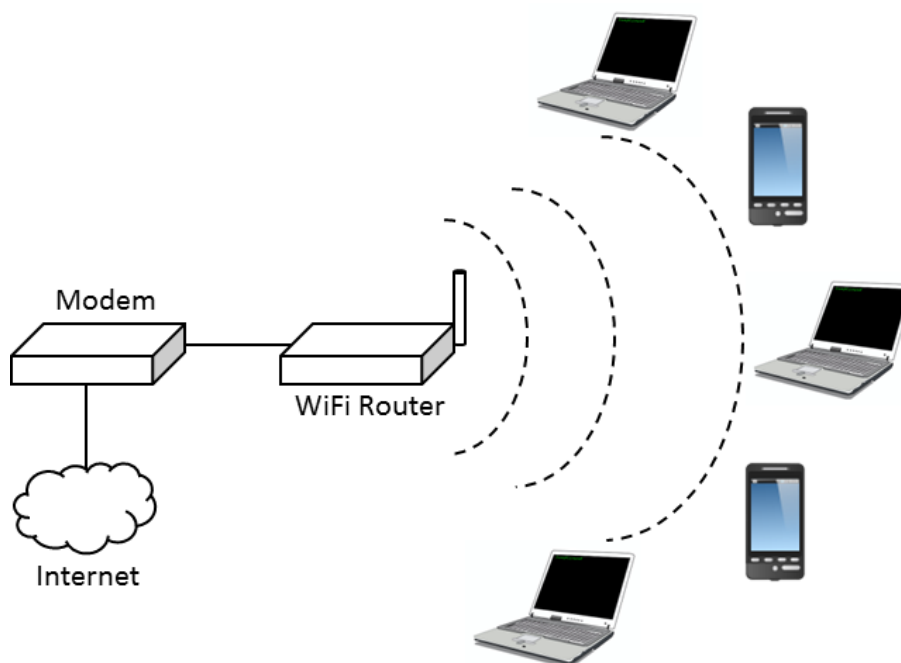
Slika 5. Wi-Fi logo korišten od Wi-Fi Alianse-a

Postoji više Wi-Fi standarda [Tablica 7] koji su se razvijali s godinama, te omogućavali veći domet i brzine. Razlog popularnosti bežičnih mreža je mobilnost i komocija koju pružaju pošto nema žica, te je domet dovoljan, a cijena za postavljanje takve mreže niska. Loša strana je što se podaci prenose radiovalovima, koje je lako moguće prislušivati. Zbog toga su razvijeni različiti tipovi sigurnosnih zaštita, od osnovnog WEP (eng. *Wireless Equivalent Privacy*) do naprednijih WPA (eng. *Wi-Fi Protected Access*) i WPA2.

Tablica 7. Wi-Fi standardi

Standard	Frekvencija	Maksimalna brzina	Domet
802.11	2.4 GHz	2 Mbit/s	20 m
802.11b	2.4 GHz	11 Mbit/s	35 m
802.11a	5 GHz	54 Mbit/s	35 m
802.11g	2.4 GHz	54 Mbit/s	38 m
802.11n	2.4 GHz	72 Mbit/s	70 m
	5 GHz	150 Mbit/s	70 m
802.11ac	5 GHz	1300 Mbit/s	35 m
802.11ad	5 GHz	7 Gbit/s	60 m

Svi uređaji koji se spajaju na bežičnu mrežu trebaju imati bežičnu mrežnu karticu koja je danas integrirana u čip. Ti uređaji se dijele na pristupne točke i klijente. Pristupne točke su osnova koja omogućava spajanje klijenata, komunikaciju između klijenata, upravljanje tokom podataka, te spajanje na druge mreže žičano. Klijenti su računala, mobilni telefoni, WLAN kamere i ostali uređaji koji imaju ugrađenu bežičnu mrežnu karticu te se spajaju na pristupne točke.



Slika 6. Primjer bežične mreže

3.3. Model i električna shema uređaja

Kako bi se vizualizirao cilj ovog projekta, napravljen je CAD model konačnog uređaja [Slika 7] u Solidworks-u. Iako su sva testiranja napravljena na Raspberry Pi Model B prve generacije, za konačni uređaj koji bi išao u proizvodnju odabran je Raspberry Pi Model B druge generacije. Najnoviji Raspberry Pi ima 6 puta brži procesor, 4 puta više radne memorije od testnog, ima provrte za montažu, više USB konektora, te je programski kompatibilan sa starijim generacijama. Uzeto je u obzir da je Raspberry Pi 2 fizički jednak Raspberry Pi Model B+, koji će se moći ugraditi u nemogućnosti nabavke zadnje verzije.

Kako su u industriji svi uređaji smješteni u ormare, odlučeno je odabrati standardno kućište koje ima mogućnost montaže na metalnu tračnicu DIN standarda dimenzija 35 x 7,5 kakva se često koristi za montažu elemenata u ormare. Uzeto je polikarbonatno kućište proizvođača CambenBoss model CNMB/6 dužine 106 mm kako bi u njega stao Raspberry Pi i ostala elektronika. Na tom kućištu bi trebalo jedino izraditi otvore za Ethernet i USB konektore.



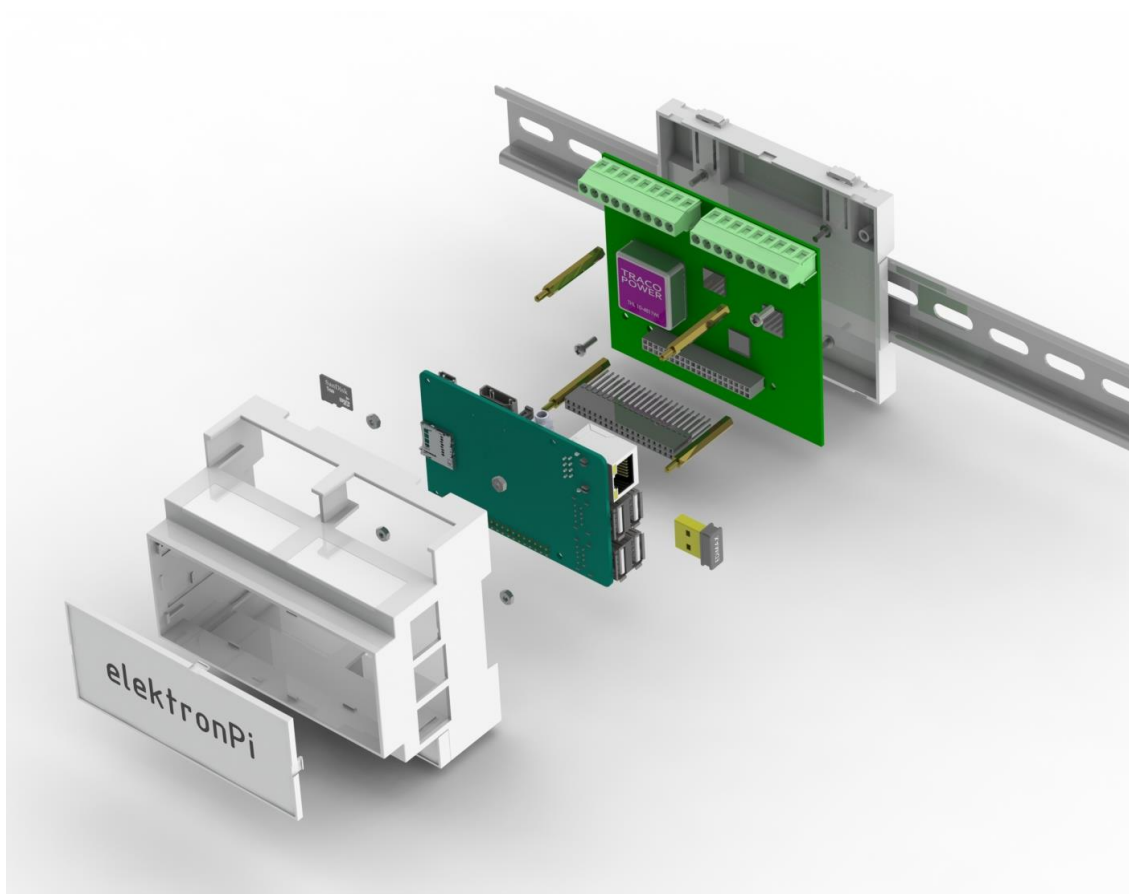
Slika 7. Uređaj za nadzor i upravljanje

Za bežičnu mrežu odabran je USB minijaturni uređaj Edimax EW-7811Un [Slika 8] težine svega 11 g. On omogućava rad kao Wi-Fi klijent, ali i kao pristupna točka što će se ovdje koristiti. Podržava standarde 802.11b/g/n te brzine do 150 Mbit/s sa dometom do 15 m i malom potrošnjom energije. Zbog malog dometa postoji mogućnost stavljanja nekog drugog uređaja za bežičnu mrežu, ali ako postoji već bežična mreža unutar lokalne mreže, dovoljno je uređaj spojiti na lokalnu mrežu preko Ethernet konektora, te će biti dostupan preko IP adrese.



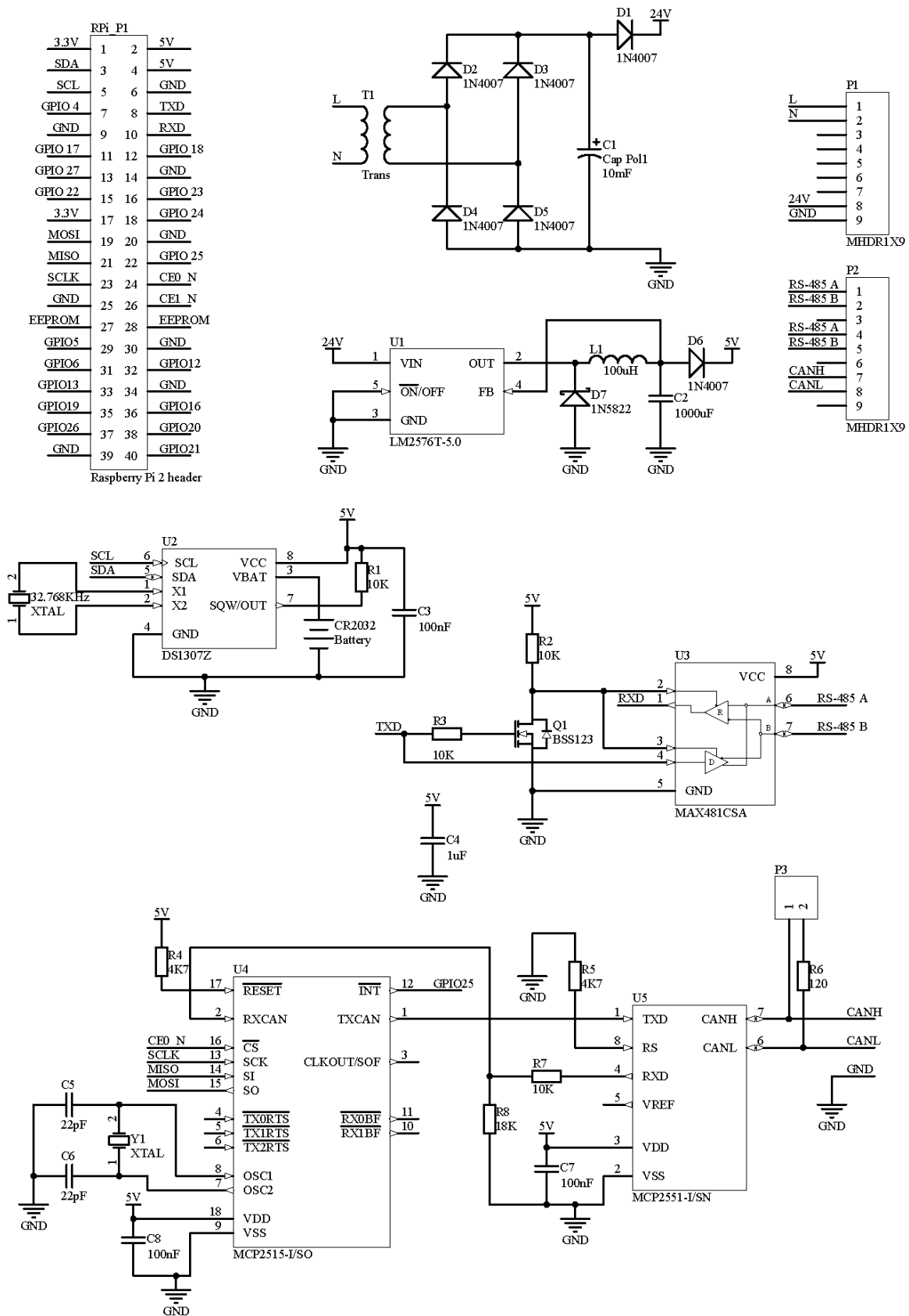
Slika 8. Edimax EW-7811Un

U industriji standardni napon koji se pojavljuje za napajanje je 24 – 48 VDC, te 230 VAC, dok je Raspberry Pi-u potrebno stabilnih 5V. Kako bi se osigurao napon napravljena je dodatna tiskana pločica koja bi sadržavala sklop za modulaciju napona preko visokih frekvencija. Za lako spajanje na uređaj na toj pločici nalaze se klemne za napajanje, te za spajanje na RS-485 mrežu, te su neke rezervirane za kasniju implementaciju tehnologije CANopen. Pošto Raspberry Pi nema ugrađen sat, informaciju o točnom vremenu prikuplja sa Internet servera i po gubitku napajanja gubi informaciju o vremenu. Ako nema pristup internetu Raspberry Pi će raditi s krivim vremenom i spremati sve podatke s krivim vremenom, zato bi se na toj pločici nalazio i sklop za stvarno vrijeme RTC (eng. *real-time clock*) koji sadrži bateriju i mikrokontroler, te može dugo pamtit i točno vrijeme.



Slika 9. Uređaj za nadzor i upravljanje – dijelovi

Raspberry Pi sadrži 40 pinski konektor koji omogućava različita proširenja. Od tih pinova neki su namijenjeni za napajanje od 3.3 V i 5V te masu, dok ostalih 26 pinova mogu poslužiti kao opći ulazi ili izlazi. Neki od tih pinova imaju posebne funkcije kao komunikacija SPI, I2C, te UART. Preko tog konektora povezani su Raspberry Pi i dodatna pločica. Za komunikaciju RS-485 korišten je čip MAX481CSA koji omogućava pretvorbu naponskog nivoa serijske komunikacije UART na naponski novi komunikacije RS-485, te komunikaciju preko dvije žice. Za RTC korišten je čip DS1307 sa kristalom frekvencije 32.768 kHz i modulacijom naponskih nivoa I2C komunikacije na koju je spojen. Kako bi se kasnije uređaj mogao proširiti i na CANopen komunikaciju, korišten je CAN kontroler MCP2515 spojen na SPI pinove, te MCP2551 koji modulira naponske nivoe CAN signala. Dodatna pločica može se proširiti još sa relejima koji bi mogli uključivati neke sigurnosne alarme ili zaštite. Ovo je samo prijedlog konačnog uređaja prema sadašnjem stanju testiranja određenih sustava, te su prije izrade konačne pločice moguća dodavanja nekih novih mogućnosti koje bi se pokazale korisnima.



Slika 10. Električna shema dodatne pločice

4. WEB APLIKACIJA ZA NADZOR I UPRAVLJANJE

Web aplikacije su programska rješenja kojima se pristupa putem web preglednika, dok se sama aplikacija nalazi na web poslužitelju, odnosno serveru. Za korištenje web aplikacije nije potrebno ništa instalirati niti skidati, već samo znati web adresu odnosno IP adresu računala na kojem se aplikacija nalazi.

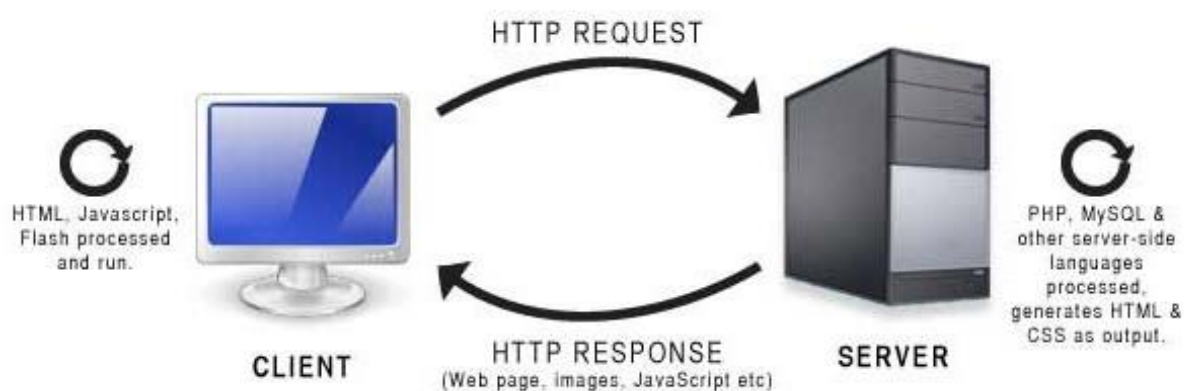
Klasična aplikacija mora biti pisana u nativnom jeziku za pojedinu platformu, te biti optimizirana za tu platformu. Web aplikacija napisana u nekom od serverskih jezika, npr. PHP, Python ili Ruby on Rails, može raditi na bilo kojoj server platformi, Linux ili Windows, bez puno optimizacije, a web aplikaciji može se pristupiti sa bilo kojeg uređaja koji ima web preglednik, bez obzira na operativni sustav, npr. Android, iOS, Windows, Linux ili MacOS. Web aplikacija mora biti jednostavna, moderna i vizualno atraktivna, koristeći najnovije tehnologije. Brzina rada web aplikacije ovisi o brzini mrežne povezanosti sa serverom na kojem se nalazi aplikacija. U ovom poglavlju opisati će se odnos server – klijent, koji su sve alati potrebni za izradu moderne web aplikacije, te funkcije i struktura izrađene web aplikacije, sučelje i skripta za prikupljanje podataka sa spojenih uređaja.



Slika 11. Razlika između klasične i web aplikacije

4.1. Internet

Internet je globalna računalna mreža koja povezuje računala i računalne mreže korištenjem internet protokola (IP) u jednu cjelinu s namjerom međusobnog razmjenjivanja informacija, odnosno pristup različitim sadržajima, servisima i uslugama kao što su elektronička pošta, prijenos datoteka, pregledavanje web-a i druge. Internet je baziran na modelu server – klijent, te svako računalo može biti server ili klijent ili oboje. Većina današnjih uređaja s pristupom na globalnu mrežu mogu biti klijenti, ali za prikaz informacija korisniku potreban je ekran i web preglednik. Takvih uređaja ima sve više oko nas, npr. mobitel, tablet, laptop, televizor, te omogućuju pregled web stranica i multimedijalnih sadržaja. Da bi računalo bilo server mora imati instaliran određeni serverski programski paket. Najprostraniji i najpoznatiji serverski programski paket je LAMP (eng. *Linux, Apache, MySQL, and PHP*) koji obuhvaća Linux operativni sustav, Apache HTTP server, MySQL koji omogućava rad s bazom podataka, te PHP skriptni jezik. Na serveru se nalaze web stranice kojima klijenti pristupaju preko porta 80, te poznavajući statičku IP adresu servera.



Slika 12. Odnos server – klijent

Web stranicama pristupa se preko HTTP (eng. *HyperText Transfer Protocol*) protokola [Slika 12] koji je baziran na zahtjev – odgovor modelu između klijenta i servera. Nakon što klijent uspostavi vezu sa udaljenim web serverom preko određenog porta, inicijalizira se prijenos podataka. Server konstantno osluškuje zahtjeve, čekajući da klijent pošalje niz znakova, kao npr. „GET / HTTP/1.1“ kojim se zahtijeva uspostavljanje komunikacije, a nakon toga i poruku koja sadrži nekoliko slovnih nizova (zaglavlje) koji određuje aspekte zahtjeva, te paket neobaveznih podataka.

Zahtjev klijenta će rezultirati slanjem odgovora sa servera, kao npr. „200 OK“ nakon čega će server poslati i svoj paket podataka koji najčešće sadrži traženu datoteku ili poruku o grešci. Odmah po ispunjenju zahtjeva klijenta, server prekida komunikaciju, po čemu se HTTP razlikuje od ostalih TCP (eng. *Transmission Control Protocol*) protokola kao FTP (eng. *File Transfer Protocol*), pri čemu je HTTP idealan za web, pošto su web stranice obično povezane s drugim stranicama na drugim serverima.

4.2. Web alati

U posljednjih 25 godina koliko postoji internet, isti je prošao kroz niz drastičnih promjena, vođen smjernicama i preporukama W3C-a (eng. *World Wide Web Consortium*) koji provodi standardizaciju izrade web stranica. Fleksibilne granice rezultirale su ogromnim brojem alata za izradu web stranica, te protokola koji se mijenjaju iz godine u godinu. Veći dio interneta danas je napravljen pomoću besplatnih alata otvorenog koda, a pošto su izvršne datoteke obične tekstualne datoteke sa različitim ekstenzijama, moguće ih je pisati u bilo kojem besplatnom tekstualnom editoru.



Slika 13. Usporedba HTML-a, CSS-a i Javascript-a

Kako bi web stranica bila moderna i interaktivna, potrebno je savladati HTML, CSS, Javascript, jQuery, PHP, MySQL i druge [Slika 13]. HTML (eng. *HyperText Markup Language*) je prezentacijski jezik za izradu stranica, te se izvršava u pregledniku. On daje stranici strukturu, te definira sve elemente sa posebnim oznakama. CSS (eng. *Cascading Style Sheets*) je stilski jezik koji se koristi za opis prezentacije dokumenta napisanog pomoću HTML jezika. JavaScript je skriptni jezik koji se izvršava u web pregledniku te stranici daje određenu dinamiku i interaktivnost. Najpoznatija JavaScript biblioteka je jQuery koja omogućuje lakši odabir nekih elemenata stranice, te sadrži različite funkcije za lakše i brže programiranje. JavaScript može u pozadini komunicirati sa serverom preko AJAX-a (eng. *Asynchronous JavaScript and XML*), te mijenjati sadržaj web stranice u pregledniku bez osvježavanja stranice [Slika 16]. Kako bi se lagano iščitali podaci koji se razmjenjuju između servera i web preglednika postoji tekstualni standard JSON (eng. *JavaScript Object Notation*).

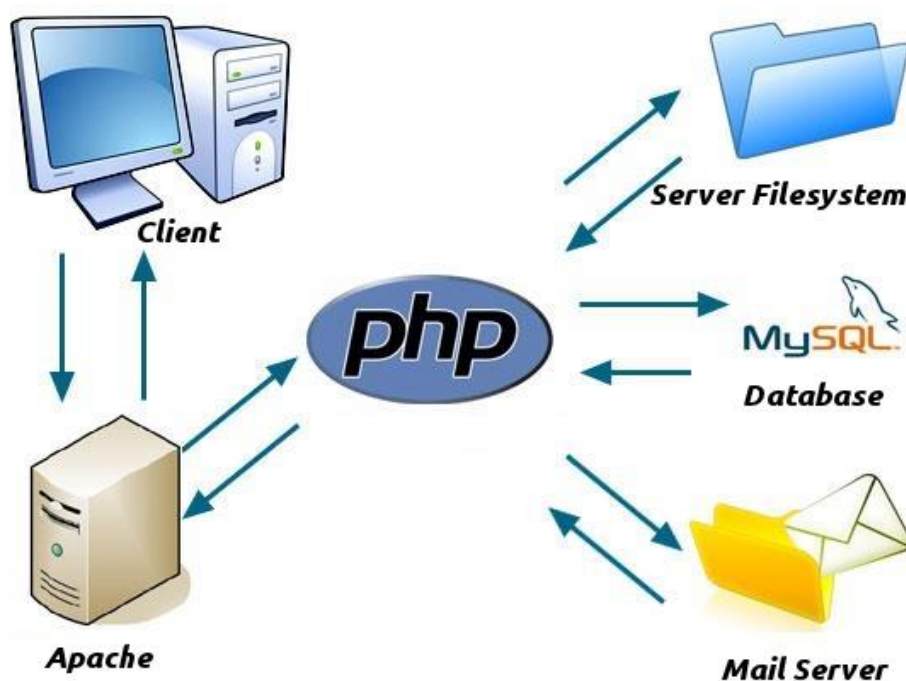
Kako se danas sve više pristupa web stranicama preko mobilnih uređaja zahtjeva se od stranice da bude prilagodljiva i čitljiva na uređajima različite veličine ekrana [Slika 14]. To se postiže korištenjem nekog od radnih okruženja, među kojima je najpopularniji Bootstrap, koji ima unaprijed definirane neke elemente.



Slika 14. Prilagodljiva web stranica veličini ekrana uređaja

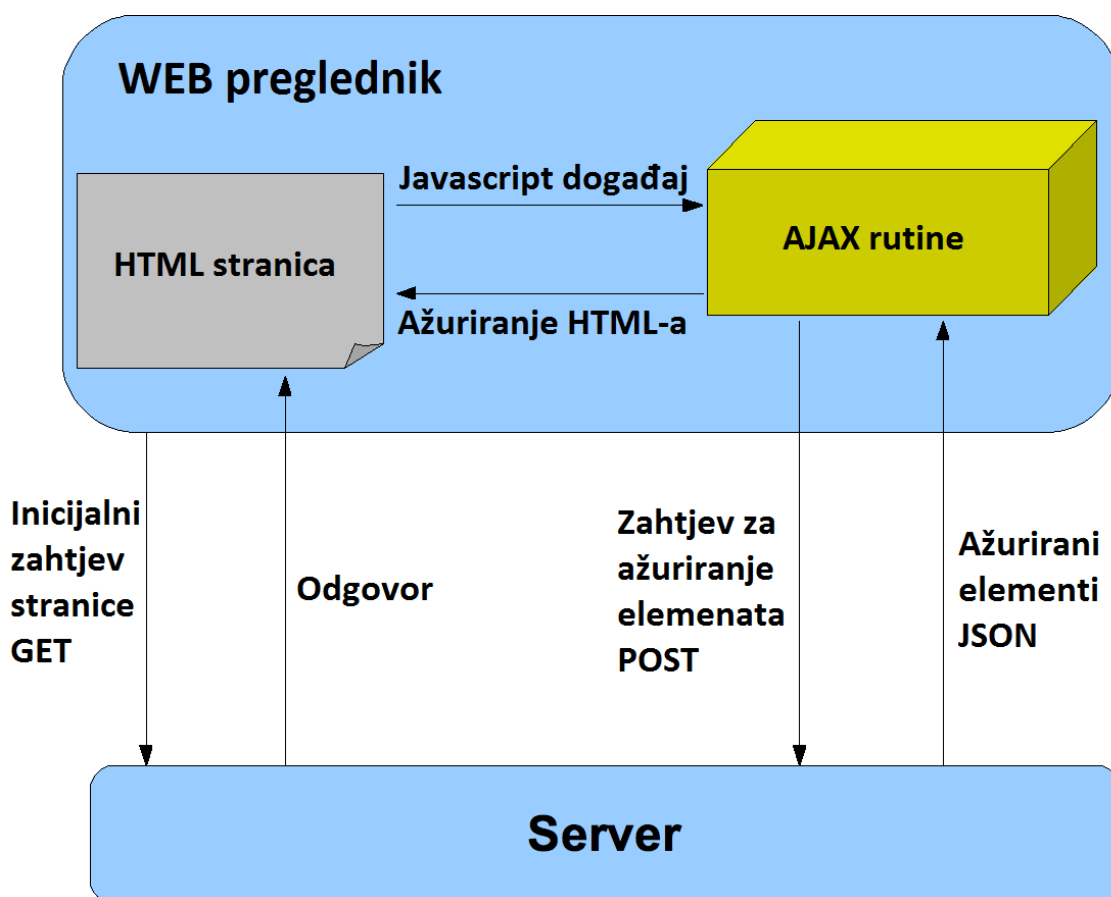
Na serverskoj strani potreban je neki programski jezik [Slika 15] koji će obraditi zahtjeve, čitati i pisati u baze podataka, pokrenuti neku skriptu, te generirati određeni HTML dokument koji će se prikazati korisniku u web pregledniku. Najrašireniji takav jezik je PHP (eng. *Hypertext preprocessor*) koji je skriptni jezik i kod koji je napisan u PHP-u nije dostupan korisniku. Kako se posljednjih godina napuštaju web aplikacije bazirane na Javi i Flash-u, tako se razvija sve više radnih okruženja baziranih na PHP-u, od kojih su najpopularniji Laravel, CakePHP, Symfony i drugi. Svi oni olakšavaju razvijanje web aplikacija, pošto već imaju ugrađene različite funkcije, te omogućuju preglednije pisanje koda korištenjem MVC (eng. *Model-view-controller*) dizajn uzorka. Tako se odvaja logika koja se izvršava na serveru, od dijela koji generira HTML dokument za korisnika, a sve to kontrolira posebna skripta. Ovom metodom se ne pozivaju određene skripte direktno, nego se u web adresi pozivaju određene kontroler skripte i metode koje se nalaze u tim skriptama kao funkcije, kojima se mogu proslijediti neki parametri.

Kako bi server mogao pametno spremati neke podatke, te upravljati njima, potreban je i neki sustav za upravljanje bazom podataka. Takav besplatni sustav otvorenog koda je i MySQL temeljen na SQL (eng. *Structured Query Language*) jeziku za izradu, traženje, ažuriranje i brisanje podataka iz relacijskih baza podataka.



Slika 15. Način rada servera sa PHP-om

Komunikacija između servera i web preglednika [Slika 16] odvija se na dva načina, POST i GET metodom. GET metodom se šalju podaci koristeći URL vezu, dok podaci poslani POST metodom su skriveni i nisu dostupni kroz adresu niti zabilježeni na računalu. POST metodom se najčešće šalju korisnički podaci i neki tajni podaci.

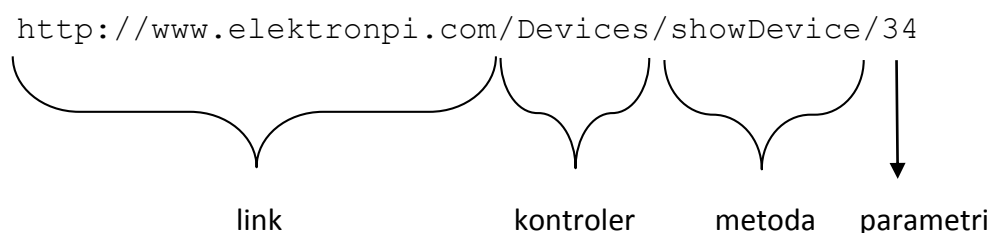


Slika 16. Ažuriranje sadržaja web stranice koristeći AJAX

4.3. Opis web aplikacije

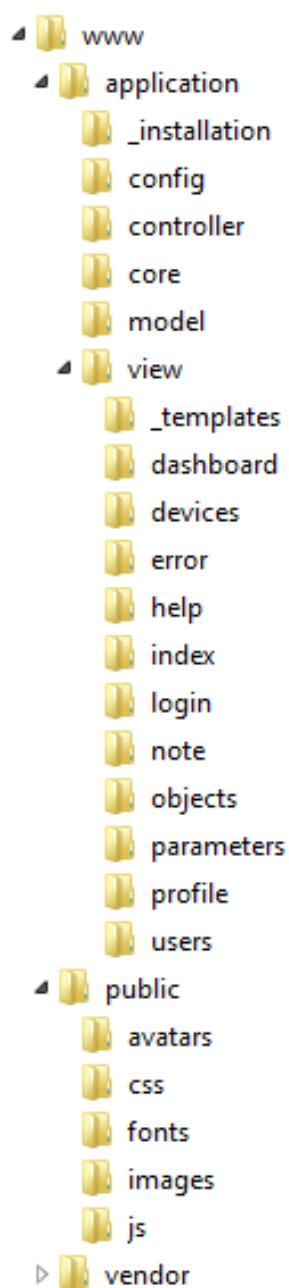
Web aplikacija razvijena za ovaj rad bazirana je na MVC dizajn uzorku korištenjem serverske razvojne okoline Huga, koja ima ugrađene funkcije za prijavljivanje korisnika, rad sa bazom podataka, detektiranje nekih grešaka i druge. Kako bi sučelje bilo prilagodljivo veličini ekrana implementirano je radno okruženje Bootstrap. U kasnijim verzijama web aplikacije trebalo bi se prebaciti na neku poznatiju serversku radnu okolinu kao Laravel zbog detaljnije dokumentacije, te nekih naprednijih funkcija koje ima implementirane.

Kod MVC dizajn uzorka u web adresi se ne pozivaju gotove statičke stranice, nego se poziva određeni kontroler koji sadrži funkciju, odnosno metodu [Slika 17]. Kontroler, metoda i parametri koji se šalju serveru odijeljeni su kosom crtom te se posebna skripta brine za njihovo odjeljivanje, kako bi se izvršila određena radnja, odnosno kako bi se obavijestio korisnik ako postoji bilo kakva pogreška ili ako željeni kontroler ili metoda ne postoje. Ovdje su stranice dinamičke te ih server generira ovisno o željenom sadržaju.



Slika 17. MVC - pozivanje funkcija

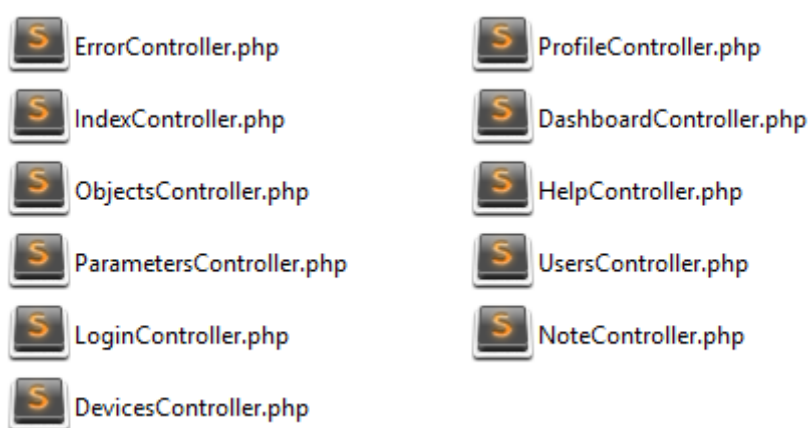
MVC dizajn web aplikacije je odličan izbor jer se odvaja prezentacijski dio od programa koji se izvršava na serveru, kao rad s bazom podataka, a s time upravlja kontroler. Kako bi kod bio čitljiviji svaka funkcija se nalazi u posebnoj datoteci. Ova razvojna okolina sastoji se od tri osnovna direktorija [Slika 18]. U direktoriju „public“ nalaze se sve javno dostupne datoteke, kao Javascript skripte, CSS datoteke za uređivanje, slike i fontovi, dok posebna datoteka `.htaccess` koja se nalazi u `www` direktoriju zabranjuje javni pristup direktorijima „application“ i „vendor“. U direktoriju „application“ nalazi se web aplikacija, a u direktoriju `vendor` se nalaze gotove web aplikacije koje omogućuju slanje mail-ova, generiranje captcha slika kod registracije i slično.



Slika 18. MVC - struktura direktorija

U direktoriju „_installation“ nalaze se datoteke za podešavanje MySQL baze podataka kod instalacije web aplikacije, dok direktorij „config“ sadrži datoteke za unos postavki kao mail podatke i korisničke podatke za bazu podataka. Direktorij „core“ sadrži datoteke, odnosno gotove funkcije potrebne za rad web aplikacije, razdvajanje web adrese, funkcije za autorizaciju, funkcije za pozivanje željenih kontrolera i metoda, funkcije za otkrivanje pogrešaka, funkcije za rad s bazom podataka, funkcije za mail, preusmjeravanje i ostale.

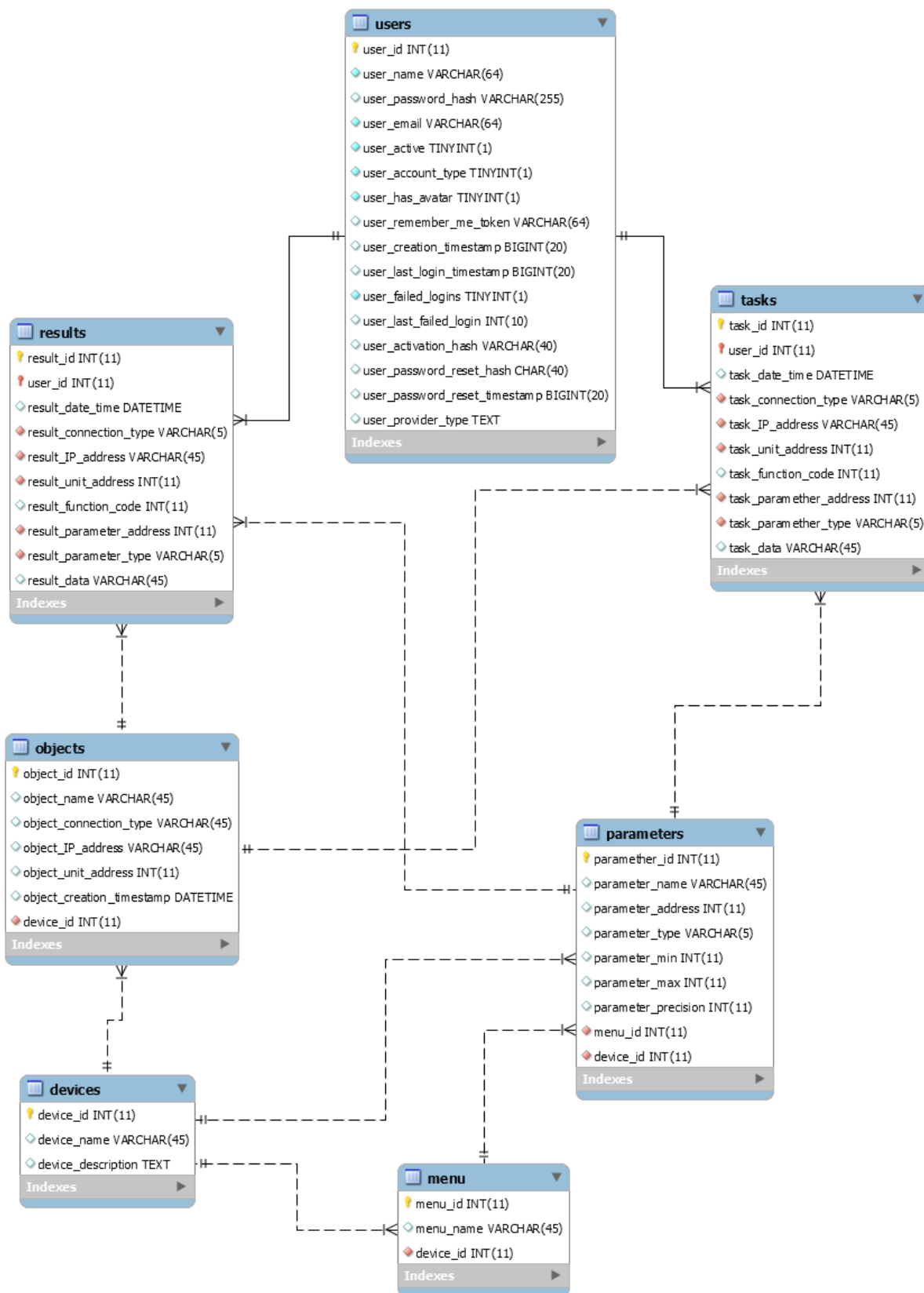
Svi kontroleri [Slika 19] nalaze se u direktoriju „controller“, sve skripte koje se izvršavaju na serveru nalaze se u direktoriju „model“, a HTML kod koji će biti prezentiran korisniku nalazi se u direktoriju „view“. Kako kod web aplikacije zaglavlje, navigacijski meni i podnožje ostaju nepromijenjeni, nego se samo mijenja sadržaj, zaglavlje i podnožje se nalaze u posebnoj direktoriju „_templates“ unutar view direktorija te se oni uvijek spajaju sa kodom za sadržaj. Kod za sadržaj podijeljen je u direktorije prema nazivu kontrolera koji će ga pozvati. Na taj način odvojen je prezentacijski dio od logičkog, te je lakše razvijati web aplikaciju.



Slika 19. Kontroleri u direktoriju controller

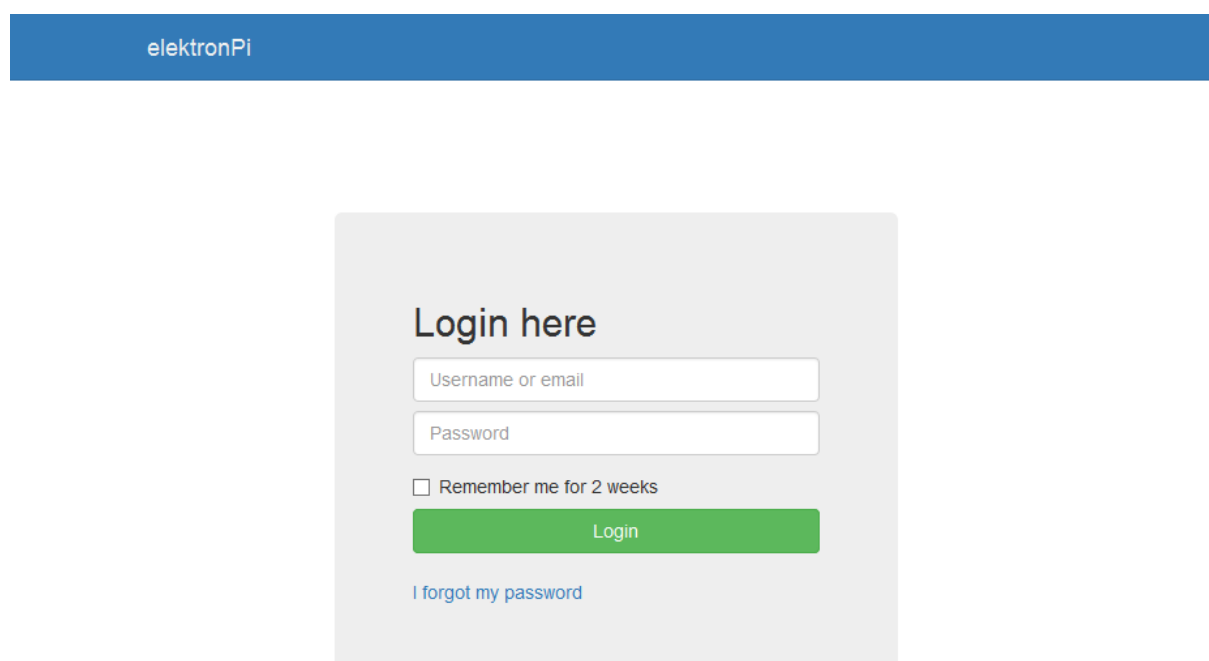
Sve informacije o uređajima nalaze se u MySQL bazi podataka [Slika 20]. U toj bazi nalazi se više tablica kako bi bio lakši pristup informacija te se smanjila redundancija. Svi regulatori i pokazivači koji su spojeni na uređaj za nadzor i upravljanje nalaze se u tablici „objects“ zajedno sa informacijama o vrsti komunikacije, te identifikacijskom oznakom vrste uređaja. Vrste uređaja nalaze se u tablici „devices“, te adrese njihovih parametara nalaze se u tablici „parameters“. U bazi podataka nalazi se i tablica korisnika sa njihovom razinom pristupa.

Pošto se PHP skripta ne može sama pokrenuti kod uključivanja i da radi u petlji, dio aplikacije za prikupljanje informacija sa mreže odvojen je u posebnu Python skriptu. To je učinjeno kako bi se kasnije mogao implementirati modul za logiranje parametara kada korisnik nije spojen na web aplikaciju. Zato se u bazi podataka nalaze tablica „tasks“ i „results“. Kada korisnik zatraži vrijednost nekog parametra sa uređaja, u tablicu „tasks“ zapišu se informacije o uređaju, adresa parametra, što se želi, te kada skripta prikupi podatke rješenje zapiše u tablicu „results“.



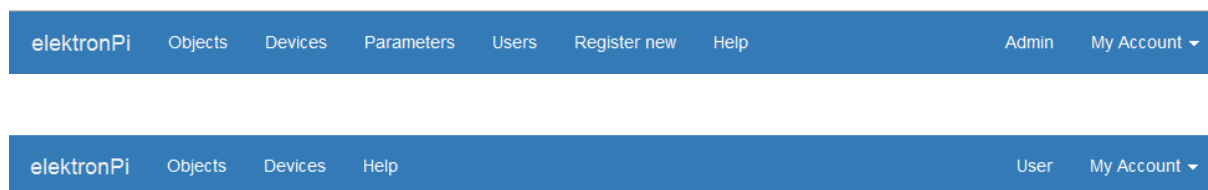
Slika 20. Relacije između tablica u MySQL bazi podataka

Po pozivanju web adrese aplikacije poziva se stranica za prijavljivanje korisnika [Slika 21]. Svoje podatke korisnik dobiva od administratora, te poslije može mijenjati lozinku. Korisnik se prijavljuje pomoću svog korisničkog imena ili email-a i lozinkom, a ako želi može odabrati mogućnost da aplikacija zapamti njegovu prijavu. Ako je korisnik zaboravio lozinku, preusmjerava se na stranicu gdje mora upisati svoju email adresu kako bi dobio link za ponovno generiranje nove lozinke. Ako se korisnik želi prijaviti sa nepostojećim korisničkim imenom aplikacija ga obavještava da ne postoji taj korisnik, a ako upiše krivu lozinku aplikacija napiše da je pogrešna lozinka. Ako se tri puta upiše pogrešna lozinka aplikacija zabranjuje na 30 sekundi prijavu tom korisniku, što je jedna od mjera protiv provaljivanja na aplikaciju.

The image shows a login form for 'elektronPi'. At the top, there is a blue header with the text 'elektronPi'. Below the header is a light gray box containing the login interface. The interface has the heading 'Login here' in bold. There are two input fields: 'Username or email' and 'Password'. Below these fields is a checkbox labeled 'Remember me for 2 weeks'. A green button labeled 'Login' is positioned below the checkbox. At the bottom of the form, there is a blue link that says 'I forgot my password'.

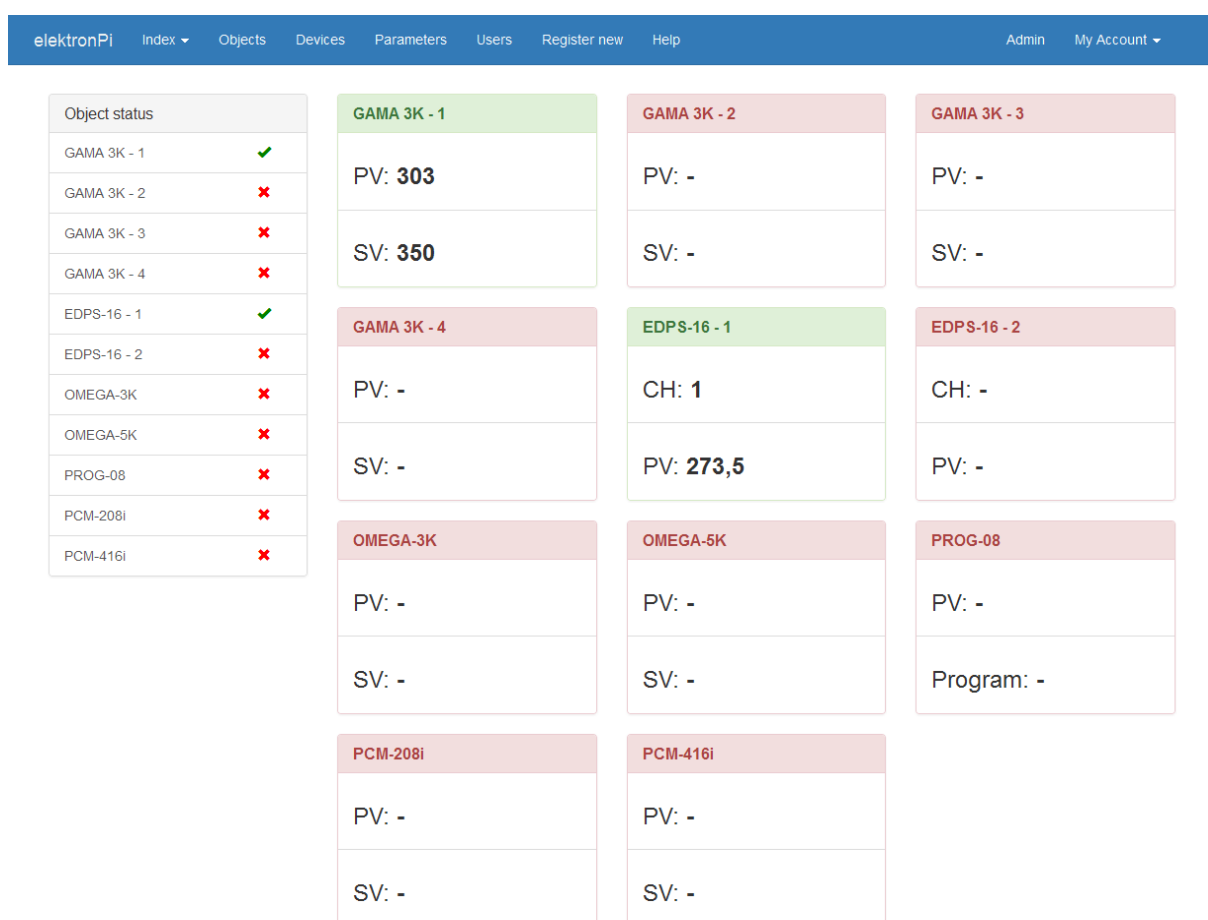
Slika 21. Prijava u web aplikaciju

Po uspješnom prijavljivanju pojavljuje se početna stranica [Slika 23] na kojoj se na vrhu nalazi navigacija, te u srednjem dijelu opći pregled stanja svih uređaja koji su konfigurirani, te njihove mjerne vrijednosti koje su inače prikazane na ekranu uređaja. Korisnik ne može vidjeti niti pristupiti dijelu navigacije koji služi za konfiguriranje uređaja, vrste uređaja i dodavanje novih korisnika, nego samo pregled spojenih uređaja, te upravljati s njima.



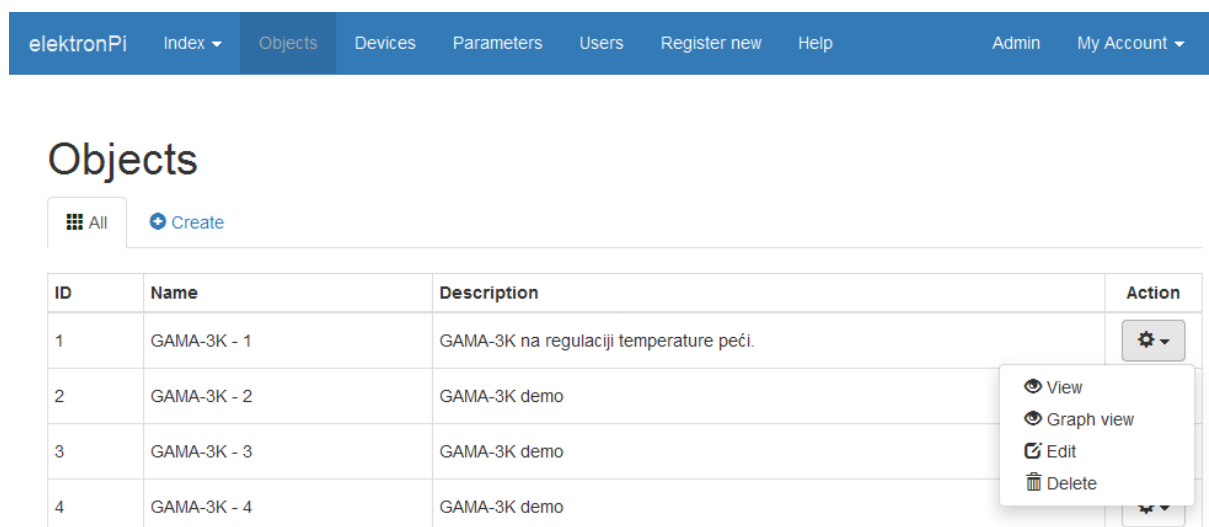
Slika 22. Usporedba navigacije administratora i korisnika

Hijerarhija je napravljena sa jednim administratorom koji može vidjeti sve na web aplikaciji, te za sada neograničenim brojem korisnika kojima je dostupan samo onaj dio aplikacije [Slika 22] za pregled, upravljanje i nadzor određenih uređaja.



Slika 23. Početna stranica web aplikacije

Postaviti novi uređaj može samo administrator. U izborniku „Objects“ [Slika 24] prikazan je popis svih konfiguriranih uređaja, te njihov opis. Padajući izbornik omogućuje prikaz svih parametara, prikaz parametara u grafu, te editiranje ili brisanje objekta, odnosno uređaja.



elektronPi Index ▾ Objects Devices Parameters Users Register new Help Admin My Account ▾

Objects

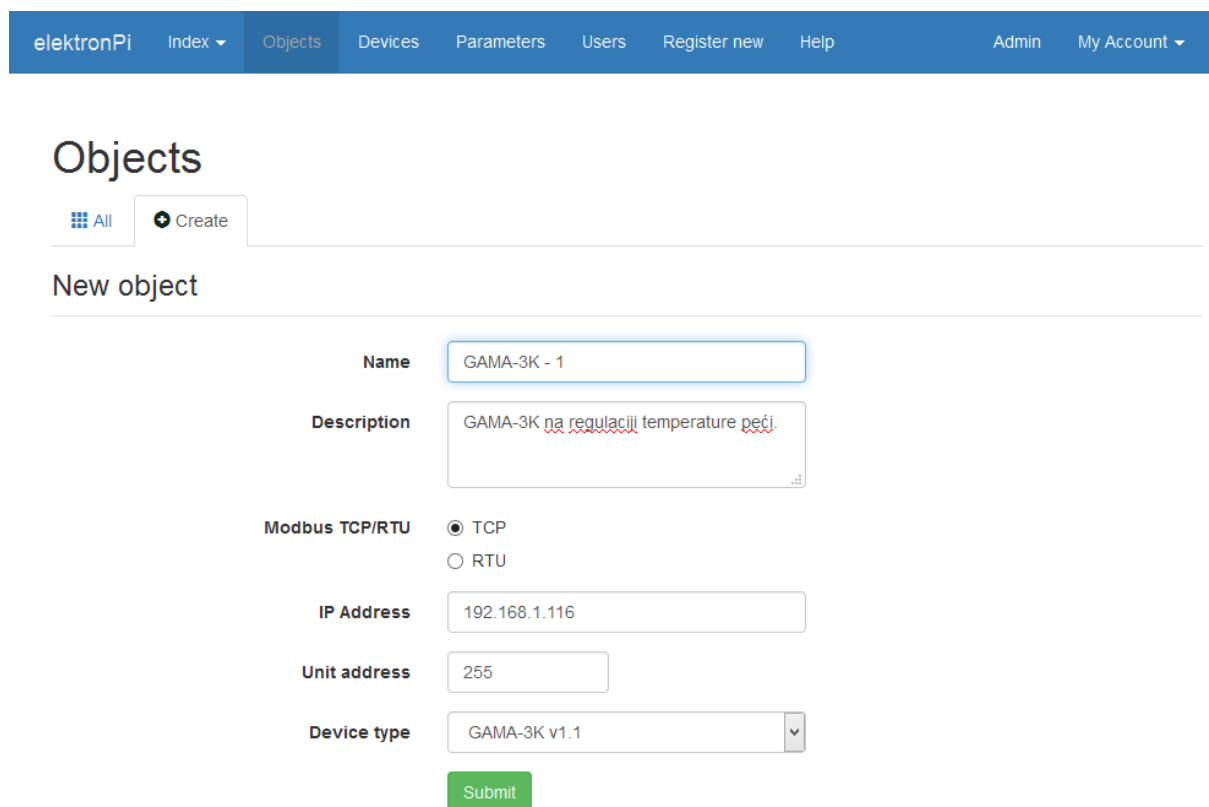
All Create

ID	Name	Description	Action
1	GAMA-3K - 1	GAMA-3K na regulaciji temperature peći.	⚙️
2	GAMA-3K - 2	GAMA-3K demo	
3	GAMA-3K - 3	GAMA-3K demo	
4	GAMA-3K - 4	GAMA-3K demo	

- View
- Graph view
- Edit
- Delete

Slika 24. Popis svih konfiguriranih uređaja

Za konfiguraciju novog uređaja potrebno je odabrati „New object“, pri čemu se preusmjerava na stranicu za postavljanje novog uređaja [Slika 25]. Potrebno je upisati ime uređaja prema lokaciji ili nekoj drugoj specifičnosti, odabrati vrstu komunikacije (TCP ili RTU), te podesiti IP adresu, adresu uređaja, i iz padajućeg izbornika vrstu uređaja koje je unaprijed konfigurirao proizvođač.



elektronPi Index ▾ Objects Devices Parameters Users Register new Help Admin My Account ▾

Objects

All Create

New object

Name

Description

Modbus TCP/RTU TCP RTU

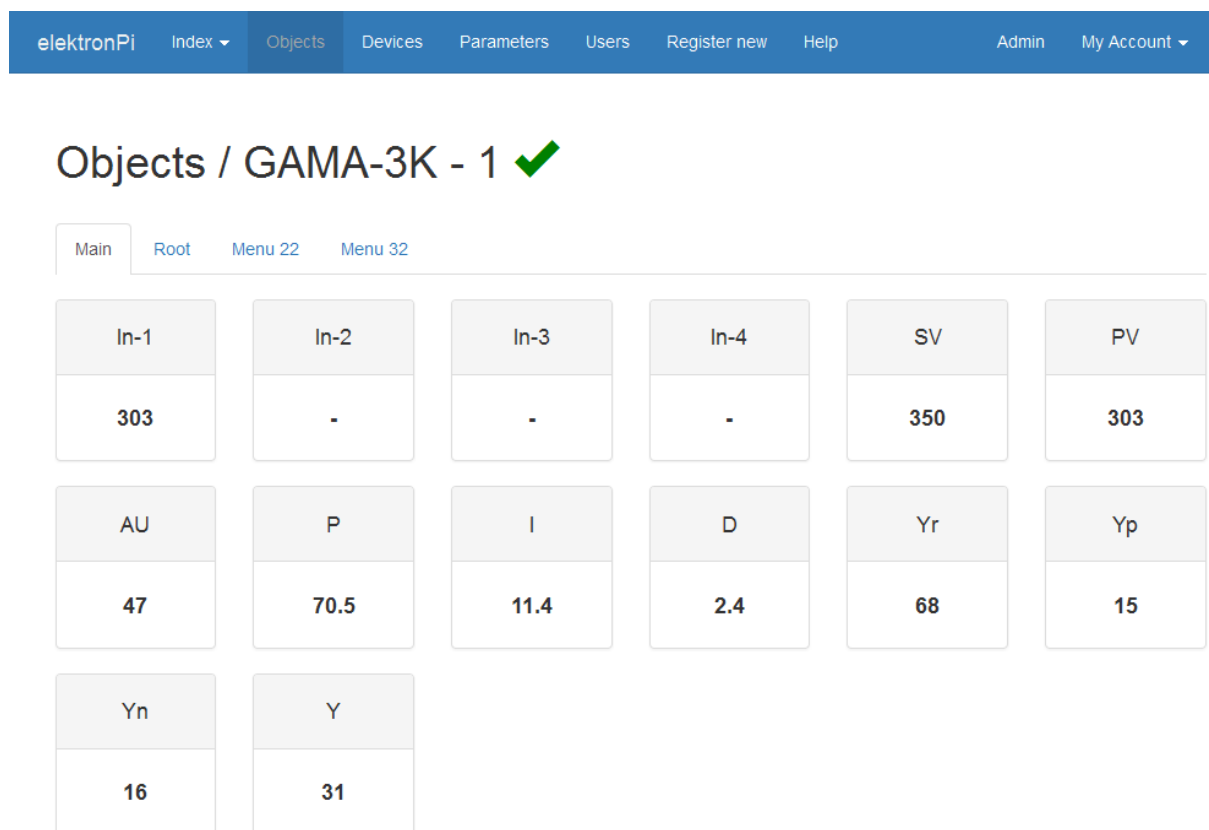
IP Address

Unit address

Device type

Slika 25. Postavljanje novog objekta

Odabirom željenog uređaja iz tablice uređaja prikazani su svi parametri i njihove vrijednosti razvrstani prema izborniku uređaja [Slika 26]. Ako su uređaji uključeni i dostupni automatski će se osvježavati vrijednosti. U kasnijoj verziji web aplikacije potrebno je dodati funkciju promjene parametara.

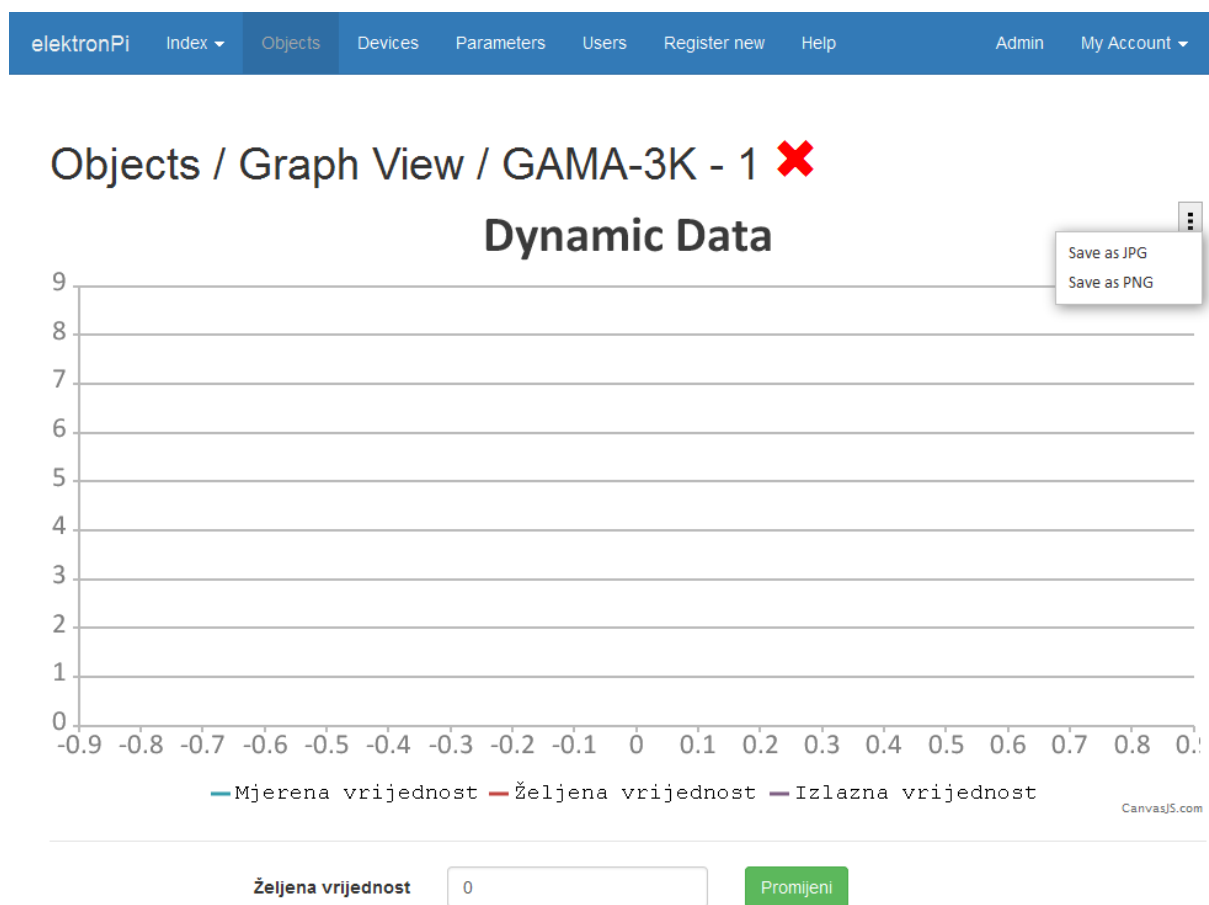


In-1	In-2	In-3	In-4	SV	PV
303	-	-	-	350	303
AU	P	I	D	Yr	Yp
47	70.5	11.4	2.4	68	15
Yn	Y				
16	31				

Slika 26. Pregled parametara jednog objekta

Za automatsko osvježavanje parametara zadužena je JavaScript skripta, koja pomoću AJAX biblioteke komunicira sa serverom i mijenja elemente HTML koda. Po pozivanju stranice skripta prvo provjerava status uređaja preko identifikacijskog broja, odnosno da li je uređaj dostupan preko mreže koja je predefiniрана. Ako je uređaj dostupan skripta šalje u JSON obliku identifikacijski broj uređaja, te identifikacijske brojeve svih parametara koji su vidljivi na ekranu, serveru zahtjev za osvježavanje. Zatim serverska skripta u tablicu „Tasks“ zapiše sve podatke za parametre koji se žele osvježiti. Drugi dio skripte šalje serveru zahtjev za čitanjem tablice rezultata. Ako se u tablici „Results“ nalaze podaci, server ih, isto u JSON obliku šalje JavaScript skripti na klijentu, te ta skripta izmjeni vrijednosti HTML elemenata, a iz tablice rezultata se ti rezultati zatim brišu.

Odabirom izbornika „Graph view“ [Slika 27] prikazuju se na grafu mjerene vrijednosti, željene vrijednosti i izlazne vrijednosti, a ispod grafa se može podesiti željena vrijednost. Za prikaz vrijednosti u grafu korištena je gotova biblioteka CanvasJS, koja omogućuje uvećanje grafa, te spremanje grafa kao slike. Graf se osvježuje svake 2 s, dok bi se u kasnijim verzijama nakon testiranja brzine osvježavanja podataka, omogućilo korisniku da odabere brzinu osvježavanja.



Slika 27. Pregled parametara objekta u grafu

Novog korisnika može registrirati samo administrator [Slika 28] dodjeljujući mu korisničko ime, lozinku, te unosom email adrese korisnika kako bi kasnije korisnik mogao potvrditi registraciju i promijeniti lozinku. Novom korisniku se automatski dodjeljuje niži status autorizacije te ne može vidjeti i pristupiti cijeloj aplikaciji. Administrator može u tablici „Users“ vidjeti sve registrirane korisnike, kada su zadnji put pristupili, te ako je potrebno neke i obrisati.

elektronPi Index ▾ Objects Devices Parameters Users Help My Account ▾


Register a new account

Username (letters/numbers, 2-64 chars)

email address (a real address)

Password (6+ characters)

Repeat your password



Please enter above characters

[Reload Captcha](#)

[Register](#)

Please note: This captcha will be generated when the img tag requests the captcha-generation (= a real image) from YOURURL/login/showcaptcha. As this is a client-side triggered request, a \$_SESSION["captcha"] dump will not show the captcha characters. The captcha generation happens AFTER the request that generates THIS page has been finished.

ELEKTRON ERMA-STRMEC

Strmec 182, Stubičke toplice

marko.les@zoho.com

Slika 28. Obrazac za registraciju novog korisnika

Kako je ova web aplikacija jako opširna, nisu implementirani svi željeni moduli. U izborniku „Help“ trebalo bi ubaciti još upute za korištenje web aplikacije, te upute za svaki uređaj kako se postavlja i podešava. Ta uputstva bi se trebala ažurirati kada je uređaj spojen na internet sa servera tvrtke Elektron Erma – Strmec. Korisniku bi trebalo još omogućiti postavljanje svojih mrežnih postavki. Za to treba u izborniku „Settings“ dodati skriptu koja bi editirala mrežne postavke.

4.4. Skripta za prikupljanje podataka

Pošto PHP skripta ne može raditi u petlji i biti stalno aktivna, nego tek na poziv nekog programa odnosno korisnika, odlučeno je odvojiti dio programa za prikupljanje podataka sa ostalih uređaja. Raspberry Pi je orijentiran oko Python jezika, koji je već instaliran i lako ga je pokrenuti. Python skripta koja radi u petlji čita iz tablice zadataka parametre, te nakon uspješno izvršenih zadataka, rješenja sprema u tablicu rezultata u bazi podataka. Kako bi skripta imala pristup bazi podataka, te mogla s njom upravljati potrebno je instalirati MySQLdb biblioteku. Za prikupljanje podataka sa RS-485 i Ethernet mreže koristi se pymodbus biblioteka koja ima podršku za Modbus RTU, ASCII i TCP.

Kako bi se omogućilo automatsko pokretanje skripte kod uključivanja Raspberry Pi-a, potrebno je izmijeniti datoteku `/etc/profile`:

```
sudo nano /etc/profile
```

te na kraj datoteke dodati putanju do skripte:

```
sudo python /home/pi/pymodbus-elektronpi.py
```

i nakon toga spremi datoteku i ponovno pokrenuti Raspberry Pi upisujući naredbu:

```
sudo reboot.
```

Skripta se vrti u beskonačnoj petlji sa pauzom od 1 ms kako se ne bi zagušio rad procesora. Skripta se prvo spoji na bazu podataka sa korisničkim imenom i lozinkom, te uzme sve redove iz tablice zadataka. Za svaki redak prvo provjerava kojim protokolom se treba spojiti, te se zatim spoji na uređaj. Nakon uspješne konekcije provjerava što je potrebno izvršiti, čitati ili pisati te s koje adrese i to izvrši. Ako je dobila uspješan odgovor od uređaja, ponovno se spaja na bazu podataka, te sprema rezultat u tablicu rezultata i iz tablice zadataka briše taj zadatak. Ako konekcija na uređaj nije uspostavljena ili je došlo do neke pogreške ta informacija se zapisuje u tablicu rezultata.

5. POSTAVLJANJE WEB APLIKACIJE NA RASPBERRY PI

Kako bi se na Raspberry Pi postavila web aplikacija, potrebno je imati instaliran odgovarajući operativni sustav na memorijskoj kartici, programski paket koji omogućava računalu da radi kao web server, te ostale programe koji upravljaju bazom podataka. Pošto se koristi posebna skripta za prikupljanje podataka potrebno je instalirati određene biblioteke, te omogućiti skripti automatsko pokretanje kod uključivanja.

5.1. Instaliranje operativnog sustava

Prvo je potrebno na SD karticu minimalnog kapaciteta 2GB instalirati RASPBIAN operativni sustav. Sa stranice Raspberry Pi-a skinuti sliku RASPBIAN, te pomoću programa Win32 Disk Imager [Slika 29] instalirati na SD karticu. Po završetku procesa u Raspberry Pi umetnuti SD karticu, spojiti na mrežu, spojiti Wi-Fi stick, spojiti pomoću HDMI kabla pločicu na monitor i spojiti tipkovnicu. Na kraju se spaja napajanje na Raspberry Pi, nakon čega se odmah podiže operativni sustav. Ako se ne pojavi prozor Raspberry Pi Software Configuration Tool (raspi-config) [Slika 30] potrebno se ulogirati sa sljedećim podacima

host: raspberrypi

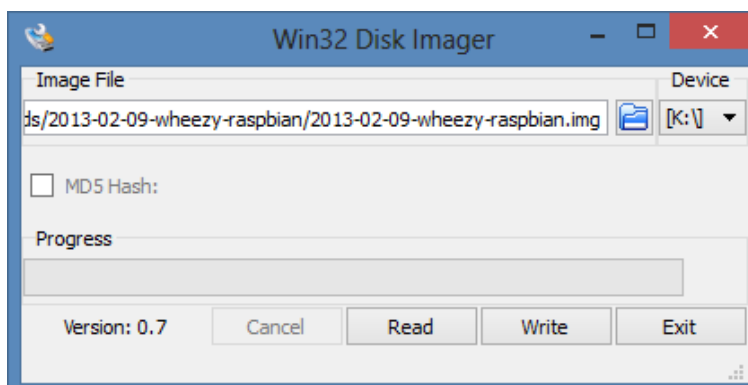
Username: pi

Password: raspberry

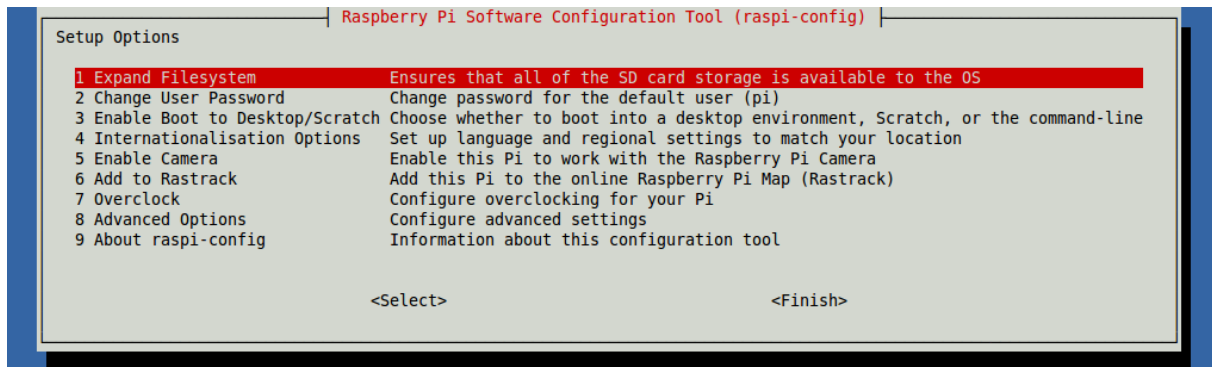
te za ulazak u meni za podešavanje upisati

```
sudo raspi-config
```

nakon čega se pojavljuje konfiguracijski meni.



Slika 29. Win32 Disk Imager za instalaciju operacijskog sustava Raspbian



Slika 30. Raspberry Pi konfiguracijski meni

Odabrati Expand Filesystem kako bi se memorija particije povećala na kapacitet cijele memorijske kartice, te u Advanced Options uključiti SSH kako bi se moglo preko mreže upravljati i podešavati uređaj. Nakon toga napraviti restart sa naredbom **sudo reboot**, te nakon ponovnog uključanja ulogirati se sa starim podacima. Zatim ažurirati operativni sustav sa sljedećim naredbama:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

Za automatsko ažuriranje operativnog sustava instalirati sljedeće:

```
sudo apt-get install ca-certificates
```

```
sudo apt-get install git-core
```

```
sudo wget https://raw.githubusercontent.com/Hexxeh/rpi-
```

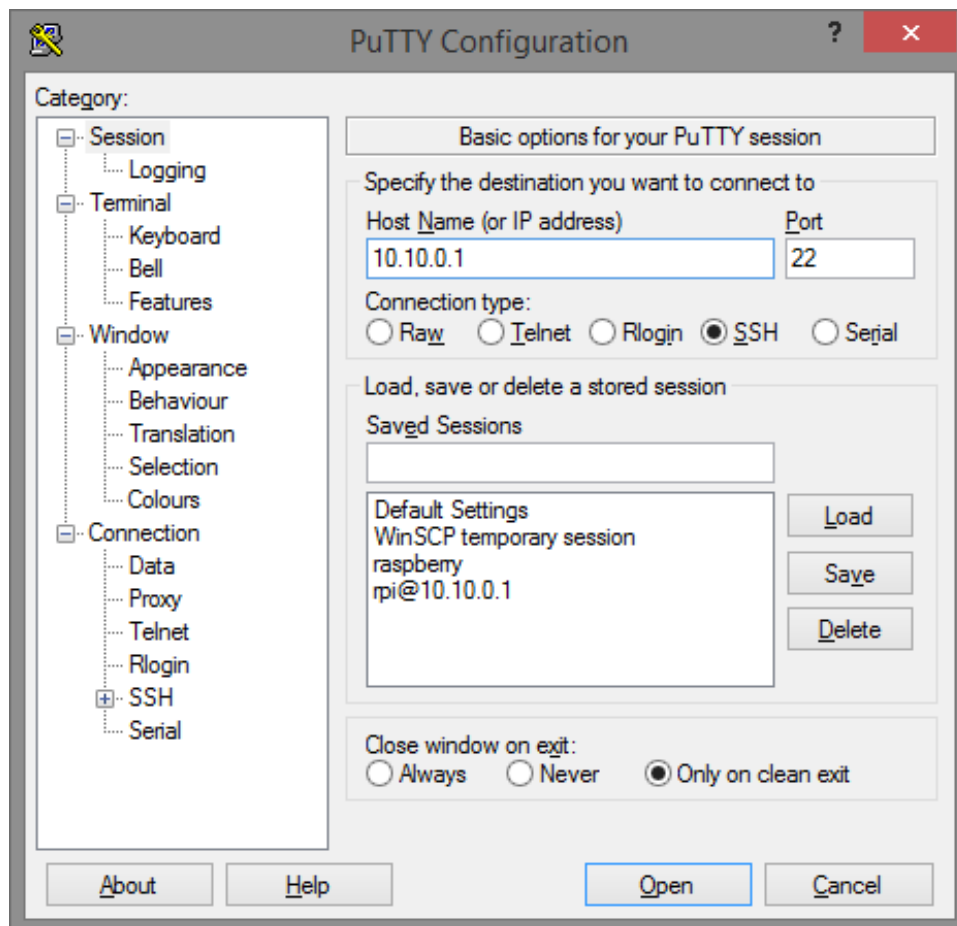
```
update/master/rpi-update -O /usr/bin/rpi-update && sudo
```

```
chmod +x /usr/bin/rpi-update
```

```
sudo rpi-update
```

```
sudo shutdown -r now
```

Sa naredbom **ifconfig** može se vidjeti IP adresu Raspberry Pi-a na mreži. Sada se može odspojiti monitor i tipkovnica, te pomoću programa Putty [Slika 31] spojiti na Raspberry Pi preko SSH protokola sa računala.



Slika 31. Putty –program za pristup uređaju preko SSH protokola

5.2. Podešavanje Wi-Fi pristupne točke

Za izradu Wi-Fi pristupne točke koristi se Edimax EW-7811Un. Za instaliranje drivera utipkati sljedeće naredbe:

```
wget http://www.daveconroy.com/wp3/wp-
content/uploads/2013/07/hostapd.zip

unzip hostapd.zip

sudo mv /usr/sbin/hostapd /usr/sbin/hostapd.bak

sudo mv hostapd /usr/sbin/hostapd.edimax

sudo ln -sf /usr/sbin/hostapd.edimax /usr/sbin/hostapd

sudo chown root.root /usr/sbin/hostapd

sudo chmod 755 /usr/sbin/hostapd
```

Za podešavanje Wi-Fi pristupne točke potrebno je izmijeniti datoteku hostapd.conf kako bi se podesio naziv mreže, vrsta enkripcije, te lozinka za pristup:

```
sudo nano /etc/hostapd/hostapd.conf
```

te unijeti sljedeće podatke:

```
interface=wlan0

driver=rtl871xdrv

ssid=(naziv wifi mreže )

channel=6

wmm_enabled=1

wpa=1

wpa_passphrase=(šifra wifi mreže)

wpa_key_mgmt=WPA-PSK

wpa_pairwise=TKIP

rsn_pairwise=CCMP

auth_algs=1

macaddr_acl=0
```

Za podešavanje IP adresa Wi-Fi-a, te Ethernet mreže potrebno je izmijeniti datoteku `interfaces`:

```
sudo nano /etc/network/interfaces
```

te upišemo sljedeće:

```
#loopback adapter
auto lo
iface lo inet loopback
#ethernet
iface eth0 inet static
address 192.168.1.118
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.1
#wlan
allow-hotplug wlan0
iface default inet dhcp
iface wlan0 inet static
address 10.10.0.1
netmask 255.255.255.0
```

Kako bi se Raspberry Pi mogao spojiti na mrežu potrebno je unijeti adrese DNS servera:

```
sudo nano /etc/resolv.conf
```

te unesemo DNS servere:

```
nameserver 192.168.1.3
nameserver 195.29.150.3
```

ili Google DNS servere:

```
nameserver 8.8.8.8
nameserver 8.8.4.4
```

Nakon toga instalira se DHCP server kako bi Raspberry Pi mogao dodjeljivati IP adrese uređajima koji se spajaju na njega:

```
sudo apt-get install isc-dhcp-server  
sudo nano /etc/dhcp/dhcpd.conf
```

te je potrebno izmijeniti sljedeće:

```
authoritative; #be careful with this setting  
  
ddns-update-style none;  
  
default-lease-time 600;  
  
max-lease-time 7200;  
  
log-facility local7;  
  
#for the wireless network on wlan0  
  
subnet 10.10.0.0 netmask 255.255.255.0 {  
    range 10.10.0.25 10.10.0.50;  
  
    option domain-name-servers 8.8.8.8, 8.8.4.4;  
  
    option routers 10.10.0.1;  
  
    interface wlan0;  
  
}
```

Nakon toga potrebno je napraviti reboot. Po uključanju može se pokrenuti DHCP server sa naredbama:

```
sudo hostapd -B /etc/hostapd/hostapd.conf  
sudo /etc/init.d/isc-dhcp-server restart
```

pri čemu se sada može spojiti na Wi-Fi mrežu.

Za automatsko pokretanja Wi-Fi mreže potrebno je unijeti sljedeće:

```
sudo nano /etc/default/hostapd
```

i izmijeniti sljedeće:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Kako bi korisnici Wi-Fi mreže imali pristup ostatku mreže, te internetu potrebno je napraviti sljedeće:

```
sudo su  
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Tada se ažurira datoteka sysctl.conf, te od komentira sljedeća linija

```
nano /etc/sysctl.conf  
net.ipv4.ip_forward=1
```

i spremi datoteka. Sljedeći korak je ubaciti IP tablicu za omogućavanje NAT-a. (eth0 je sučelje koje ima pristup internetu)

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Isto tako treba spremi postavke IP tablice, kako bi bili dostupni kod svakog pokretanja Raspberry Pi-a. Za spremanje ovih postavki u datoteku, upišu se sljedeće linije:

```
iptables-save > /etc/iptables.up.rules
```

Kako bi se pokrenuli kod pokretanja, mora se napraviti skripta u /etc/network/if-pre-up.d/

```
nano /etc/network/if-pre-up.d/iptables
```

sa sljedećim sadržajem:

```
#!/bin/sh  
#This script restores iptables upon reboot  
iptables-restore < /etc/iptables.up.rules  
exit 0
```

Tada se provjeravaju dopuštenja kako bi se skripta pokrenula kod pokretanja:

```
chown root:root /etc/network/if-pre-up.d/iptables  
chmod +x /etc/network/if-pre-up.d/iptables  
chmod 755 /etc/network/if-pre-up.d/iptables
```

nakon čega je omogućeno spajanje se na Wi-Fi mrežu, te kroz nju na Internet.

5.3. Instaliranje LAMP stack-a

LAMP stack je skraćenica za servere na kojima se vrti Linux operativni sustav, program za web server je Apache, a koriste MySQL za spremanje u bazi podataka, a kao programski jezik koriste PHP.

5.3.1. Apache web server sa PHP-om

Nakon što se ulogira u SSH potrebno je upisati sljedeće:

```
sudo apt-get update
```

te ako se pojavi greška, samo ponoviti naredbu. Nakon toga instalira se Apache web server sa sljedećom naredbom:

```
sudo apt-get install apache2 php5 libapache2-mod-php5
```

Po pitanju „Do you want to continue“ pritisne se „y“ i enter. Počekati da se postupak sam završi. Po završetku ako se pojavi greška, upisati sljedeće za njezino otklanjanje:

```
sudo groupadd www-dana
```

```
sudo usermod -g www-data www-dana
```

Sada restartati Apache sa linijom:

```
sudo service apache2 restart
```

pri čemu ne bi trebalo biti nikakvih grešaka. Ako se sada pokrene Internet preglednik u računalu i upiše IP adresa od Raspberry Pi-a prikazat će se Apache instalacijska stranica.

Za uključenje htaccess datoteka mora se modificirati config datoteka:

```
sudo nano /etc/apache2/sites-enabled/000-default
```

i pritisnuti enter, te izmijeniti sljedeću liniju:

```
"AllowOverride None" – promijeniti u "AllowOverride ALL"
```

Zatim se drži tipka CTRL i pritisne „X“, zatim tipku „Y“ i enter, te se restarta Apache sa:

```
sudo service apache2 restart
```

5.3.2. MySQL

Za instalaciju MySQL-a potrebno se ponovno ulogirati u SSH ili samo nastaviti sa naredbom:

```
sudo apt-get install mysql-server mysql-client php5-mysql
```

i pritisnuti enter. Po pitanju „Do you want to continue“ pritisnuti „y“ i enter. Počekati da se skine program i instalira, te na sljedeće pitanje upisati lozinku za pristup MySQL-u kao root user:

```
"New password for the MySQL "root" user"
```

5.3.3. FTP server

Kao FTP server koristi se VSFTP, koji je jednostavno podesiti. Ponovno pokrenuti SSH, ili nastavite ako je već pokrenut, te upisati:

```
sudo chown -R pi /var/www  
sudo apt-get install vsftpd
```

Zatim je potrebno izmijeniti datoteku vsftpd.conf:

```
sudo nano /etc/vsftpd.conf
```

U datoteci promijeniti sljedeće linije:

```
anonymous_enable=YES promijeniti u anonymous_enable=NO  
#local_enable=YES promijeniti u local_enable=YES  
#write_enable=YES promijeniti u write_enable=YES
```

i na kraj dodati liniju:

```
force_dot_files=YES
```

Zatim se drži tipka CTRL i pritisne „X“, zatim tipka „Y“ i enter, te se ponovno pokrene FTP server:

```
sudo service vsftpd restart
```

Sada je potrebno promijeniti direktorij svaki puta kada se ulogira na FTP server, ovo se rješava na sljedeći način:

```
sudo -i  
passwd root
```

upiše se bilo koja privremena lozinka, te izađe iz SSH:

```
exit  
exit
```

Pokrene se novi SSH, te ulogira se sljedećim podacima:

```
Username: root  
Password: ona upisana iznad
```

te izmijeniti datoteku passwd:

```
nano /etc/passwd
```

i izmijeniti liniju :

```
"pi:x;1000:1000:Raspberry Pi User,,,:home/pi:/bin/bash"
```

tako da se ispred nje doda znak #:

```
"#pi:x;1000:1000:Raspberry Pi User,,,:home/pi:/bin/bash"
```

Zatim držati tipku CTRL i pritisnuti „X“, zatim tipku „Y“ i enter, te upisati sljedeće:

```
usermod -d /var/www pi
```

zatvoriti SSH i pokrenete novi sa podacima:

```
Username: pi  
Password: lozinka
```

i upisati sljedeće:

```
sudo -i  
usermod -L root  
exit
```

Sada je moguće pristupiti sa FTP klijentom koristeći sljedeće podatke:

```
Host: IP adresa Raspberry Pi-a  
Username: pi  
Password: lozinka  
Port: 21
```

5.3.4. phpMyAdmin

Instaliranje phpMyAdmin-a [Slika 32] je jednostavno i omogućuje lakše korištenje MySQL baza. Sa stranice <http://www.phpmyadmin.net> skine se RAR datoteka, otpakira u neki direktorij, te učita na server preko FTP klijenta.

Kada se u web preglednik upiše <http://raspberrypi-ip/admin> potrebno se ulogirati sa sljedećim podacima:

Username: **root**

Password: (lozinka upisana kod instalacije MySQL-a)

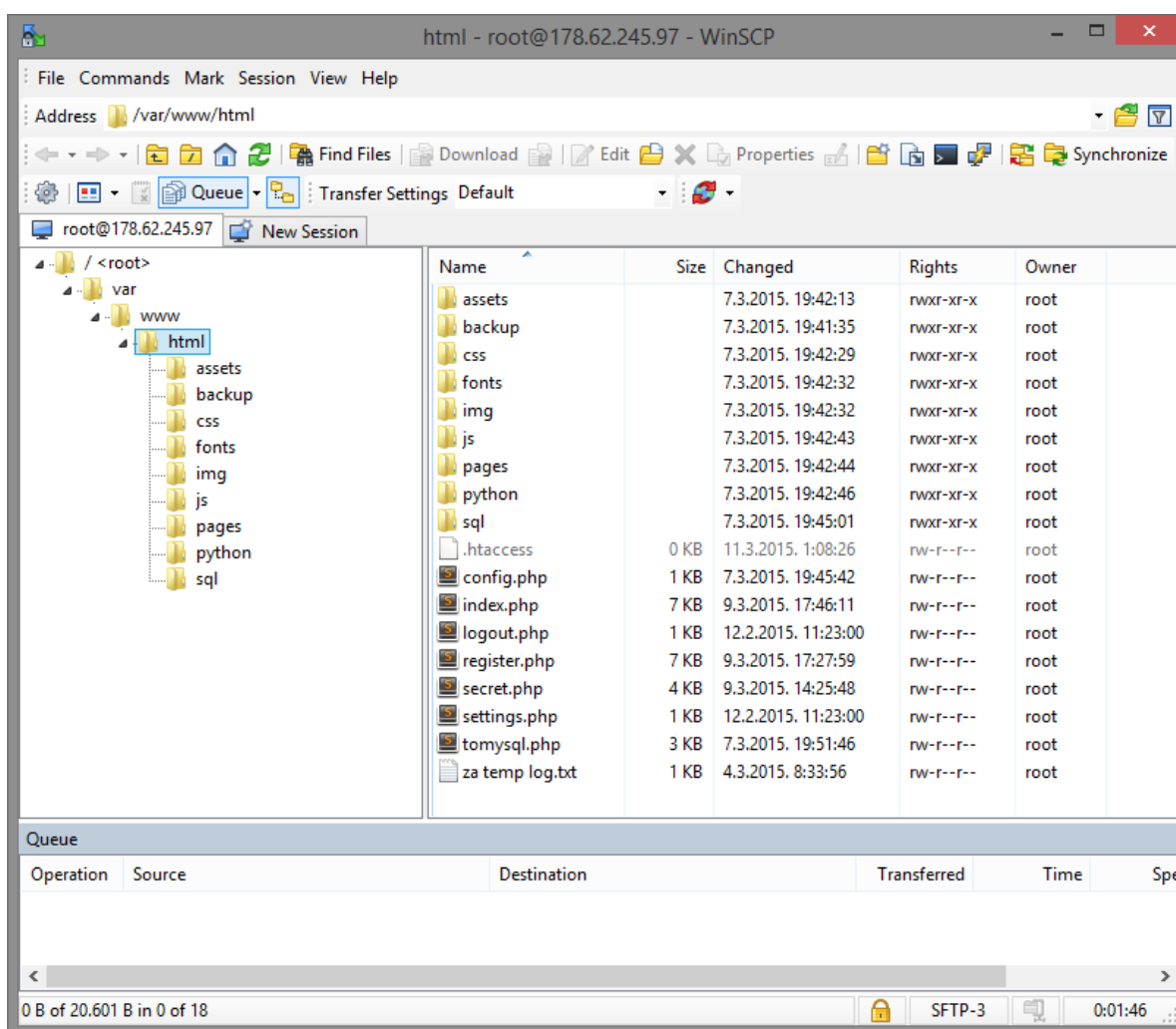
The screenshot shows the phpMyAdmin interface for a MySQL database named 'elektronpi'. The selected table is 'input'. The table contains 25 rows of data. The columns are: ID_input, DateTime, IP_address, Unit_address, Function_code, Address, Type, and Data. The data shows a sequence of entries from ID 1 to 25, with dates and times, IP addresses, and unit addresses.

ID_input	DateTime	IP_address	Unit_address	Function_code	Address	Type	Data
1	2015-05-07 21:30:30	TCP192.168.1.150	255	3	0	int	0
2	2015-05-07 21:30:30	TCP192.168.1.150	255	3	1	int	0
3	2015-05-07 21:30:30	TCP192.168.1.150	255	3	2	int	0
4	2015-05-07 21:30:30	TCP192.168.1.150	255	3	3	int	0
5	2015-05-07 21:30:30	TCP192.168.1.150	255	3	4	int	0
6	2015-05-07 21:30:30	TCP192.168.1.150	255	3	5	int	0
7	2015-05-07 21:30:30	TCP192.168.1.150	255	3	6	int	0
8	2015-05-07 21:30:30	TCP192.168.1.150	255	3	7	int	0
9	2015-05-07 21:30:30	TCP192.168.1.150	255	3	8	int	0
10	2015-05-07 21:30:30	TCP192.168.1.150	255	3	9	int	0
11	2015-05-07 21:30:32	TCP192.168.1.150	255	3	0	int	0
12	2015-05-07 21:30:32	TCP192.168.1.150	255	3	1	int	0
13	2015-05-07 21:30:32	TCP192.168.1.150	255	3	2	int	0
14	2015-05-07 21:30:32	TCP192.168.1.150	255	3	3	int	0
15	2015-05-07 21:30:32	TCP192.168.1.150	255	3	4	int	0
16	2015-05-07 21:30:32	TCP192.168.1.150	255	3	5	int	0
17	2015-05-07 21:30:32	TCP192.168.1.150	255	3	6	int	0
18	2015-05-07 21:30:32	TCP192.168.1.150	255	3	7	int	0
19	2015-05-07 21:30:32	TCP192.168.1.150	255	3	8	int	0
20	2015-05-07 21:30:32	TCP192.168.1.150	255	3	9	int	0
21	2015-05-07 21:30:33	TCP192.168.1.150	255	3	0	int	0
22	2015-05-07 21:30:33	TCP192.168.1.150	255	3	1	int	0
23	2015-05-07 21:30:33	TCP192.168.1.150	255	3	2	int	0
24	2015-05-07 21:30:33	TCP192.168.1.150	255	3	3	int	0
25	2015-05-07 21:30:33	TCP192.168.1.150	255	3	4	int	0

Slika 32. Izgled phpMyAdmin sučelja za upravljanje MySQL-om

5.4. Inicijalizacija Web aplikacije

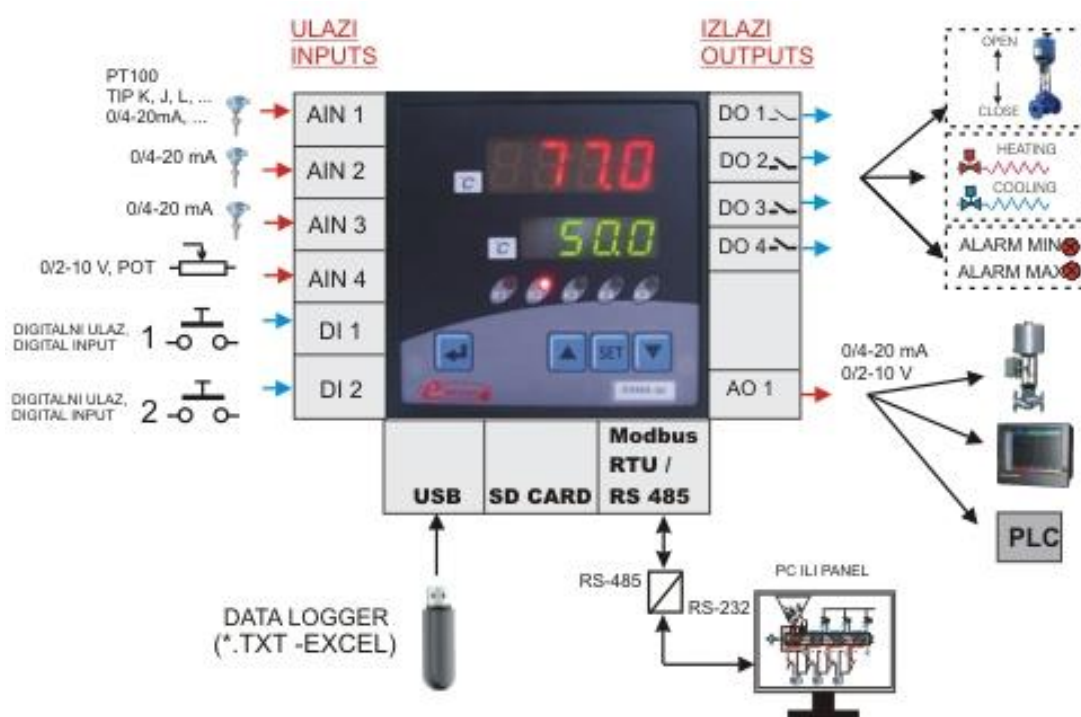
Za postavljanje web aplikacije potrebno je ulogirati se sa FTP klijentom kao WinSCP [Slika 33] i u www direktorij ubaciti sve datoteke, te omogućiti javni pristup njima. Pomoću phpMyAdmin sučelja potrebno je postaviti bazu podataka pomoću skripte u direktoriju _installation, te postaviti novog korisnika i njegova prava pristupa tim datotekama. Sada bi na IP adresi uređaja 10.10.0.1 ako se spaja preko bežične mreže trebala pojaviti početna stranica koja traži podatke za prijavu. Potrebno se ulogirati sa admin podacima kako bi se podesili uređaji i dodali novi korisnici.



Slika 33. WinSCP – prebacivanje datoteka na server preko mreže

6. ISPITIVANJE NA REALNOM INDUSTRIJSKOM PROCESU

Cilj ispitivanja rada uređaja na realnom industrijskom procesu je vidjeti pouzdanost takvog uređaja u realnim uvjetima. Za ispitivanje korišten je Raspberry Pi Model B prve generacije [Slika 35] sa samo 256 MB radne memorije. Uređaj se nalazi u plastičnom kućištu te je izmjerena radna temperatura procesora tijekom testiranja koja je iznosila oko 50 °C. Uređaj se napaja preko mikro USB konektora i izvora 5V/300mA. Na USB konektor spojen je pretvarač sa USB-a na RS-232, a zatim pretvarač signala sa RS-232 na RS-485. Na kraju sa dvije žice povezan je PID regulator GAMA-3k [Slika 34] tvrtke Elektron Erma-Strmec koji regulira temperaturu u peći za kaljenje.



Slika 34. Mogućnosti PID regulatora GAMA-3k

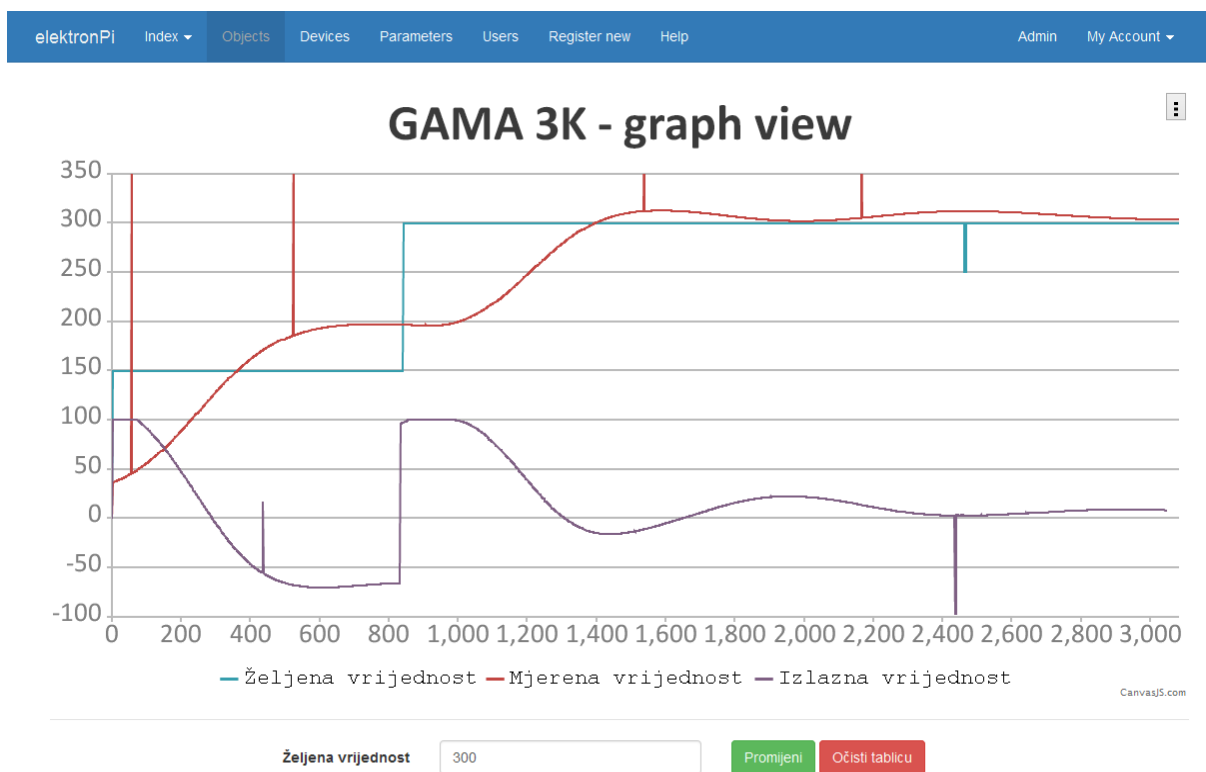
Regulator ima 4 ulaza, jedan univerzalni analogni, 2 strujna i jedan naponski, te 4 relejna izlaza i jedan analogni univerzalni izlaz. Na dva LED displaya prikazane su mjerene i zadane vrijednosti, a pomoću tipkovnice vrši se parametrizacija i prikaz ostalih parametara regulacije. Regulator ima mogućnost logiranja parametara na USB memoriju ili SD karticu, ali se priključak za to nalazi iza uređaja. Kao komunikaciju prema ostaloj periferiji podržava standard Modbus RTU preko RS-485 mreže, dok novije verzije podržavaju i Modbus TCP preko Ethernet mreže.



Slika 35. Ispitivanje uređaja na regulatoru GAMA-3k

U ovom slučaju na regulator je spojena Ni-CrNi sonda za mjerenje temperature u peći do 1200 °C, a izlazni relej upravlja sa sklopnikom koji uključuje i isključuje grijač snage 2 kW. Peć za kaljenje izolirana je od okoline te služi za umjeravanje sonde za temperaturu. Parametri PID regulatora su unaprijed podešeni preko ugrađene funkcije za automatsko podešavanje.

U grafičkom sučelju uređaja za nadzor i upravljanje preko web preglednika na laptopu koji je spojen na Wi-Fi mrežu uređaja, podešeni su samo neki osnovni parametri zbog nepotpune dokumentacije regulatora, kao mjerena veličina, zadana veličina i izlazna veličina iz regulatora. Prvi cilj je bila uspostava komunikacije sa regulatorom, znači podešavanje adrese, te parametara komunikacije, koji moraju biti isti kao i oni podešeni na regulatoru. Nakon toga u grafičkom sučelju [Slika 36] zadana je željena vrijednost, pri čemu je regulator krenuo sa radom, a u web aplikaciji se iscrtava na grafu mjerena vrijednost i izlazna vrijednost iz regulatora sa periodom osvježavanja 2 s.



Slika 36. Graf temperature

Pošto nisu do kraja svi moduli web aplikacije implementirani, zbog veličine samog projekta, druga ispitivanja nisu napravljena. Kako bi se saznale stvarne mogućnosti i brzine prikupljanja podataka potrebno je spojiti više uređaja, te istovremeno sa svih prikupljati podatke.

7. ZAKLJUČAK

U radu je ponuđeno jedno moguće rješenje za bežični nadzor i upravljanje industrijskih procesa, odnosno regulatora unutar industrijskog postrojenja. Prema zahtjevima proizvođača regulatora osmišljen je uređaj temeljen na računalo Raspberry Pi, kojeg je moguće montirati u industrijsku okolinu. Uređaj podržava Modbus protokol preko RS-485 i Ethernet mreže za spajanje podređenih uređaja, dok korisnik može pristupiti web aplikaciji za nadzor i upravljanje preko lokalne mreže sa računala ili preko bežične mreže sa svog mobilnog telefona ili tablet računala. Na ovaj način omogućeno je spajanje više korisnika na sustav, bez instaliranja bilo kakvog programa na svoje računalo, preko web preglednika. Zbog opširnosti web aplikacije nisu implementirani svi moduli, pa je testiran rad uređaja na jednom podređenom regulatoru i snimana karakteristika regulacije na grafu.

Mogućnosti za daljnji razvoj ovakvog sustava vrlo su velike. Potrebna su daljnja testiranja web aplikacije sa radom na više uređaja, te nabavka najnovije verzije Raspberry Pi. Noviju verziju web aplikacije treba bazirati na boljem serverskom radnom okruženju Laravel, koja sadrži puno korisnih funkcija i ima bolju dokumentaciju. Grafičko sučelje potrebno je doraditi kako bi ga korisnik mogao prilagoditi prema svojoj potrebi. Treba razmotriti i mogućnost nadogradnje da takav uređaj omogućuje nadzor i upravljanje preko interneta.

LITERATURA

- [1] Membrey, P., Hows, D.: Learn Raspberry Pi with Linux, Technology in Action, 2012.
- [2] Richardson, M., Wallace, S.: Getting Started with Raspberry Pi, O'Reilly Media, 2012.
- [3] Raspberry Pi Foundation: Raspberry Pi Documentation, <https://www.raspberrypi.org>, 10. ožujka 2015.
- [4] Modbus Organisation: Modbus Application Protocol Specification V1.1b3, <http://www.modbus.org>, 18. ožujka 2015.
- [5] Gast, M.: 802.11 Wireless Networks: The Definitive Guide, O'Reilly Media, 2005.
- [6] Pitt, C.: Pro PHP MVC, Apress, 2012.
- [7] Dev Metal: Huge Documentation, <https://github.com/panique/huge>, 20. ožujka 2015.
- [8] Rahman, S. F.: Jump Start Bootstrap, SitePoint, 2014.

PRILOZI

Skripta za prikupljanje podataka

```
1. #!/usr/bin/python
2. import MySQLdb
3. import time
4. from pymodbus.client.sync import ModbusTcpClient
5. from pymodbus.client.sync import ModbusSerialClient
6. from pymodbus.constants import Endian
7. from pymodbus.payload import BinaryPayloadDecoder
8. from pymodbus.payload import BinaryPayloadBuilder
9. from pymodbus.diag_message import *
10. from pymodbus.file_message import *
11. from pymodbus.other_message import *
12. from pymodbus.mei_message import *
13. from pymodbus.transaction import ModbusRtuFramer
14.
15. print "working..."
16. while True:
17.     db = MySQLdb.connect("localhost", "elektronpi", "barica0581", "elektronpi")
18.     curs=db.cursor()
19.     sql_all = "SELECT * FROM input LIMIT 0,1"
20.     curs.execute (sql_all)
21.     rows= curs.fetchall()
22.     for row in rows:
23.         #print str(row)
24.         if row[2].find("TCP") != -1:
25.             protokol = "TCP"
26.             client = ModbusTcpClient(row[2].replace("TCP",""))
27.         elif row[2].find("RTU") != -1:
28.             protokol = "RTU"
29.             try:
30.
31.                 client = ModbusSerialClient(retries=10, rtscts=True, framer=ModbusRtuFramer, method='rt
32.                 u', port='/dev/ttyUSB0', stopbits = 1, bytesize = 8, parity = 'E', baudrate = 38400, ti
33.                 meout=0.1)
34.             except:
35.                 pass
36.             try:
37.                 connection = client.connect()
38.             except:
39.                 connection = 0
40.             if connection==True:
41.                 try:
42.                     if row[4] == 3:
43.                         if row[6] == 'int':
44.                             result = client.read_holding_registers(row[5],1,unit=row[3])
45.                             client.close()
46.                             if result.function_code==row[4]:
47.
48.                 sql_ins = "INSERT INTO output (DateTime, IP_address, Unit_address, Function_code, Addre
49.                 ss, Type, Data) VALUES (NOW(),%s, %s, %s, %s, %s, %s)"
50.
51.                 curs.execute(sql_ins,(str(row[2]),str(row[3]),str(row[4]),str(row[5]),str(row[6]),str(r
52.                 esult.registers[0])))
53.                 db.commit()
54.             else:
```

```
48. sql_ins = "INSERT INTO output (DateTime, IP_address, Unit_address, Function_code, Address, Type, Data) VALUES (NOW(),%s, %s, %s, %s, %s, %s)"
49. curs.execute(sql_ins,(str(row[2]),str(row[3]),str(row[4]),str(row[5]),str(row[6]),'error'))
50.         db.commit()
51.         elif row[6] == 'float':
52.             result =client.read_holding_registers(row[5],2,unit=row[3])
53.             client.close()
54.             if result.function_code==row[4]:
55.
56. sql_ins = "INSERT INTO output (DateTime, IP_address, Unit_address, Function_code, Address, Type, Data) VALUES (NOW(),%s, %s, %s, %s, %s, %s)"
57. decoder = BinaryPayloadDecoder.fromRegisters(result.registers, endian=Endian.Big)
58.         decoded = decoder.decode_32bit_float()
59.         decoded = format(decoded, '.3f')
60. curs.execute (sql_ins,(str(row[2]),str(row[3]),str(row[4]),str(row[5]),str(row[6]),str(decoded)))
61.         db.commit()
62.         else:
63.
64. sql_ins = "INSERT INTO output (DateTime, IP_address, Unit_address, Function_code, Address, Type, Data) VALUES (NOW(),%s, %s, %s, %s, %s, %s)"
65. curs.execute (sql_ins,(str(row[2]),str(row[3]),str(row[4]),str(row[5]),str(row[6]),'error'))
66.         db.commit()
67.         elif row[4] == 4:
68.             if row[6] == 'int':
69.                 result = client.read_input_registers(row[5],1,unit=row[3])
70.                 client.close()
71.                 if result.function_code==row[4]:
72.
73. sql_ins = "INSERT INTO output (DateTime, IP_address, Unit_address, Function_code, Address, Type, Data) VALUES (NOW(),%s, %s, %s, %s, %s, %s)"
74. curs.execute (sql_ins,(str(row[2]),str(row[3]),str(row[4]),str(row[5]),str(row[6]),str(result.registers[0])))
75.         db.commit()
76.         else:
77.
78. sql_ins = "INSERT INTO output (DateTime, IP_address, Unit_address, Function_code, Address, Type, Data) VALUES (NOW(),%s, %s, %s, %s, %s, %s)"
79. curs.execute (sql_ins,(str(row[2]),str(row[3]),str(row[4]),str(row[5]),str(row[6]),'error'))
80.         db.commit()
81.         elif row[6] == 'float':
82.             result = client.read_input_registers(row[5],2,unit=row[3])
83.             client.close()
84.             if result.function_code==row[4]:
85.
86. sql_ins = "INSERT INTO output (DateTime, IP_address, Unit_address, Function_code, Address, Type, Data) VALUES (NOW(),%s, %s, %s, %s, %s, %s)"
87. decoder = BinaryPayloadDecoder.fromRegisters(result.registers, endian=Endian.Big)
88.         decoded = decoder.decode_32bit_float()
89.         decoded = format(decoded, '.3f')
90. curs.execute (sql_ins,(str(row[2]),str(row[3]),str(row[4]),str(row[5]),str(row[6]),str(decoded)))
91.         db.commit()
92.         else:
```



```
88. sql_ins = "INSERT INTO output (DateTime, IP_address, Unit_address, Function_code, Address, Type, Data) VALUES (NOW(),%s, %s, %s, %s, %s, %s)"
89. curs.execute (sql_ins,(str(row[2]),str(row[3]),str(row[4]),str(row[5]),str(row[6]),'error'))
90.         db.commit()
91.         elif row[4] == 6:
92.             if row[6] == 'int':
93.                 result = client.write_register(row[5],int(row[7]),unit=row[3])
94.                 client.close()
95.             if row[6] == 'float':
96.                 builder = BinaryPayloadBuilder(endian=Endian.Big)
97.                 builder.add_32bit_float(float(row[7]))
98.                 payload = builder.build()
99.
100. client.write_registers(row[5],payload, unit=row[3], skip_encode=True)
101.         client.close()
102.         sql_del = "DELETE FROM input WHERE ID_input LIKE (%s)"
103.         curs.execute (sql_del,{str(row[0])})
104.         db.commit()
105.     except:
106.         #print "no connection"
107.
108. sql_ins = "INSERT INTO output (DateTime, IP_address, Unit_address, Function_code, Address, Type, Data) VALUES (NOW(),%s, %s, %s, %s, %s, %s)"
109. curs.execute (sql_ins,(str(row[2]),str(row[3]),str(row[4]),str(row[5]),str(row[6]),'try error'))
110.         db.commit()
111.         sql_del = "DELETE FROM input WHERE ID_input LIKE (%s)"
112.         curs.execute(sql_del,{str(row[0])})
113.         db.commit()
114.         time.sleep(1)
115.     else:
116.         #print "no connection"
117.
118. sql_ins = "INSERT INTO output (DateTime, IP_address, Unit_address, Function_code, Address, Type, Data) VALUES (NOW(),%s, %s, %s, %s, %s, %s)"
119. curs.execute (sql_ins,(str(row[2]),str(row[3]),str(row[4]),str(row[5]),str(row[6]),'no connection'))
120.         db.commit()
121.         sql_del = "DELETE FROM input WHERE ID_input LIKE (%s)"
122.         curs.execute(sql_del,{str(row[0])})
123.         db.commit()
124.         time.sleep(1)
125.     try:
126.         curs.close()
127.         db.close()
128.         client.close()
129.     except:
130.         pass
131.     time.sleep(0.001)
```